pysbol2 Documentation

Release 2.3.0

Bryan Bartley

Mar 30, 2018

Contents

pySBOL is a SWIG-Python wrapper around libSBOL, a module for reading, writing, and constructing genetic designs according to the standardized specifications of the Synthetic Biology Open Language (SBOL).

CHAPTER 1

Introduction

pySBOL provides Python interfaces and their implementation for Synthetic Biology Open Language (SBOL). The current version of pySBOL implements SBOL Core Specification 2.1.0. The library provides an API to work with SBOL objects, the functionality to read GenBank, FASTA, and SBOL version 1 and 2 documents as XML/RDF files, to write GenBank, FASTA, and SBOL version 1 and 2 documents, and to validate the correctness of SBOL 2 documents. This is a Python binding for C/C++ based libSBOL. Currently, pySBOL supports Python version 2.7 and 3.6 only. pySBOL is made freely available under the Apache 2.0 license.

To install, go to Installation Page.

- The current snapshot of pySBOL is available on GitHub.
- Any problems or feature requests for pySBOL should be reported on the GitHub issue tracker.
- An overview of pySBOL can be found here.
- For further information about the pySBOL library, its implementation, or its usage, please feel free to contact the libSBOL team.

pySBOL is brought to you by Bryan Bartley, Kiri Choi, and SBOL Developers.

Current support for the development of pySBOL is generously provided by the NSF through the Synthetic Biology Open Language Resource collaborative award.

CHAPTER 2

Installation

Currently, we support Python 2.7 and Python 3.6 for both 32 bit and 64 bit architecture. Python by default comes with package manager. Follow the steps below to install pySBOL. If you have Windows, and would like to try our Windows binary installers, check Using Installer for Windows section.

2.1 Using Pip

pySBOL is available for Windows and Mac OSX via PyPI, which is the simplest method to obtain pySBOL. To install pySBOL using pip, run following line on console:

```
pip install pysbol
```

If you encounter permission errors on Mac OSX, you may install pysbol to your user site-packages directory as follows:

```
pip install pysbol --user
```

Or alternatively, you may install as a super-user:

```
sudo -H pip install pysbol
```

To update pySBOL using pip, run:

```
pip install -U pysbol
```

2.2 Using Python

1 - Download the source code of latest release here and extract it. If you would like to try out our latest snapshot, use git and type following command in the console or terminal which will clone the source under pysbol folder.

```
git clone https://github.com/SynBioDex/pysbol.git
```

2 - Open your console or terminal. Go to package's root directory and Run the installer script by using the following command line. This will install pySBOL2 to the Python release associated with the console or terminal you are using.

```
python setup.py install
```

If you are having problems, make sure your console/terminal is associated with the right Python environment you wish to use.

3 - Test the pySBOL by importing it in Python.

```
import sbol
```

If you have trouble importing the module with the setup script, check to see if there are multiple Python installations on your machine and also check the output of the setup script to see which version of Python is the install target. You can also test the module locally from inside the Mac_OSX/sbol or Win_32/sbol folders.

2.3 Using Installer for Windows

We provide binary installers for Windows users only. Simply download the installers and execute it to install it. Installer will look for your local Python distributions.

Be sure to use the installers with the same Python version and architecture with the one installed in your local machine!

2.4 For Linux Users

Currently, Linux users should build pySBOL from source through libSBOL. Go to libSBOL installation page and follow the instructions for Debian/Ubuntu.

CHAPTER 3

Getting Started with SBOL

This beginner's guide introduces the basic principles of pySBOL for new users. Most of the examples discussed in this guide are excerpted from the example script . For more comprehensive documentation about the API, refer to documentation about specific classes and methods. For more detail about the SBOL standard, visit sbolstandard.org or refer to the specification document.

3.1 Creating an SBOL Document

In a previous era, engineers might sit at a drafting board and draft a design by hand. The engineer's drafting sheet in pySBOL is called a Document. The Document serves as a container, initially empty, for SBOL data objects which represent elements of a biological design. Usually the first step is to construct a Document in which to put your objects. All file I/O operations are performed on the Document. The `read` and `write` methods are used for reading and writing files in SBOL format.

```
>>> doc = Document()
>>> doc.read('crispr_example.xml')
>>> doc.write('crispr_example_out.xml')
```

Reading a Document will wipe any existing contents clean before import. However, you can import objects from multiple files into a single Document object using Document.append(). This can be advantageous when you want to integrate multiple objects from different files into a single design. This kind of data integration is an important and useful feature of SBOL.

Each SBOL object in a Document is uniquely identified by a special string of characters called a Uniform Resource Identifier (URI). A URI is used as a key to retrieve objects from the Document. To see the identity of objects in a Document, objects can be iterated over following a Python iterator pattern.

```
>>> for obj in doc:
...     print(obj)
...
http://sbols.org/CRISPR_Example/mKate_seq/1.0.0
http://sbols.org/CRISPR_Example/gRNA_b_nc/1.0.0
http://sbols.org/CRISPR_Example/mKate_cds/1.0.0
```

```
.
.
```

These URIs are said to be **sbol-compliant**. An sbol-compliant URI consists of a scheme, a namespace, a local identifier (also called a `displayId`), and a version number. In this tutorial, we use URIs of the type `http://sbols.org/CRISPR_Example/my_obj/1.0.0`, where the scheme is indicated by `http://`, the namespace is `http://sbols.org/CRISPR_Example`, the local identifier is `my_object`, and the version is `1.0.0`. SBOL-compliant URIs enable shortcuts that make the pySBOL API easier to use and are enabled by default. However, users are not required to use sbol-compliant URIs if they don't want to, and this option can be turned off.

Based on our inspection of objects contained in the Document above, we can see that these objects were all created in the namespace `http://sbols.org/CRISPR_Example`. Thus, in order to take advantage of SBOL-compliant URIs, we set an environment variable that configures this namespace as the default. In addition we set some other configuration options.

```
>>> setHomespace('http://sbols.org/CRISPR_Example')
>>> Config.setOption('sbol_compliant_uris', True)
>>> Config.setOption('sbol_typed_uris', False)
```

A Document may contain different types of SBOL objects, including ComponentDefinitions, ModuleDefinitions, Sequences, and Models. These objects are collectively referred to as TopLevel objects because they can be referenced directly from a Document. The total count of objects contained in a Document is determined using the `len` function. To view an inventory of objects contained in the Document, simply `print` it.

```
>>> len(doc)
31
>>> print(doc)
Attachment.....................0
Collection.....................0
CombinatorialDerivation........0
ComponentDefinition............25
Implementation.................0
Model..........................0
ModuleDefinition...............2
Sequence.......................4
Analysis.......................0
Build..........................0
Design.........................0
SampleRoster...................0
Test...........................0
Activity.......................0
Agent..........................0
Plan...........................0
Annotation Objects.............0
---
Total..........................31
```

Objects are categorized into object store according to their respective SBOL type. To review objects of a given type, simply iterate over the objects in that store. For example:

```
>>> for obj in doc:
...     print(obj)
...
http://sbols.org/CRISPR_Example/mKate_seq/1.0.0
http://sbols.org/CRISPR_Example/gRNA_b_nc/1.0.0
http://sbols.org/CRISPR_Example/mKate_cds/1.0.0
.
```

.

