
pysbol2 Documentation

Release 2.3.0

Bryan Bartley

Feb 21, 2018

Contents

1	Introduction	3
2	Installation	5
2.1	Using Pip	5
2.2	Using Python	5
2.3	Using Installer for Windows	6
2.4	For Linux Users	6
3	Getting Started with SBOL	7
3.1	Creating an SBOL Document	7
3.2	Creating SBOL Data Objects	8
3.3	Adding Objects to a Document	9
3.4	Getting, Setting, and Editing Optional Fields	9
3.5	Creating and Editing Child Objects	9
3.6	Creating and Editing Reference Properties	10
3.7	Iterating and Indexing List Properties	10
4	Biological Parts Repositories	11
4.1	Mining Genetic Parts From Online Repositories	11
4.2	Searching Part Repos	12
4.3	Submitting Designs to a Repo	13
5	SBOL Examples	15
5.1	Computer-aided Design with PySBOL	15
5.2	Hierarchical DNA Assembly	15
5.3	Sequence Assembly	16
5.4	Iterating through a Primary Sequence of Components	16
5.5	Full Example Code	16
6	Testing pySBOL	19
7	API	21
8	Indices and tables	127
	Python Module Index	129

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).

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.



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.

Getting Started with SBOL

This beginner's guide introduces the basic principles of pySBOL for new users. For more comprehensive documentation about the API, refer to documentation about specific classes and methods for detailed information about the API. 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 LibSBOL is called a Document. The Document serves as a container, initially empty, for SBOL data objects. All file I/O operations are performed on the Document to populate it with SBOL objects representing design elements. Usually the first step is to create an SBOLDocument in which to put your objects. This can be done by calling the Document constructor. The read and write methods are used for reading and writing files in SBOL format.

```
doc = Document()
doc.read("CRISPR_example.xml")
print (len(Document))
doc.write("CRISPR_example.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 ComponentDefinitions from multiple files into a single design, for example.

A Document may contain different types of SBOL objects, including ComponentDefinitions, ModuleDefinitions, Sequences, SequenceAnnotations, and Models. These objects are collectively referred to as Top Level 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.

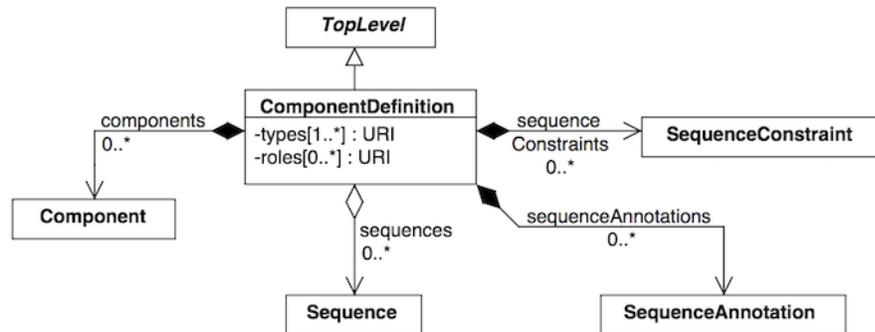
In order to review the ComponentDefinitions contained in a Document, use a Python iterator:

```
for cd in doc.componentDefinitions:
    print cd
```

This will print the unique identity of each object (see the next section). Similarly, you can iterate through `Document.moduleDefinitions()`, `Document.sequences()`, `Document.sequenceAnnotations()`, and `Document.models()`.

3.2 Creating SBOL Data Objects

Both structural and functional details of biological designs can be described with SBOL data objects. The principle classes for describing the structure and primary sequence of a design are ComponentDefinitions, Components, and Sequences, SequenceAnnotations. The principle classes for describing the function of a design are ModuleDefinitions, Modules, and Interactions. In the official SBOL specification document, these classes and their properties are represented as a special kind of box diagram. Each box represents a record of data that describe a particular kind of SBOL object. For example, following is the diagram for a ComponentDefinition which will be referred to in later sections.



When a new object is created, it must be assigned a unique identity, or uniform resource identifier (URI). A typical URI consists of a scheme, a namespace, and an identifier, although other forms of URI's are allowed. In this tutorial, we use URI's of the type `http://sys-bio.org/my_design`, where the scheme is indicated by `http://`, the namespace is `sys-bio.org` and the identifier is `my_design`.

Objects can be created by calling their respective constructors. The following constructs a ModuleDefinition:

```
crispr_template = ModuleDefinition('http://sys-bio.org/CRISPRTemplate')
```

LibSBOL provides a few global configuration options that make URI construction easy. The first configuration option allows you to specify a default namespace for new object creation. If the default namespace is set, then only an identifier needs to be passed to the constructor. This identifier will be automatically appended to the default namespace. Setting the default namespace is like signing your homework and claims ownership of an object.