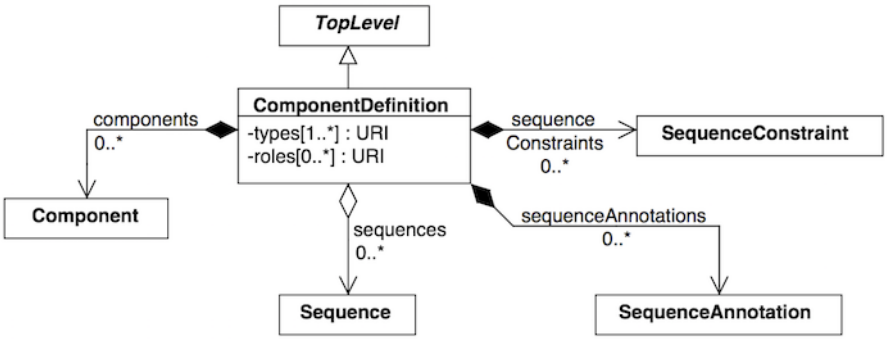
These objects are sorted into object stores based on the type of object. For example to view `ComponentDefinition` objects specifically, iterate through the *Document.componentDefinitions* store:

Similarly, you can iterate through Document.moduleDefinitions(), Document.sequences(), Document.models(), or any top level object. The last type of object, Annotation Objects is a special case which will be discussed later.

3.2 Creating SBOL Data Objects

Biological designs can be described with SBOL data objects, including both structural and functional features. The principle classes for describing the structure and primary sequence of a design are ComponentDefinitions, Components, Sequences, and SequenceAnnotations. The principle classes for describing the function of a design are ModuleDefinitions, Modules, Interactions, and Participations. Other classes such as Design, Build, Test, Analysis, Activity, and Plan are used for managing workflows.

In the official SBOL specification document, classes and their properties are represented as box diagrams. Each box represents an SBOL class and its attributes. Following is an example of the diagram for the ComponentDefinition class which will be referred to in later sections. These class diagrams follow conventions of the Unified Modeling Language.



As introduced in the previous section, SBOL objects are identified by a uniform resource identifier (URI). When a new object is constructed, the user must assign a unique identity. The identity is ALWAYS the first argument supplied to the constructor of an SBOL object. Depending on which configuration options for pySBOL are specified, different algorithms are applied to form the complete URI of the object. The following examples illustrate these different configuration options.

The first set of configuration options demonstrates 'open-world' mode, which means that URIs are explicitly specified in full by the user, and the user is free to use whatever convention or conventions they want to form URIs. Open-world configuration can be useful sometimes when integrating data objects derived from multiple files or web resources, because it makes no assumptions about the format of URIs.

```
>>> setHomespace('')
>>> Config.setOption('sbol_compliant_uris', False)
>>> Config.setOption('sbol_typed_uris', False)
>>> crispr_template = ModuleDefinition('http://sbols.org/CRISPR_Example/CRISPR_
↪Template')
>>> print(crispr_template)
http://sbols.org/CRISPR_Example/CRISPR_Template
```

The second set of configuration options demonstrates use of a default namespace for constructing URIs. The advantage of this approach is simply that it reduces repetitive typing. Instead of typing the full namespace for a URI every time an object is created, the user simply specifies the local identifier. The local identifier is appended to the namespace. This is a handy shortcut especially when working interactively in the Python interpreter.

```
>>> setHomespace('http://sbols.org/CRISPR_Example/')
>>> Config.setOption('sbol_compliant_uris', False)
>>> Config.setOption('sbol_typed_uris', False)
>>> crispr_template = ModuleDefinition('CRISPR_Template')
>>> print(crispr_template)
http://sbols.org/CRISPR_Example/CRISPR_Template
```

The third set of configuration options demonstrates SBOL-compliant mode. In this example, a version number is appended to the end of the URI. Additionally, when operating in SBOL-compliant mode, the URIs of child objects are algorithmically constructed according to automated rules (not shown here).

```
>>> setHomespace('http://sbols.org/CRISPR_Example/')
>>> Config.setOption('sbol_compliant_uris', True)
>>> Config.setOption('sbol_typed_uris', False)
>>> crispr_template = ModuleDefinition('CRISPR_Template')
>>> print(crispr_template)
http://sbols.org/CRISPR_Example/CRISPR_Template/1.0.0
```

The final example demonstrates typed URIs. When this option is enabled, the type of SBOL object is included in the URI. Typed URIs are useful because sometimes the user may want to re-use the same local identifier for multiple objects. Without typed URIs this may lead to collisions between non-unique URIs. This option is enabled by default, but the example file CRISPR_example.py does not use typed URIs, so for all the examples in this guide this option is assumed to be disabled.

```
>>> setHomespace('http://sbols.org/CRISPR_Example/')
>>> Config.setOption('sbol_compliant_uris', True)
>>> Config.setOption('sbol_typed_uris', True)
>>> crispr_template_md = ModuleDefinition('CRISPR_Template')
>>> print(crispr_template)
http://sbols.org/CRISPR_Example/ModuleDefinition/CRISPR_Template/1.0.0
>>> crispr_template_cd = ComponentDefinition('CRISPR_Template')
http://sbols.org/CRISPR_Example/ComponentDefinition/CRISPR_Template/1.0.0
```

Constructors for SBOL objects follow a fairly predictable pattern. The first argument is ALWAYS the identity of the object. Other arguments may follow, depending on in the SBOL class has required attributes. Attributes are required if the specification says they are. In a UML diagram, required fields are indicated as properties with a cardinality of 1 or more. For example, a ComponentDefinition (see the UML diagram above) has only one required field, `types`, which specifies one or more molecular types for a component. Required fields SHOULD be specified when calling a constructor. If they are not, they will be assigned default values. The following creates a protein component. If the BioPAX term for protein were not specified, then the constructor would create a ComponentDefinition of type BIOPAX_DNA by default.

```
>>> cas9 = ComponentDefinition('Cas9', BIOPAX_PROTEIN)  # Constructs a protein␣
↪component
>>> target_promoter = ComponentDefinition('target_promoter')  # Constructs a DNA␣
↪component by default
```

3.3 Using Ontology Terms for Attribute Values

Notice the `ComponentDefinition.types` attribute is specified using a predefined constant. The `ComponentDefinition.types` property is one of many SBOL attributes that uses ontology terms as property values. The `ComponentDefinition.types` property uses the *BioPax ontology <https://bioportal.bioontology.org/ontologies/BP/?p=classes&conceptid=root>* to be specific. Ontologies are standardized, machine-readable vocabularies that categorize concepts within a domain of scientific study. The SBOL 2.0 standard unifies many different ontologies into a high-level, object-oriented model.

Ontology terms also take the form of Uniform Resource Identifiers. Many commonly used ontological terms are built-in to pySBOL as predefined constants. If an ontology term is not provided as a built-in constant, its URI can often be found by using an ontology browser tool online. Browse Sequence Ontology terms here <http://www.sequenceontology.org/browser/obob.cgi> ‘ and ‘Systems Biology Ontology terms here. While the SBOL specification often recommends particular ontologies and terms to be used for certain attributes, in many cases these are not rigid requirements. The advantage of using a recommended term is that it ensures your data can be interpreted or visualized by other applications that support SBOL. However in many cases an application developer may want to develop their own ontologies to support custom applications within their domain.

The following example illustrates how the URIs for ontology terms can be easily constructed, assuming they are not already part of pySBOL's built-in ontology constants.

```
>>> SO_ENGINEERED_FUSION_GENE = SO + '0000288'  # Sequence Ontology term
>>> SO_ENGINEERED_FUSION_GENE
'http://identifiers.org/so/SO:0000288'
>>> SBO_DNA_REPLICATION = SBO + '0000204'  # Systems Biology Ontology term
>>> SBO_DNA_REPLICATION
'http://identifiers.org/biomodels.sbo/SBO:0000204'
```

3.4 Adding and Getting Objects from a Document

In some cases a developer may want to use SBOL objects as intermediate data structures in a computational biology workflow. In this case the user is free to manipulate objects independently of a Document. However, if the user wishes to write out a file with all the information contained in their object, they must first add it to the Document. This is done using `add` methods. The names of these methods follow a simple pattern, simply “add” followed by the type of object.

```
>>> doc.addModuleDefinition(crispr_template)
>>> doc.addComponentDefinition(cas9)
```

Objects can be retrieved from a Document by using `get` methods. These methods ALWAYS take the object's full URI as an argument.

```
>>> crispr_template = doc.getModuleDefinition('http://sbols.org/CRISPR_Example/CRISPR_
↪Template/1.0.0')
>>> cas9 = doc.getComponentDefinition('http://sbols.org/CRISPR_Example/cas9_generic/1.
↪0.0')
```

When working interactively in a Python environment, typing long form URIs can be tedious. Operating in SBOL-compliant mode allows the user an alternative means to retrieve objects from a Document using local identifiers.

```
>>> Config.setOption('sbol_compliant_uris', True)
>>> Config.setOption('sbol_typed_uris', False)
>>> crispr_template = doc.moduleDefinitions['CRISPR_Template']
>>> cas9 = doc.componentDefinitions['cas9_generic']
```

3.5 Getting, Setting, and Editing Attributes

The attributes of an SBOL object can be accessed like other Python class objects, with a few special considerations. For example, to get the values of the `displayId` and `identity` properties of any object :

Note that `displayId` gives only the shorthand, local identifier for the object, while the `identity` property gives the full URI.

The attributes above return singleton values. Some attributes, like `ComponentDefinition.roles` and `ComponentDefinition.types` support multiple values. Generally these attributes have plural names. If an attribute supports multiple values, then it will return a list. If the attribute has not been assigned any values, it will return an empty list.

```
>>> cas9.types
['http://www.biopax.org/release/biopax-level3.owl#Protein']
>>> cas9.roles
[]
```

Setting an attribute follows the ordinary convention for assigning attribute values:

```
>>> crispr_template.description = 'This is an abstract, template module'
```

To set multiple values:

```
>>> plasmid = ComponentDefinition('pBB1', BIOPAX_DNA, '1.0.0')
>>> plasmid.roles = [ SO_PLASMID, SO_CIRCULAR ]
```

Although properties such as `types` and `roles` behave like Python lists in some ways, beware that list operations like `append` and `extend` do not work directly on these kind of attributes, due to the nature of the C++ bindings. If you need to append values to an attribute, use the following idiom:

```
>>> plasmid.roles = [ SO_PLASMID ]
>>> plasmid.roles = plasmid.roles + [ SO_CIRCULAR ]
```

To clear all values from an attribute, set to None:

```
>>> plasmid.roles = None
```

3.6 Creating, Adding and Getting Child Objects

Some SBOL objects can be composed into hierarchical parent-child relationships. In the specification diagrams, these relationships are indicated by black diamond arrows. In the UML diagram above, the black diamond indicates that ComponentDefinitions are parents of SequenceAnnotations. Properties of this type can be modified using the add method and passing the child object as the argument.

```
>>> point_mutation = SequenceAnnotation('PointMutation')
>>> target_promoter.sequenceAnnotations.add(point_mutation)
```

Alternatively, the create method captures the construction and addition of the SequenceAnnotation in a single function call. The create method ALWAYS takes one argument–the URI of the new object. All other values are initialized with default values. You can change these values after object creation, however.

```
>>> target_promoter.sequenceAnnotations.create('PointMutation')
```

Conversely, to obtain a Python reference to the SequenceAnnotation from its identity:

```
>>> point_mutation = target_promoter.sequenceAnnotations.get('PointMutation')
```

Or equivalently:

```
>>> point_mutation = target_promoter.sequenceAnnotations['PointMutation']
```

3.7 Creating and Editing Reference Properties

Some SBOL objects point to other objects by way of URI references. For example, ComponentDefinitions point to their corresponding Sequences by way of a URI reference. These kind of properties correspond to white diamond arrows in UML diagrams, as shown in the figure above. Attributes of this type contain the URI of the related object.

```
>>> eyfp_gene = ComponentDefinition('EYFPGene', BIOPAX_DNA)
>>> seq = Sequence('EYFPSequence', 'atgnnntaa', SBOL_ENCODING_IUPAC)
>>> eyfp_gene.sequence = seq
>>> print (eyfp_gene.sequence)
'http://sbols.org/Sequence/EYFPSequence/1.0.0'
```

Note that assigning the `seq` object to the `eyfp_gene.sequence` actually results in assignment of the object's URI. An equivalent assignment is as follows:

```
>>> eyfp_gene.sequence = seq.identity
>>> print (eyfp_gene.sequence)
'http://sbols.org/Sequence/EYFPSequence/1.0.0'
```

3.8 Iterating and Indexing List Properties

Some properties can contain multiple values or objects. Additional values can be specified with the add method. In addition you may iterate over lists of objects or values.

```
# Iterate through objects (black diamond properties in UML)
for p in cas9_complex_formation.participations:
    print(p)
    print(p.roles)

# Iterate through references (white diamond properties in UML)
for role in reaction_participant.roles:
    print(role)
```

Numerical indexing of lists works as well:

```
for i_p in range(0, len(cas9_complex_formation.participations)):
    print(cas9_complex_formation.participations[i_p])
```

3.9 Searching a Document

To see if an object with a given URI is already contained in a Document or other parent object, use the `find` method. Note that `find` function returns the target object cast to its base type which is `SBOLObject`, the generic base class

for all SBOL objects. The actual SBOL type of this object, however is `ComponentDefinition`. If necessary the base class can be downcast using the `cast` method.

```
>>> obj = doc.find('http://sbols.org/CRISPR_Example/mKate_gene/1.0.0')
>>> obj
SBOLObject
>>> parseClassName(obj.type)
'ComponentDefinition'
>>> cd = obj.cast(ComponentDefinition)
>>> cd
ComponentDefinition
```

The `find` method is probably more useful as a boolean conditional when the user wants to automatically construct URIs for objects and needs to check if the URI is unique or not. If the object is found, `find` returns an object reference (True), and if the object is not found, it returns None (False). The following code snippet demonstrates a function that automatically generates ComponentDefinitions.

```
def createNextComponentDefinition(doc, local_id):
    i_cdef = 0
    cdef_uri = getHomespace() + '/%s_%d/1.0.0' %(local_id, i_cdef)
    while doc.find(cdef_uri):
        i_cdef += 1
        cdef_uri = getHomespace() + '/%s_%d/1.0.0' %(local_id, i_cdef)
    doc.componentDefinitions.create('%s_%d' %(local_id, i_cdef))
```

3.10 Creating Biological Designs

This concludes the basic methods for manipulating SBOL data structures. Now that you're familiar with these basic methods, you are ready to learn about libSBOL's high-level design interface for synthetic biology. See SBOL Examples.

CHAPTER 4

Biological Parts Repositories

4.1 Mining Genetic Parts From Online Repositories

In today's modern technological society, a variety of interesting technologies can be assembled from "off-the-shelf" components, including cars, computers, and airplanes. Synthetic biology is inspired by a similar idea. Synthetic biologists aim to program new biological functions into organisms by assembling genetic code from off-the-shelf DNA sequences. PySBOL puts an inventory of biological parts at your fingertips.