```
setHomespace("http://sys-bio.org")
crispr_template = ModuleDefinition("CRISPRTemplate")
print (crispr_template.identity.get())
```

Another configuration option enables automatic construction of SBOL-compliant URIs. These URIs consist of a namespace, an identifier, AND a Maven version number. In addition, SBOL-compliance simplifies autoconstruction of certain types of SBOL objects, as we will see later. LibSBOL operates in SBOL-compliant mode by default. However, some power users will prefer to operate in “open-world” mode and provide the full raw URI when constructing objects. To disable URI construction, SBOL-compliance use `setOption('sbol_compliant_uris', 'False')`.

Some constructors have required fields. In the specification document, 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, the type, which specifies the molecular type of a component. Arguments to a constructor are always determined by whether the official SBOL specification document indicates if it is required. Required fields SHOULD be specified when calling a constructor. If they are not, then 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 DNA by default.

```
cas9 = ComponentDefinition("Cas9", BIOPAX_PROTEIN)
```

Notice the type is specified using a predefined constant. The `ComponentDefinition.types` property is one of many SBOL properties that use standard ontology terms as property values. The `ComponentDefinition.types` property uses the Sequence Ontology to be specific. Many commonly used ontological terms are provided by libSBOL as predefined constants in the `constants.h` header. See the help page for the `sbol.ComponentDefinition` class or other specific class to find a table that lists the available terms.

3.3 Adding Objects to 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 a templated add method.

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

Only TopLevel objects need to be added to a Document. These top level objects include ComponentDefinitions, ModuleDefinitions, Sequences, Models. Child objects are automatically associated with the parent object's Document.

3.4 Getting, Setting, and Editing Optional Fields

Objects may also include optional fields. These are indicated in UML as properties having a cardinality of 0 or more. Except for the molecular type field, all properties of a ComponentDefinition are optional. Optional properties can only be set after the object is created. The following code creates a DNA component which is designated as a promoter:

```
target_promoter = ComponentDefinition('TargetPromoter', BIOPAX_DNA, '1.0.0')
target_promoter.roles.set(SO_PROMOTER)
```

All properties have a set and a get method. To view the value of a property:

```
print(target_promoter.roles.get())
```

This returns the string `http://identifiers.org/so/SO:0000167` which is the Sequence Ontology term for a promoter.

Note also that some properties support a list of values. A property with a cardinality indicated by an asterisk symbol indicates that the property may hold an arbitrary number of values. For example, a ComponentDefinition may be assigned multiple roles. Calling `set` on a method always overwrites the first value of a property, while the `add` method always appends a value to the end of a list.

```
target_promoter.roles.add(SO "0000568")
```

3.5 Creating and Editing 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.annotations.add(point_mutation);
```

If you are operating in SBOL-compliant mode, you may prefer to take a shortcut:

```
target_promoter.annotations.create("PointMutation");
```

The create method captures the construction and addition of the SequenceAnnotation in a single function call. Another advantage of the create method is the construction of SBOL-compliant URIs. If operating in SBOL-compliant mode, you will almost always want to use the create method. 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. When operating in open-world mode, it is preferable to follow the first example and use the constructor and add method.

3.6 Creating and Editing Reference Properties

Some SBOL objects point to other objects by way of references. For example, ComponentDefinitions point to their corresponding Sequences. Properties of this type should be set with the URI of the related object.

```
eyfp_gene = ComponentDefinition("EYFPGene", BIOPAX_DNA);
seq = Sequence("EYFPSequence", "atggnntaa", SBOL_ENCODING_IUPAC);
eyfp_gene.sequences.set(seq.identity.get());
```

3.7 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.get())

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

Numerical indexing of lists works as well:

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

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](#).