For example, the iGEM Registry of Standard Biological Parts is an online resource that many synthetic biologists are familiar with. The Registry is an online database that catalogs a vast inventory of genetic parts, mostly contributed by students in the iGEM competition. These parts are now available in SBOL format in the SynBioHub knowledgebase, hosted by Newcastle University. The code example below demonstrates how a programmer can access these data.

The following code example shows how to pull data about biological components from the SynBioHub repository. In order to pull a part, simply locate the web address of that part by browsing the SynBioHub repository online. Alternatively, pySBOL also supports programmatic querying of SynBioHub (see below).

The interface with the SynBioHub repository is represented by a *PartShop* object. The following code retrieves parts corresponding to promoter, coding sequence (CDS), ribosome binding site (RBS), and transcriptional terminator. These parts are imported into a *Document* object, which must be initialized first. See Getting Started with SBOL for more about creating *Documents*.

```
igem = PartShop("https://synbiohub.org")
igem.pull("http://synbiohub.org/public/igem/BBa_T9002/1", doc)
igem.pull("http://synbiohub.org/public/igem/BBa_B0032/1", doc)
igem.pull("http://synbiohub.org/public/igem/BBa_E0040/1", doc)
igem.pull("http://synbiohub.org/public/igem/BBa_B0012/1", doc)

t9002 = doc.getComponentDefinition('http://synbiohub.org/public/igem/BBa_T9002/1')
b0032 = doc.getComponentDefinition('http://synbiohub.org/public/igem/BBa_B0032/1')
e0040 = doc.getComponentDefinition('http://synbiohub.org/public/igem/BBa_E0040/1')
b0012 = doc.getComponentDefinition('http://synbiohub.org/public/igem/BBa_B0012/1')
```

4.2 Searching Part Repos

PySBOL supports three kinds of searches: a **general search**, an **exact search**, and an **advanced search**.

The following query conducts a **general search** which scans through *identity*, *name*, *description*, and *displayId* properties for a match to the search text, including partial, case-insensitive matches to substrings of the property value. Search results are returned as a *SearchResponse* object.

```
records = igem.search("plasmid")
```

By default, the general search looks only for *ComponentDefinitions*, and only returns 25 records at a time in order to prevent server overload. The search above is equivalent to the one below, which explicitly specifies which kind of SBOL object to search for, an offset of 0 (explained below), and a limit of 25 records.

```
records = igem.search("plasmid", SBOL_COMPONENT_DEFINITION, 0, 25)
```

Of course, these parameters can be changed to search for different type of SBOL objects or to return more records. For example, some searches may match a large number of objects, more than the specified limit allows. In this case, it is possible to specify an offset and to retrieve additional records in successive requests. The total number of objects in the repository matching the search criteria can be found using the searchCount method, which has the same call signature as the search method. It is a good idea to put a small delay between successive requests to prevent server overload. The following example demonstrates how to do this. The 100 millisecond delay is implemented using cross-platform C++11 headers chrono and thread. As of the writing of this documentation, this call retrieves 391 records.

```
import time

records = SearchResponse()
search_term = "plasmid"
limit = 25
total_hits = igem.searchCount(search_term)
for offset in range(0, total_hits, limit):
    records.extend( igem.search(search_term, SBOL_COMPONENT_DEFINITION, offset,␣
↪limit) )
    time.sleep(0.1)
```

A *SearchResponse* object is returned by a query and contains multiple records. Each record contains basic data, including identity, displayId, name, and description fields. *It is very important to realize however that the search does not retrieve the complete ComponentDefinition!* In order to retrieve the full object, the user must call *pullComponentDefinition* while specifying the target object's identity.

Records in a *SearchResponse* can be accessed using iterators or numeric indices. The interface for each record behaves exactly like any other SBOL object:

```
for record in records:
    print( record.identity.get() )
```

The preceding examples concern **general searches**, which scan through an object's metadata for partial matches to the search term. In contrast, the **exact search** explicitly specifies which property of an object to search, and the value of that property must exactly match the search term. The following **exact search** will search for *ComponentDefinitions* with a role of promoter:

records = igem.search(SO_PROMOTER, SBOL_COMPONENT_DEFINITION, SBOL_ROLES, 0, 25); .. end

Finally, the **advanced search** allows the user to configure a search with multiple criteria by constructing a *SearchQuery* object. The following query looks for promoters that have an additional annotation indicating that the promoter is regulated (as opposed to constitutive):

```
q = SearchQuery();
q["objectType"].set(SBOL_COMPONENT_DEFINITION);
q["limit"].set(25);
q["offset"].set(0);
q["role"].set(SO_PROMOTER);
q["role"].add("http://wiki.synbiohub.org/wiki/Terms/igem#partType/Regulatory");
total_hits = igem.searchCount(q);
records = igem.search(q);
```

4.3 Submitting Designs to a Repo

Users can submit their SBOL data directly to a PartShop using the PySBOL API. This is important, so that other synthetic biologists may access the data and build off each other's work. Submitting to a repository is also important for reproducing published scientific work. The synthetic biology journal ACS Synthetic Biology now encourages authors to submit SBOL data about their genetically engineered DNA to a repository like SynBioHub or JBEI-ICE. In order to submit to a PartShop remotely, the user must first vist the appropriate website and register. Once the user has established an account, they can then log in remotely using PySBOL.

```
login("johndoe@example.org", password)
submit(doc)
```

CHAPTER 5

SBOL Examples

See Full Example Code for full example code.

5.1 Computer-aided Design with PySBOL

An advantage of the SBOL data format over GenBank is the ability to represent DNA as abstract components without specifying an exact sequence. An **abstract design** can be used as a template, with sequence information filled in later. In SBOL, a ComponentDefinition represents a biological component whose general function is known while its sequence is currently either unknown or unspecified. The intended function of the component is specified using a descriptive term from the Sequence Ontology (SO), a standard vocabulary for describing genetic parts. As the following example shows, some common SO terms are built in to PySBOL as pre-defined constants (see constants.h). This code example defines the new component as a gene by setting its *roles* property to the SO term for *gene*. Other terms may be found by browsing the Sequence Ontology online.

```
# Construct an abstract design for a gene
gene = ComponentDefinition('gene_example')
gene.roles = SO_GENE
```

Design abstraction is an important engineering principle for synthetic biology. Abstraction enables the engineer to think at a high-level about functional characteristics of a system while hiding low-level physical details. For example, in electronics, abstract schematics are used to describe the function of a circuit, while hiding the physical details of how a printed circuit board is laid out. Computer-aided design (CAD) programs allow the engineer to easily switch back and forth between abstract and physical representations of a circuit. In the same spirit, PySBOL enables a CAD approach for designing genetic constructs and other forms of synthetic biology.

5.2 Hierarchical DNA Assembly

PySBOL also includes methods for assembling biological components into **abstraction hierarchies**. This is important rom a biological perspective, because DNA sequences and biological structures in general exhibit hierarchical organization, from the genome, to operons, to genes, to lower level genetic operators. The following code assembles

an abstraction hierarchy that describes a gene cassete. Note that subcomponents must belong to a *Document* in order to be assembled, so a *Document* is passed as a parameter.

The gene cassette below is composed of genetic subcomponents including a promoter, ribosome binding site (RBS), coding sequence (CDS), and transcriptional terminator, expressed in SBOL Visual schematic glyphs. The next example demonstrates how an abstract design for this gene is assembled from its subcomponents.

```
gene.assemblePrimaryStructure([ r0010, b0032, e0040, b0012 ], doc)
```

After creating an abstraction hierarchy, it is then possible to iterate through an object's primary structure of components:

```
for component_definition in gene.getPrimaryStructure()):
    print (component_definition.identity)
```

This returns a list of *ComponentDefinitions* arranged in their primary sequence. *Caution!* It is also possible to iterate through components as follows, but this way is *not* guaranteed to return components in sequential order. This is because SBOL supports a variety of structural descriptions, not just primary structure.

```
for component in gene.components:
    print (component.definition)
```

5.3 Sequence Assembly

A **complete design** adds explicit sequence information to the components in a **template design** or **abstraction hierarchy**. In order to complete a design, *Sequence* objects must first be created and associated with the promoter, CDS, RBS, terminator subcomponents. In contrast to the ComponentDefinition.assemble() method, which assembles a template design, the *Sequence.compile* method recursively generates the complete sequence of a hierarchical design from the sequence of its subcomponents. Compiling a DNA sequence is analogous to a programmer compiling their code. *In order to compile a 'Sequence', you must first assemble a template design from 'ComponentDefinitions', as described in the previous section.*

```
gene_seq = Sequence('gene_seq')
gene_seq.sequences.set(gene_seq.identity)
gene_seq.compile()
print (gene_seq.elements)
```

5.4 Iterating through a Primary Sequence of Components

Sometimes it is desired to iterate through individual components inside a sequence of components. One application of this is to check the order of a sequence of components. To do so, one can simply implement typical forloop used in Python. The example below shows how one would iterate through a primary sequence of components to validate the correct order.

```
doc = Document()

gene = ComponentDefinition('BB0001')
promoter = ComponentDefinition('R0010')
CDS = ComponentDefinition('B0032')
RBS = ComponentDefinition('E0040')
terminator = ComponentDefinition('B0012')
```

```
doc.addComponentDefinition([gene, promoter, CDS, RBS, terminator])

gene.assemble([ promoter, RBS, CDS, terminator ])
primary_sequence = gene.getPrimaryStructure()
for component in primary_sequence:
    print(component.displayId)
```

The output is shown below, which captures the correct order.

```
R0010
E0040
B0032
B0012
```

5.5 Full Example Code

Full example code is provided below, which will create a file called "gene_cassette.xml"

```
from sbol import *

setHomespace('http://sys-bio.org')
doc = Document()

gene = ComponentDefinition('gene_example')
promoter = ComponentDefinition('R0010')
CDS = ComponentDefinition('B0032')
RBS = ComponentDefinition('E0040')
terminator = ComponentDefinition('B0012')

promoter.roles = SO_PROMOTER
CDS.roles = SO_CDS
RBS.roles = SO_RBS
terminator.roles = SO_TERMINATOR

doc.addComponentDefinition(gene)
doc.addComponentDefinition(promoter)
doc.addComponentDefinition(CDS)
doc.addComponentDefinition(RBS)
doc.addComponentDefinition(terminator)

gene.assemblePrimaryStructure([ promoter, RBS, CDS, terminator ])

first = gene.getFirstComponent()
print(first.identity)
last = gene.getLastComponent()
print(last.identity)

promoter_seq = Sequence('R0010', 'ggctgca')
RBS_seq = Sequence('B0032', 'aattatataaa')
CDS_seq = Sequence('E0040', "atgtaa")
terminator_seq = Sequence('B0012', 'attcga')
gene_seq = Sequence('BB0001')

promoter.sequence = promoter_seq
CDS.sequence = CDS_seq
```

```
RBS.sequence = RBS_seq
terminator.sequence = terminator_seq
gene.sequence = gene_seq

gene_seq.assemble()

print(promoter_seq.elements)
print(RBS_seq.elements)
print(CDS_seq.elements)
print(terminator_seq.elements)
print(gene_seq.elements)

result = doc.write('gene_cassette.xml')
print(result)
```

CHAPTER 6

Testing pySBOL

pySBOL comes with a testing function to check the integrity of the library. To run the tester, simply execute the following command.

```
import sbol
sbol.testSBOL()
```

The output tells you whether certain test has been passed or not.

```
testAddComponentDefinition (sbol.unit_tests.TestComponentDefinitions) ... ok
testCDDisplayId (sbol.unit_tests.TestComponentDefinitions) ... ok
testRemoveComponentDefinition (sbol.unit_tests.TestComponentDefinitions) ... ok
testAddSeqence (sbol.unit_tests.TestSequences) ... ok
testRemoveSequence (sbol.unit_tests.TestSequences) ... ok
testSeqDisplayId (sbol.unit_tests.TestSequences) ... ok
testSequenceElement (sbol.unit_tests.TestSequences) ... ok
testDiscard (sbol.unit_tests.TestMemory) ... ok
```

CHAPTER 7

API

class ActivityProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class AgentProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class AnalysisProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.

- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class AssociationProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class AttachmentProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class BuildProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Collection(**args*)
The Collection class is a class that groups together a set of TopLevel objects that have something in common.

Some examples of Collection objects: . Results of a query to find all ComponentDefinition objects in a repository that function as promoters . A set of ModuleDefinition objects representing a library of genetic logic gates. . A ModuleDefinition for a complexdesign, and all of the ModuleDefinition, ComponentDefinition, Sequence, and Model objects used to provide its full specification.

copy(**args*)
Copy an object and automatically increment its version.

If the optional version argument is specified, it will be used instead of incrementing the copied object's version. An object may also be copied into a new document and a new namespace, assuming compliant URIs.

- ***SBOLClass*** [] The type of SBOL object being copied
- ***new_doc*** [] The new copies will be attached to this Document. NULL by default.
- ***ns*** [] This namespace will be substituted for the current namespace (as configured by setHomespace) in all SBOL-compliat URIs.
- ***version*** [] A new version

The full URI of the created object.

class CollectionProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class CombinatorialDerivationProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Component(**args*)
The Component class is used to compose ComponentDefinition objects into a structural hierarchy. For example, the ComponentDefinition of a gene could contain four Component objects: a promoter, RBS, CDS, and terminator. In turn, the ComponentDefinition of the promoter Component could contain Component objects defined as various operator sites.

class ComponentDefinition(**args*)
The ComponentDefinition class represents the structural entities of a biological design.

The primary usage of this class is to represent structural entities with designed sequences, such as DNA, RNA, and proteins, but it can also be used to represent any other entity that is part of a design, such as small molecules, proteins, and complexes

assemble(**args*)
Assembles the provided vector of Components into a structural hierarchy.

update SequenceAnnotation starts and ends

Autoconstructs the required Components and SequenceConstraints. The resulting data structure is a partial design, still lacking a specific DNA (or other) sequence. To fully realize a design, use Sequence::assemble().

- ***list_of_components*** [] A list of subcomponents that will compose this ComponentDefinition

copy(**args*)
Copy an object and automatically increment its version.

If the optional version argument is specified, it will be used instead of incrementing the copied object's version. An object may also be copied into a new document and a new namespace, assuming compliant URIs.

- ***SBOLClass*** [] The type of SBOL object being copied
- ***new_doc*** [] The new copies will be attached to this Document. NULL by default.
- ***ns*** [] This namespace will be substituted for the current namespace (as configured by setHomespace) in all SBOL-compliat URIs.
- ***version*** [] A new version

The full URI of the created object.

getDownstreamComponent(*current_component*)
Get the downstream Component.

The downstream component

getFirstComponent()
Gets the first Component in a linear sequence.

The first component in sequential order

getInSequentialOrder()
Orders this ComponentDefinition's member Components into a linear arrangement based on Sequence Constraints.

Primary sequence structure

getLastComponent()
Gets the last Component in a linear sequence.

The last component in sequential order

getUpstreamComponent(*current_component*)
Get the upstream Component.