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 visit 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)
```


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.set(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 from 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 cassette. 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.assemble([ 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.get())
```

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.get())
```

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.get())
gene_seq.compile()
print (gene_seq.elements.get())
```

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.

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

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.set(SO_PROMOTER)
CDS.roles.set(SO_CDS)
RBS.roles.set(SO_RBS)
terminator.roles.set(SO_TERMINATOR)

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

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

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

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

doc.addSequence([promoter_seq, CDS_seq, RBS_seq, terminator_seq, gene_seq])

promoter.sequences.set(promoter_seq.identity.get())
CDS.sequences.set(CDS_seq.identity.get())
RBS.sequences.set(RBS_seq.identity.get())
terminator.sequences.set(terminator_seq.identity.get())
gene.sequences.set(gene_seq.identity.get())

gene_seq.assemble()

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

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

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
testAddSequence (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
```


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 setHomepage) in all SBOL-compliant 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-compliant 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-riden 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 description of 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-compliant 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 setHomepage) in all SBOL-compliant 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- **SBOLClass** [] The type of SBOL object contained in this OwnedObject property
- **sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- **sbo_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 (sbo_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 SBOObjects to a container.
- **sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects 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 SBOObject 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- *SBOLClass* [] The type of SBOL object contained in this OwnedObject property
- *sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- *sbo_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 (sbo_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 SBOObjects to a container.
- *sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- *SBOLClass* [] The type of SBOL object contained in this OwnedObject property
- *sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- *sbo_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 (sbo_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 SBOObjects to a container.
- *sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- *SBOLClass* [] The type of SBOL object contained in this OwnedObject property
- *sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- *sbo_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 (sbo_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 SBOObjects to a container.
- *sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- **SBOLClass** [] The type of SBOL object contained in this OwnedObject property
- **sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- **sbo_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 (sbo_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 SBOObjects to a container.
- **sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- **SBOLClass** [] The type of SBOL object contained in this OwnedObject property
- **sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- **sbo_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 (sbo_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 SBOObjects to a container.
- **sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObject 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 SBOObjects.
- **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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 (sbo_obj)

Basic setter for OwnedObject SBOL IntProperty.

- *SBOLClass* [] The type of SBOL object contained in this OwnedObject property
- *sbo_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 SBOObject in the container. appends another object to those already contained in this OwnedObject property. In SBOLCompliant mode, the create method is preferred
- *sbo_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 (sbo_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 SBOObjects to a container.
- *sbo_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 SBOObjects 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 SBOObjects 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 SBOObjects 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 SBOObjects 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 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 the object 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-compliant 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

Indices and tables

- genindex
- modindex
- search



S

`sbol.libsbol`, 21

A

ActivityProperty (class in sbol.libsbol), 21
add() (ActivityProperty method), 21
add() (AgentProperty method), 22
add() (AnalysisProperty method), 22
add() (AssociationProperty method), 23
add() (AttachmentProperty method), 23
add() (BuildProperty method), 24
add() (CollectionProperty method), 25
add() (CombinatorialDerivationProperty method), 26
add() (ComponentDefinitionProperty method), 28
add() (ComponentProperty method), 28
add() (DesignProperty method), 32
add() (FunctionalComponentProperty method), 36
add() (ImplementationProperty method), 36
add() (InteractionProperty method), 37
add() (LocationProperty method), 38
add() (MapsToProperty method), 39
add() (ModelProperty method), 40
add() (ModuleDefinitionProperty method), 41
add() (ModuleProperty method), 42
add() (OwnedActivity method), 42
add() (OwnedAgent method), 45
add() (OwnedAnalysis method), 48
add() (OwnedAssociation method), 50
add() (OwnedAttachment method), 53
add() (OwnedBuild method), 56
add() (OwnedCollection method), 58
add() (OwnedCombinatorialDerivation method), 61
add() (OwnedComponent method), 64
add() (OwnedComponentDefinition method), 66
add() (OwnedDesign method), 69
add() (OwnedFunctionalComponent method), 72
add() (OwnedImplementation method), 74
add() (OwnedInteraction method), 77
add() (OwnedLocation method), 80
add() (OwnedMapsTo method), 82
add() (OwnedModel method), 85
add() (OwnedModule method), 88
add() (OwnedModuleDefinition method), 90
add() (OwnedParticipation method), 93
add() (OwnedPlan method), 96
add() (OwnedSampleRoster method), 98
add() (OwnedSequence method), 101
add() (OwnedSequenceAnnotation method), 104
add() (OwnedSequenceConstraint method), 106
add() (OwnedTest method), 109
add() (OwnedUsage method), 112
add() (OwnedVariableComponent method), 114
add() (ParticipationProperty method), 117
add() (PlanProperty method), 118
add() (ReferencedObject method), 118
add() (SampleRosterProperty method), 120
add() (SequenceAnnotationProperty method), 121
add() (SequenceConstraintProperty method), 121
add() (SequenceProperty method), 122
add() (TestProperty method), 123
add() (UsageProperty method), 124
add() (VariableComponentProperty method), 124
addComponentDefinition() (Document method), 32
addModuleDefinition() (Document method), 32
addNamespace() (Document method), 32
addReference() (ReferencedObject method), 118
addSequence() (Document method), 32
AgentProperty (class in sbol.libsbol), 21
AnalysisProperty (class in sbol.libsbol), 22
append() (Document method), 32
assemble() (ComponentDefinition method), 26
assemble() (ModuleDefinition method), 40
assemble() (Sequence method), 120
AssociationProperty (class in sbol.libsbol), 23
AttachmentProperty (class in sbol.libsbol), 23

B

BuildProperty (class in sbol.libsbol), 24

C

clear() (ActivityProperty method), 21

- clear() (AgentProperty method), 22
- clear() (AnalysisProperty method), 22
- clear() (AssociationProperty method), 23
- clear() (AttachmentProperty method), 23
- clear() (BuildProperty method), 24
- clear() (CollectionProperty method), 25
- clear() (CombinatorialDerivationProperty method), 26