The upstream component

hasDownstreamComponent(*current_component*)
Checks if the specified Component has a Component downstream in linear arrangement on the DNA strand.

Checks that the appropriate SequenceConstraint exists.

- ***current_component*** [] A Component in this ComponentDefinition

1 if found, 0 if not

hasUpstreamComponent(*current_component*)
Checks if the specified Component has a Component upstream in linear arrangement on the DNA strand.

Checks that the appropriate SequenceConstraint exists.

- ***current_component*** [] A Component in this ComponentDefinition

1 if found, 0 if not

participate(*species*)
A convenience method that assigns a component to participate in a biochemical reaction.

Behind the scenes, it auto-constructs a FunctionalComponent for this ComponentDefinition and assigns it to a Participation

- ***species*** [] A Participation object (ie, participant species in a biochemical Interaction).

updateSequence(**args*)
Assemble a parent ComponentDefinition's Sequence from its subcomponent Sequences.

- ***composite_sequence*** [] A recursive parameter, use default value

The assembled parent sequence

class ComponentDefinitionProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class ComponentInstance(**args*, ***kwargs*)

class ComponentProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Config
A class which contains global configuration variables for the libSBOL environment. Intended to be used like a static class, configuration variables are accessed through the 'Config' object.

static getOption(*option*)
Get current option value for online validation and conversion.

- ***option*** [] The option key

static setOption(**args*)
Configure options for online validation and conversion Option

Description

Values

validate

Enable validation and conversion requests through the online validator

True or False

validatorURL

The http request endpoint for validation

A valid URL, set to http://www.async.ece.utah.edu/sbol-validator/endpoint.php by default

output

File format for conversion

SBOL2, SBOL1, FASTA, GenBank

diff

Report differences between two files

True or False

noncompliantUrisAllowed

If set to false, URIs in the file will not be checked for compliance with the SBOL specification

True or False

incompleteDocumentsAllowed

If set to false, not all referenced objects must be described within the given main_file

True or False

bestPracticesCheck

If set to true, the file is checked for the best practice rules set in the SBOL specification

True or False

failOnFirstError

If set to true, the validator will fail at the first error

True or False

displayFullErrorStackTrace

If set to true (and failOnFirstError is true) the validator will provide a stack trace for the first validation error

True or False

topLevelToConvert

uriPrefix

Required for conversion from FASTA and GenBank to SBOL1 or SBOL2, used to generate URIs

True or False

version

Adds the version to all URIs and to the document

A valid Maven version string

wantFileBack

Whether or not to return the file contents as a string

True or False

- ***option*** [] The option key
- ***value*** [] The option value

Config_getOption(*option*)

Get current option value for online validation and conversion.

- ***option*** [] The option key

Config_setOption(**args*)

Configure options for online validation and conversion Option

Description

Values

validate

Enable validation and conversion requests through the online validator

True or False

validatorURL

The http request endpoint for validation

A valid URL, set to http://www.async.ece.utah.edu/sbol-validator/endpoint.php by default

output

File format for conversion

SBOL2, SBOL1, FASTA, GenBank

diff

Report differences between two files

True or False

noncompliantUrisAllowed

If set to false, URIs in the file will not be checked for compliance with the SBOL specification

True or False

incompleteDocumentsAllowed

If set to false, not all referenced objects must be described within the given main_file

True or False

bestPracticesCheck

If set to true, the file is checked for the best practice rules set in the SBOL specification

True or False

failOnFirstError

If set to true, the validator will fail at the first error

True or False

displayFullErrorStackTrace

If set to true (and failOnFirstError is true) the validator will provide a stack trace for the first validation error

True or False

topLevelToConvert

uriPrefix

Required for conversion from FASTA and GenBank to SBOL1 or SBOL2, used to generate URIs

True or False

version

Adds the version to all URIs and to the document

A valid Maven version string

wantFileBack

Whether or not to return the file contents as a string

True or False

- ***option*** [] The option key
- ***value*** [] The option value

class Cut(**args*)

The Cut class specifies a location between two coordinates of a Sequence's elements.

class DesignProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Document(**args*)

Read and write SBOL using a Document class. The Document is a container for Components, Modules, and all other SBOLObjects.

addComponentDefinition(**args*)

Adds a component definition or a list of component definitions to a sbol::Document object.

- ***componentDefinition*** [] ComponentDefinition object or a list of ComponentDefinition objects

addModuleDefinition(**args*)

Adds a module definition or a list of module definitions to a sbol::Document object.

- ***moduleDefinition*** [] ModuleDefinition object or a list of ModuleDefinition objects

addNamespace(**args*)

Add a new namespace to this Document.

- ***ns*** [] The namespace, eg. http://sbols.org/v2#
- ***prefix*** [] The namespace prefix, eg. sbol

addSequence(**args*)

Adds a sequence or a list of sequences to a sbol::Document object.

- ***sequence*** [] Sequence object or a list of Sequence objects

append(*filename*)
Read an RDF/XML file and attach the SBOL objects to this Document.

New objects will be added to the existing contents of the Document

- ***filename*** [] The full name of the file you want to read (including file extension)

copy(*ns*, *doc=None*)

find(*uri*)
Search recursively for an SBOLObject in this Document that matches the uri.

- ***uri*** [] The identity of the object to search for

A pointer to the SBOLObject, or NULL if an object with this identity doesn't exist

find_property(*uri*)
Search this object recursively to see if it contains a member property with the given RDF type.

- ***uri*** [] The RDF type of the property to search for.

A pointer to the object that contains a member property with the specified RDF type, NULL otherwise

getActivity(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getAgent(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getAnalysis(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getAttachment(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getBuild(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getCollection(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve

- ***SBOLClass*** [] The type of SBOL object

getCombinatorialDerivation(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getComponentDefinition(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getDesign(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getImplementation(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getModel(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getModuleDefinition(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getNamespaces()
A vector of namespaces Get namespaces contained in this Document

getPlan(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getSampleRoster(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve

- ***SBOLClass*** [] The type of SBOL object

getSequence(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

getTest(*uri*)
Retrieve an object from the Document.

- ***uri*** [] The identity of the SBOL object you want to retrieve
- ***SBOLClass*** [] The type of SBOL object

read(*filename*)
Read an RDF/XML file and attach the SBOL objects to this Document.

Existing contents of the Document will be wiped.

- ***filename*** [] The full name of the file you want to read (including file extension)

request_validation(*sbol*)
Submit this Document to the online validator.

The validation results

validate()
Run validation on this Document.

The validation results

write(*filename*)
Serialize all objects in this Document to an RDF/XML file.

- ***filename*** [] The full name of the file you want to write (including file extension)

A string with the validation results, or empty string if validation is disabled

class FunctionalComponent(**args*)
The FunctionalComponent class is used to specify the functional usage of a ComponentDefinition inside a ModuleDefinition. The ModuleDefinition describes how the that describes how the FunctionalComponent interacts with others and summarizes their aggregate function.

connect(*interface_component*)
This method connects module inputs and outputs.

This convenience method auto-constructs a MapsTo object. See Biosystem Design for an example

- ***interface_component*** [] An input or output component from another ModuleDefinition that corresponds with this component.

isMasked()
Used to tell if a FunctionalComponent is linked to an equivalent FunctionalComponent in another ModuleDefinition.

1 if the FunctionalComponent has been over-rided by another FunctionalComponent, 0 if it hasn't.

mask(*masked_component*)
This method is used to state that FunctionalComponents in separate ModuleDefinitions are functionally equivalent.

Using this method will override the FunctionalComponent in the argument with the FunctionalComponent calling the method. This is useful for overriding a generic, template component with an explicitly defined component. This convenience method auto-constructs a MapsTo object. See Biosystem Design for an example

- ***masked_component*** [] The FunctionalComponent that is being masked (over-ridden)

class FunctionalComponentProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class GenericLocation(**args*)

the GenericLocation class is included as a starting point for specifying regions on Sequence objects with encoding properties other than IUPAC and potentially nonlinear structure. This class can also be used to set the orientation of a SequenceAnnotation and any associated Component when their parent ComponentDefinition is a partial design that lacks a Sequence.

class Identified(**args*)

All SBOL-defined classes are directly or indirectly derived from the Identified abstract class.

An Identified object is identified using a Uniform Resource Identifier (URI), a unique string that identifies and refers to a specific object in an SBOL document or in an online resource such as a DNA repository.

class ImplementationProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class IntProperty(**args*)
IntProperty objects are used to contain integers.

They can be used as member objects inside custom SBOL Extension classes.

get()
Basic getter for all SBOL literal properties.

An integer

getAll()
Retrieve a vector of objects from the IntProperty.

class Interaction(**args*)
The Interaction class provides more detailed descriptionof how the FunctionalComponents are intended to work together. For example, this class can be used to represent different forms of genetic regulation (e.g., transcriptional activation or repression), processes from the central dogma of biology (e.g. transcription and translation), and other basic molecular interactions (e.g., non-covalent binding or enzymatic phosphorylation).

class InteractionProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Location(**args*)
The Location class specifies the strand orientation of a Component and can be further extended by the Range, Cut, and GenericLocation classes.

class LocationProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class MapsTo(**args*)

The purpose of the MapsTo class is to make identity relationships between different ComponentInstances in functional and structural hierarchies more clear. For example, a MapsTo object may be used to connect outputs and inputs between different low-level ModuleDefinitions contained in a higher level Module Definition. A MapsTo object may also be used to override a generic Component in a low-level ModuleDefinition with an explicit Component in a high-level ModuleDefinition, for example mapping a generic gene to an explicit component with a name and sequence.

class MapsToProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Model(**args*)

The purpose of the Model class is to serve as a placeholder for an external computational model and provide additional meta-data to enable better reasoning about the contents of this model.

In this way, there is minimal duplication of standardization efforts and users of SBOL can formalize the function of a ModuleDefinition in the language of their choice.

copy(**args*)

Copy an object and automatically increment its version.

If the optional version argument is specified, it will be used instead of incrementing the copied object's version. An object may also be copied into a new document and a new namespace, assuming compliant URIs.

- ***SBOLClass*** [] The type of SBOL object being copied
- ***new_doc*** [] The new copies will be attached to this Document. NULL by default.

- ***ns*** [] This namespace will be substituted for the current namespace (as configured by setHomespace) in all SBOL-compliat URIs.
- ***version*** [] A new version

The full URI of the created object.

class ModelProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Module(**args*)

The Module class represents a submodule of a ModuleDefinition within a hierarchical design.

class ModuleDefinition(**args*)

The ModuleDefinition class represents a grouping of structural and functional entities in a biological design. The primary usage of this class is to assert the molecular interactions and abstract function of its child entities.

assemble(**args*)

Assemble a high-level ModuleDefinition from lower-level submodules.

Autoconstructs Module objects in the process.

- ***list_of_modules*** [] A list of pointers to the submodule ModuleDefinitions

copy(**args*)

Copy an object and automatically increment its version.

If the optional version argument is specified, it will be used instead of incrementing the copied object's version. An object may also be copied into a new document and a new namespace, assuming compliant URIs.

- ***SBOLClass*** [] The type of SBOL object being copied
- ***new_doc*** [] The new copies will be attached to this Document. NULL by default.
- ***ns*** [] This namespace will be substituted for the current namespace (as configured by setHomespace) in all SBOL-compliat URIs.
- ***version*** [] A new version

The full URI of the created object.

setInput(**args*)
Defines an input for a system module.

- ***input*** [] A ComponentDefinition that defines the input

A FunctionalComponent that is derived from the argument ComponentDefinition and configured as this ModuleDefinition's input (it's direction property is set to SBOL_DIRECTION_IN)

setOutput(**args*)
Defines an output for a system module.

- ***output*** [] A ComponentDefinition that defines the output

A FunctionalComponent that is derived from the argument ComponentDefinition and configured as this ModuleDefinition's output (it's direction property is set to SBOL_DIRECTION_OUT)

class ModuleDefinitionProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class ModuleProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class OwnedActivity(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedAgent(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedAnalysis(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedAssociation(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedAttachment(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedBuild(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedCollection(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedCombinatorialDerivation(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedComponent(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedComponentDefinition(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedDesign(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedFunctionalComponent(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedImplementation(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedInteraction(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedLocation(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedMapsTo(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedModel(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedModule(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedModuleDefinition(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedParticipation(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedPlan(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedSampleRoster(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedSequence(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedSequenceAnnotation(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedSequenceConstraint(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedTest(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedUsage(**args*)

A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)

Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()

Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.

- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll()

Retrieve a vector of objects from the OwnedObject.

getCut(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation(**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.

- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange(**args*)
Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class OwnedVariableComponent(**args*)
A container property that contains child objects.

Creates a composition out of two or more classes. In the SBOL specification, compositional relationships are indicated in class diagrams by arrows with black diamonds. A compositional relationship means that deleting the parent object will delete the child objects, and adding the parent object to a Document will also add the child object. Owned objects are stored in arbitrary order.

- ***SBOLClass*** [] The type of child SBOL object contained by this Property

add(*sbol_obj*)
Appends the new value to a list of values, for properties that allow it.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***sbol_obj*** [] A child object to add to this container property. Adds a child object to the parent object. This method always appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred

clear()
Remove all children objects from the parent and destroy them.

create(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createCut(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createGenericLocation(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

createRange(*uri*)

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this specialization for OwnedObject properties which contain multiple types of SBOLObjects.
- ***uri*** [] If SBOLCompliance is enabled, this should be the displayId for the new child object. If not enabled, this should be a full raw URI.

A reference to the child object Autoconstructs a child object and attaches it to the parent object. The new object will be constructed with default values specified in the constructor for this type of object. If SBOLCompliance is enabled, the child object's identity will be constructed using the supplied displayId argument. Otherwise, the user should supply a full URI. check uniqueness of URI in Document

get (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getAll ()

Retrieve a vector of objects from the OwnedObject.

getCut (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getGenericLocation (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

getRange (**args*)

Get the child object.

- ***SBOLClass*** [] The type of the child object
- ***SBOLSubClass*** [] A derived class of SBOLClass. Use this type specialization when adding multiple types of SBOLObjects to a container.
- ***uri*** [] The specific URI for a child object if this OwnedObject property contains multiple objects,

A reference to the child object Returns a child object from the OwnedObject property. If no URI is specified, the first object in this OwnedObject property is returned.

remove(**args*)
Remove an object from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(*sbol_obj*)
Basic setter for OwnedObject SBOL IntProperty.

- ***SBOLClass*** [] The type of SBOL object contained in this OwnedObject property
- ***sbol_obj*** [] A child object to add to this container property. Assigns a child object to this OwnedObject container property. This method always overwrites the first SBOLObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- ***sbol_obj*** [] The child object Sets the first object in the container

class Participation(**args*)
Each Participation represents how a particular FunctionalComponent behaves in its parent Interaction.

class ParticipationProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class PlanProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Range(**args*)
A Range object specifies a region via discrete, inclusive start and end positions that correspond to indices for characters in the elements String of a Sequence. Note that the index of the first location is 1, as is typical practice in biology, rather than 0, as is typical practice in computer science.

class ReferencedObject(**args*)
A reference to another SBOL object Contains a Uniform Resource Identifier (URI) that refers to an an associated object.