- clear() (ComponentDefinitionProperty method), 28
- clear() (ComponentProperty method), 28
- clear() (DesignProperty method), 32
- clear() (FunctionalComponentProperty method), 36
- clear() (ImplementationProperty method), 37
- clear() (InteractionProperty method), 37
- clear() (LocationProperty method), 38
- clear() (MapsToProperty method), 39
- clear() (ModelProperty method), 40
- clear() (ModuleDefinitionProperty method), 41
- clear() (ModuleProperty method), 42
- clear() (OwnedActivity method), 42
- clear() (OwnedAgent method), 45
- clear() (OwnedAnalysis method), 48
- clear() (OwnedAssociation method), 50
- clear() (OwnedAttachment method), 53
- clear() (OwnedBuild method), 56
- clear() (OwnedCollection method), 58
- clear() (OwnedCombinatorialDerivation method), 61
- clear() (OwnedComponent method), 64
- clear() (OwnedComponentDefinition method), 66
- clear() (OwnedDesign method), 69
- clear() (OwnedFunctionalComponent method), 72
- clear() (OwnedImplementation method), 74
- clear() (OwnedInteraction method), 77
- clear() (OwnedLocation method), 80
- clear() (OwnedMapsTo method), 82
- clear() (OwnedModel method), 85
- clear() (OwnedModule method), 88
- clear() (OwnedModuleDefinition method), 90
- clear() (OwnedParticipation method), 93
- clear() (OwnedPlan method), 96
- clear() (OwnedSampleRoster method), 98
- clear() (OwnedSequence method), 101
- clear() (OwnedSequenceAnnotation method), 104
- clear() (OwnedSequenceConstraint method), 106
- clear() (OwnedTest method), 109
- clear() (OwnedUsage method), 112
- clear() (OwnedVariableComponent method), 114
- clear() (ParticipationProperty method), 117
- clear() (PlanProperty method), 118
- clear() (SampleRosterProperty method), 120
- clear() (SequenceAnnotationProperty method), 121
- clear() (SequenceConstraintProperty method), 121
- clear() (SequenceProperty method), 122
- clear() (TestProperty method), 123
- clear() (UsageProperty method), 124
- clear() (VariableComponentProperty method), 124
- Collection (class in sbol.libsbol), 24
- CollectionProperty (class in sbol.libsbol), 25
- CombinatorialDerivationProperty (class in sbol.libsbol), 25
- compare() (SBOLObject method), 119
- Component (class in sbol.libsbol), 26
- ComponentDefinition (class in sbol.libsbol), 26
- ComponentDefinitionProperty (class in sbol.libsbol), 28
- ComponentInstance (class in sbol.libsbol), 28
- ComponentProperty (class in sbol.libsbol), 28
- Config (class in sbol.libsbol), 29
- Config_getOption() (in module sbol.libsbol), 30
- Config_setOption() (in module sbol.libsbol), 30
- connect() (FunctionalComponent method), 35
- copy() (Collection method), 25
- copy() (ComponentDefinition method), 26
- copy() (Document method), 33
- copy() (Model method), 39
- copy() (ModuleDefinition method), 40
- copy() (Sequence method), 120
- create() (OwnedActivity method), 43
- create() (OwnedAgent method), 45
- create() (OwnedAnalysis method), 48
- create() (OwnedAssociation method), 51
- create() (OwnedAttachment method), 53
- create() (OwnedBuild method), 56
- create() (OwnedCollection method), 59
- create() (OwnedCombinatorialDerivation method), 61
- create() (OwnedComponent method), 64
- create() (OwnedComponentDefinition method), 67
- create() (OwnedDesign method), 69
- create() (OwnedFunctionalComponent method), 72
- create() (OwnedImplementation method), 75
- create() (OwnedInteraction method), 77
- create() (OwnedLocation method), 80
- create() (OwnedMapsTo method), 83
- create() (OwnedModel method), 85
- create() (OwnedModule method), 88
- create() (OwnedModuleDefinition method), 91
- create() (OwnedParticipation method), 93
- create() (OwnedPlan method), 96
- create() (OwnedSampleRoster method), 99
- create() (OwnedSequence method), 101
- create() (OwnedSequenceAnnotation method), 104
- create() (OwnedSequenceConstraint method), 107
- create() (OwnedTest method), 109
- create() (OwnedUsage method), 112
- create() (OwnedVariableComponent method), 115
- create() (ReferencedObject method), 118
- createCut() (OwnedActivity method), 43
- createCut() (OwnedAgent method), 45
- createCut() (OwnedAnalysis method), 48
- createCut() (OwnedAssociation method), 51

- createCut() (OwnedAttachment method), 53
 createCut() (OwnedBuild method), 56
 createCut() (OwnedCollection method), 59
 createCut() (OwnedCombinatorialDerivation method), 61
 createCut() (OwnedComponent method), 64
 createCut() (OwnedComponentDefinition method), 67
 createCut() (OwnedDesign method), 69
 createCut() (OwnedFunctionalComponent method), 72
 createCut() (OwnedImplementation method), 75
 createCut() (OwnedInteraction method), 77
 createCut() (OwnedLocation method), 80
 createCut() (OwnedMapsTo method), 83
 createCut() (OwnedModel method), 85
 createCut() (OwnedModule method), 88
 createCut() (OwnedModuleDefinition method), 91
 createCut() (OwnedParticipation method), 93
 createCut() (OwnedPlan method), 96
 createCut() (OwnedSampleRoster method), 99
 createCut() (OwnedSequence method), 101
 createCut() (OwnedSequenceAnnotation method), 104
 createCut() (OwnedSequenceConstraint method), 107
 createCut() (OwnedTest method), 109
 createCut() (OwnedUsage method), 112
 createCut() (OwnedVariableComponent method), 115
 createGenericLocation() (OwnedActivity method), 43
 createGenericLocation() (OwnedAgent method), 46
 createGenericLocation() (OwnedAnalysis method), 48
 createGenericLocation() (OwnedAssociation method), 51
 createGenericLocation() (OwnedAttachment method), 54
 createGenericLocation() (OwnedBuild method), 56
 createGenericLocation() (OwnedCollection method), 59
 createGenericLocation() (OwnedCombinatorialDerivation method), 62
 createGenericLocation() (OwnedComponent method), 64
 createGenericLocation() (OwnedComponentDefinition method), 67
 createGenericLocation() (OwnedDesign method), 70
 createGenericLocation() (OwnedFunctionalComponent method), 72
 createGenericLocation() (OwnedImplementation method), 75
 createGenericLocation() (OwnedInteraction method), 78
 createGenericLocation() (OwnedLocation method), 80
 createGenericLocation() (OwnedMapsTo method), 83
 createGenericLocation() (OwnedModel method), 86
 createGenericLocation() (OwnedModule method), 88
 createGenericLocation() (OwnedModuleDefinition method), 91
 createGenericLocation() (OwnedParticipation method), 94
 createGenericLocation() (OwnedPlan method), 96
 createGenericLocation() (OwnedSampleRoster method), 99
 createGenericLocation() (OwnedSequence method), 102
 createGenericLocation() (OwnedSequenceAnnotation method), 104
 createGenericLocation() (OwnedSequenceConstraint method), 107
 createGenericLocation() (OwnedTest method), 110
 createGenericLocation() (OwnedUsage method), 112
 createGenericLocation() (OwnedVariableComponent method), 115
 createRange() (OwnedActivity method), 43
 createRange() (OwnedAgent method), 46
 createRange() (OwnedAnalysis method), 49
 createRange() (OwnedAssociation method), 51
 createRange() (OwnedAttachment method), 54
 createRange() (OwnedBuild method), 57
 createRange() (OwnedCollection method), 59
 createRange() (OwnedCombinatorialDerivation method), 62
 createRange() (OwnedComponent method), 65
 createRange() (OwnedComponentDefinition method), 67
 createRange() (OwnedDesign method), 70
 createRange() (OwnedFunctionalComponent method), 73
 createRange() (OwnedImplementation method), 75
 createRange() (OwnedInteraction method), 78
 createRange() (OwnedLocation method), 81
 createRange() (OwnedMapsTo method), 83
 createRange() (OwnedModel method), 86
 createRange() (OwnedModule method), 89
 createRange() (OwnedModuleDefinition method), 91
 createRange() (OwnedParticipation method), 94
 createRange() (OwnedPlan method), 97
 createRange() (OwnedSampleRoster method), 99
 createRange() (OwnedSequence method), 102
 createRange() (OwnedSequenceAnnotation method), 105
 createRange() (OwnedSequenceConstraint method), 107
 createRange() (OwnedTest method), 110
 createRange() (OwnedUsage method), 113
 createRange() (OwnedVariableComponent method), 115
 Cut (class in sbol.libsbol), 31
- ## D
- decrementMajor() (VersionProperty method), 125
 decrementMinor() (VersionProperty method), 125
 decrementPatch() (VersionProperty method), 125
 DesignProperty (class in sbol.libsbol), 31
 Document (class in sbol.libsbol), 32
- ## F
- find() (Document method), 33
 find() (SBOLObject method), 119
 find_property() (Document method), 33
 find_property() (SBOLObject method), 119
 FunctionalComponent (class in sbol.libsbol), 35
 FunctionalComponentProperty (class in sbol.libsbol), 36

G

- GenericLocation (class in sbol.libsbol), 36
- get() (IntProperty method), 37
- get() (OwnedActivity method), 44
- get() (OwnedAgent method), 46
- get() (OwnedAnalysis method), 49
- get() (OwnedAssociation method), 52
- get() (OwnedAttachment method), 54