The object it points to may be another resource in this Document or an external reference, for example to an object in an external repository. In the SBOL specification, association by reference is indicated in class diagrams by arrows with open (white) diamonds.

add(**args*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

addReference(*uri*)

create(*uri*)
Creates another SBOL object derived from TopLevel and adds it to the Document.

- ***uri*** [] In "open world" mode, this is a full URI and the same as the returned URI. If the default namespace for libSBOL has been configured, then this argument should simply be a local identifier. If SBOL-compliance is enabled, this argument should be the intended displayId of the new object. A full URI is automatically generated and returned.

The full URI of the created object.

set(**args*)
Basic setter for SBOL ReferencedObject.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

setReference(*uri*)

class SBOLObject(**args*)
An SBOLObject converts a class data structure into an RDF triple store and contains methods for serializing and parsing RDF triples.

compare(*comparand*)
Compare two SBOL objects or Documents.

The behavior is currently undefined for objects with custom annotations or extension classes.

- ***comparand*** [] A pointer to the object being compared to this one.

1 if the objects are identical, 0 if they are different

find(*uri*)
Search this object recursively to see if an object with the URI already exists.

- ***uri*** [] The URI to search for.

A pointer to theobject with this URI if it exists, NULL otherwise

find_property(*uri*)
Search this object recursively to see if it contains a member property with the given RDF type.

- ***uri*** [] The RDF type of the property to search for.

A pointer to the object that contains a member property with the specified RDF type, NULL otherwise

getClassName(*type*)
Parses a local class name from the RDF-type of this SBOL Object

getProperties()
Gets URIs for all properties contained by this object.

This includes SBOL core properties as well as custom annotations. Use this to find custom extension data in an SBOL file.

A vector of URIs that identify the properties contained in this object

getPropertyValue(*property_uri*)
Get the value of a custom annotation property by its URI.

- ***property_uri*** [] The URI for the property

The value of the property or SBOL_ERROR_NOT_FOUND

getPropertyValues(*property_uri*)
Get all values of a custom annotation property by its URI.

- ***property_uri*** [] The URI for the property

A vector of property values or SBOL_ERROR_NOT_FOUND

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

class SampleRosterProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class Sequence(**args*)
The primary structure (eg, nucleotide or amino acid sequence) of a ComponentDefinition object.

assemble(**args*)
Calculates the complete sequence of a high-level Component from the sequence of its subcomponents.

{rior to assembling the the complete sequence, you must assemble a template design by calling ComponentDefinition::assemble for the ComponentDefinition that references this Sequence.

- ***composite_sequence*** [] Typically no value for the composite sequence should be specified by the user. This parameter is used to hold the composite sequence as it is passed to function calls at a higher-level of the recursion stack.

copy(**args*)
Copy an object and automatically increment its version.

If the optional version argument is specified, it will be used instead of incrementing the copied object's version. An object may also be copied into a new document and a new namespace, assuming compliant URIs.

- ***SBOLClass*** [] The type of SBOL object being copied
- ***new_doc*** [] The new copies will be attached to this Document. NULL by default.
- ***ns*** [] This namespace will be substituted for the current namespace (as configured by setHomespace) in all SBOL-compliat URIs.
- ***version*** [] A new version

The full URI of the created object.

class SequenceAnnotation(**args*)

The SequenceAnnotation class describes one or more regions of interest on the Sequence objects referred to by its parent ComponentDefinition. In addition, SequenceAnnotation objects can describe the substructure of their parent ComponentDefinition through association with the Component objects contained by this ComponentDefinition.

class SequenceAnnotationProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()

Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()

The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)

Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)

Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class SequenceConstraint(**args*)

The SequenceConstraint class can be used to assert restrictions on the relative, sequence-based positions of pairs of Component objects contained by the same parent ComponentDefinition. The primary purpose of this class is to enable the specification of partially designed ComponentDefinition objects, for which the precise positions or orientations of their contained Component objects are not yet fully determined.

class SequenceConstraintProperty(**args*)

Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)

Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class SequenceProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class TestProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class TextProperty(**args*)
TextProperty objects are used to contain string literals.

They can be used as member objects inside custom SBOL Extension classes.

get()
Basic getter for all SBOL literal properties.

A string literal

getAll()
Retrieve a vector of objects from the TextProperty.

class TopLevel(**args*)
All SBOL classes derived from TopLevel appear as top level nodes in the RDF/XML document tree and SBOL files. An abstract class.

class URIProperty(**args*)
A URIProperty may contain a restricted type of string that conforms to the specification for a Uniform Resource Identifier (URI), typically consisting of a namespace authority followed by an identifier.

A URIProperty often contains a reference to an SBOL object or may contain an ontology term.

get()
Basic getter for all SBOL literal properties.

A string of characters used to identify a resource

getAll()
Retrieve a vector of objects from the URIProperty.

class UsageProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class VariableComponentProperty(**args*)
Member properties of all SBOL objects are defined using a Property object.

The Property class provides a generic interface for accessing SBOL objects. At a low level, the Property class converts SBOL data structures into RDF triples.

- ***The*** [] SBOL specification currently supports string, URI, and integer literal values.

add(*new_value*)
Appends the new value to a list of values, for properties that allow it.

- ***new_value*** [] A new string which will be added to a list of values.

clear()
Remove all children objects from the parent and destroy them.

getOwner()

getTypeURI()
The uniform resource identifier that describes the RDF-type of this SBOL Object

remove(*index=0*)
Remove a Property from the list of objects and destroy it.

- ***uri*** [] The identity of the object to be destroyed. This can be a displayId of the object or a full URI may be provided.
- ***index*** [] A numerical index for the object.

set(**args*)
Basic setter for SBOL Property.

- ***new_value*** [] A new integer value for the property, which is converted to a raw string during serialization.

validate(*arg=None*)

write()

class VersionProperty(*property_owner*, *type_uri*, *lower_bound*, *upper_bound*, *initial_value*)
Contains a version number for an SBOL object.

The VersionProperty follows Maven versioning semantics and includes a major, minor, and patch version number. Specifically, libSBOL currently only supports using '.' as a delimiter (e.g.: v2.0.1). If the user does not want to follow Maven versioning, they can specify an arbitrary version string using the set() method.

decrementMajor()
Decrement major version.

decrementMinor()
Decrement major version.

decrementPatch()
Decrement major version.

incrementMajor()
Increment major version.

incrementMinor()
Increment minor version.

incrementPatch()
Increment patch version.

major()
Get major version.

The major version as an integer Splits the version string by a delimiter and returns the major version number

minor()
Get minor version.

The minor version as an integer Splits the version string by a delimiter and returns the minor version number

patch()
Get patch version.

The patch version as an integer Splits the version string by a delimiter and returns the patch version

getFileFormat()
Returns currently accepted file format.

getHomespace()
Returns the current default namespace for autocreation of URIs when a new SBOL object is created.

hasHomespace()
Checks if a valid default namespace has been defined.

setFileFormat(*file_format*)
Sets file format to use.

setHomespace(*ns*)
Sets the default namespace for autocreation of URIs when a new SBOL object is created.

- ***ns*:** Homespace

testSBOL()
Function to run test suite for pySBOL

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index

D

F

G

H

I

L

M

O

P

R

S

T

U

V

W