- get() (OwnedBuild method), 57
- get() (OwnedCollection method), 60
- get() (OwnedCombinatorialDerivation method), 62
- get() (OwnedComponent method), 65
- get() (OwnedComponentDefinition method), 68
- get() (OwnedDesign method), 70
- get() (OwnedFunctionalComponent method), 73
- get() (OwnedImplementation method), 76
- get() (OwnedInteraction method), 78
- get() (OwnedLocation method), 81
- get() (OwnedMapsTo method), 84
- get() (OwnedModel method), 86
- get() (OwnedModule method), 89
- get() (OwnedModuleDefinition method), 92
- get() (OwnedParticipation method), 94
- get() (OwnedPlan method), 97
- get() (OwnedSampleRoster method), 100
- get() (OwnedSequence method), 102
- get() (OwnedSequenceAnnotation method), 105
- get() (OwnedSequenceConstraint method), 108
- get() (OwnedTest method), 110
- get() (OwnedUsage method), 113
- get() (OwnedVariableComponent method), 116
- get() (TextProperty method), 123
- get() (URIProperty method), 123
- getActivity() (Document method), 33
- getAgent() (Document method), 33
- getAll() (IntProperty method), 37
- getAll() (OwnedActivity method), 44
- getAll() (OwnedAgent method), 46
- getAll() (OwnedAnalysis method), 49
- getAll() (OwnedAssociation method), 52
- getAll() (OwnedAttachment method), 54
- getAll() (OwnedBuild method), 57
- getAll() (OwnedCollection method), 60
- getAll() (OwnedCombinatorialDerivation method), 62
- getAll() (OwnedComponent method), 65
- getAll() (OwnedComponentDefinition method), 68
- getAll() (OwnedDesign method), 70
- getAll() (OwnedFunctionalComponent method), 73
- getAll() (OwnedImplementation method), 76
- getAll() (OwnedInteraction method), 78
- getAll() (OwnedLocation method), 81
- getAll() (OwnedMapsTo method), 84
- getAll() (OwnedModel method), 86
- getAll() (OwnedModule method), 89
- getAll() (OwnedModuleDefinition method), 92
- getAll() (OwnedParticipation method), 94
- getAll() (OwnedPlan method), 97
- getAll() (OwnedSampleRoster method), 100
- getAll() (OwnedSequence method), 102
- getAll() (OwnedSequenceAnnotation method), 105
- getAll() (OwnedSequenceConstraint method), 108
- getAll() (OwnedTest method), 110
- getAll() (OwnedUsage method), 113
- getAll() (OwnedVariableComponent method), 116
- getDesign() (Document method), 34
- getDownstreamComponent() (ComponentDefinition method), 27
- getFileFormat() (in module sbol.libsbol), 125
- getFirstComponent() (ComponentDefinition method), 27
- getGenericLocation() (OwnedActivity method), 44
- getGenericLocation() (OwnedAgent method), 47
- getAll() (OwnedModuleDefinition method), 92
- getAll() (OwnedParticipation method), 94
- getAll() (OwnedPlan method), 97
- getAll() (OwnedSampleRoster method), 100
- getAll() (OwnedSequence method), 102
- getAll() (OwnedSequenceAnnotation method), 105
- getAll() (OwnedSequenceConstraint method), 108
- getAll() (OwnedTest method), 110
- getAll() (OwnedUsage method), 113
- getAll() (OwnedVariableComponent method), 116
- getAll() (TextProperty method), 123
- getAll() (URIProperty method), 123
- getAnalysis() (Document method), 33
- getAttachment() (Document method), 33
- getBuild() (Document method), 33
- getClassName() (SBOObject method), 119
- getCollection() (Document method), 33
- getCombinatorialDerivation() (Document method), 34
- getComponentDefinition() (Document method), 34
- getCut() (OwnedActivity method), 44
- getCut() (OwnedAgent method), 46
- getCut() (OwnedAnalysis method), 49
- getCut() (OwnedAssociation method), 52
- getCut() (OwnedAttachment method), 54
- getCut() (OwnedBuild method), 57
- getCut() (OwnedCollection method), 60
- getCut() (OwnedCombinatorialDerivation method), 62
- getCut() (OwnedComponent method), 65
- getCut() (OwnedComponentDefinition method), 68
- getCut() (OwnedDesign method), 70
- getCut() (OwnedFunctionalComponent method), 73
- getCut() (OwnedImplementation method), 76
- getCut() (OwnedInteraction method), 78
- getCut() (OwnedLocation method), 81
- getCut() (OwnedMapsTo method), 84
- getCut() (OwnedModel method), 86
- getCut() (OwnedModule method), 89
- getCut() (OwnedModuleDefinition method), 92
- getCut() (OwnedParticipation method), 94
- getCut() (OwnedPlan method), 97
- getCut() (OwnedSampleRoster method), 100
- getCut() (OwnedSequence method), 102
- getCut() (OwnedSequenceAnnotation method), 105
- getCut() (OwnedSequenceConstraint method), 108
- getCut() (OwnedTest method), 110
- getCut() (OwnedUsage method), 113
- getCut() (OwnedVariableComponent method), 116

- getGenericLocation() (OwnedAnalysis method), 49
- getGenericLocation() (OwnedAssociation method), 52
- getGenericLocation() (OwnedAttachment method), 55
- getGenericLocation() (OwnedBuild method), 57
- getGenericLocation() (OwnedCollection method), 60
- getGenericLocation() (OwnedCombinatorialDerivation method), 63
- getGenericLocation() (OwnedComponent method), 65
- getGenericLocation() (OwnedComponentDefinition method), 68
- getGenericLocation() (OwnedDesign method), 71
- getGenericLocation() (OwnedFunctionalComponent method), 73
- getGenericLocation() (OwnedImplementation method), 76
- getGenericLocation() (OwnedInteraction method), 79
- getGenericLocation() (OwnedLocation method), 81
- getGenericLocation() (OwnedMapsTo method), 84
- getGenericLocation() (OwnedModel method), 87
- getGenericLocation() (OwnedModule method), 89
- getGenericLocation() (OwnedModuleDefinition method), 92
- getGenericLocation() (OwnedParticipation method), 95
- getGenericLocation() (OwnedPlan method), 97
- getGenericLocation() (OwnedSampleRoster method), 100
- getGenericLocation() (OwnedSequence method), 103
- getGenericLocation() (OwnedSequenceAnnotation method), 105
- getGenericLocation() (OwnedSequenceConstraint method), 108
- getGenericLocation() (OwnedTest method), 111
- getGenericLocation() (OwnedUsage method), 113
- getGenericLocation() (OwnedVariableComponent method), 116
- getHomespace() (in module sbol.libsbol), 125
- getImplementation() (Document method), 34
- getInSequentialOrder() (ComponentDefinition method), 27
- getLastComponent() (ComponentDefinition method), 27
- getModel() (Document method), 34
- getModuleDefinition() (Document method), 34
- getNamespaces() (Document method), 34
- getOption() (Config static method), 29
- getOwner() (ActivityProperty method), 21
- getOwner() (AgentProperty method), 22
- getOwner() (AnalysisProperty method), 22
- getOwner() (AssociationProperty method), 23
- getOwner() (AttachmentProperty method), 23
- getOwner() (BuildProperty method), 24
- getOwner() (CollectionProperty method), 25
- getOwner() (CombinatorialDerivationProperty method), 26
- getOwner() (ComponentDefinitionProperty method), 28
- getOwner() (ComponentProperty method), 28
- getOwner() (DesignProperty method), 32
- getOwner() (FunctionalComponentProperty method), 36
- getOwner() (ImplementationProperty method), 37
- getOwner() (InteractionProperty method), 37
- getOwner() (LocationProperty method), 38
- getOwner() (MapsToProperty method), 39
- getOwner() (ModelProperty method), 40
- getOwner() (ModuleDefinitionProperty method), 41
- getOwner() (ModuleProperty method), 42
- getOwner() (ParticipationProperty method), 117
- getOwner() (PlanProperty method), 118
- getOwner() (SampleRosterProperty method), 120
- getOwner() (SequenceAnnotationProperty method), 121
- getOwner() (SequenceConstraintProperty method), 122
- getOwner() (SequenceProperty method), 122
- getOwner() (TestProperty method), 123
- getOwner() (UsageProperty method), 124
- getOwner() (VariableComponentProperty method), 124
- getPlan() (Document method), 34
- getProperties() (SBOLObject method), 119
- getPropertyValue() (SBOLObject method), 119
- getPropertyValues() (SBOLObject method), 119
- getRange() (OwnedActivity method), 44
- getRange() (OwnedAgent method), 47
- getRange() (OwnedAnalysis method), 50
- getRange() (OwnedAssociation method), 52
- getRange() (OwnedAttachment method), 55
- getRange() (OwnedBuild method), 58
- getRange() (OwnedCollection method), 60
- getRange() (OwnedCombinatorialDerivation method), 63
- getRange() (OwnedComponent method), 66
- getRange() (OwnedComponentDefinition method), 68
- getRange() (OwnedDesign method), 71
- getRange() (OwnedFunctionalComponent method), 74
- getRange() (OwnedImplementation method), 76
- getRange() (OwnedInteraction method), 79
- getRange() (OwnedLocation method), 82
- getRange() (OwnedMapsTo method), 84
- getRange() (OwnedModel method), 87
- getRange() (OwnedModule method), 90
- getRange() (OwnedModuleDefinition method), 92
- getRange() (OwnedParticipation method), 95
- getRange() (OwnedPlan method), 98
- getRange() (OwnedSampleRoster method), 100
- getRange() (OwnedSequence method), 103
- getRange() (OwnedSequenceAnnotation method), 106
- getRange() (OwnedSequenceConstraint method), 108
- getRange() (OwnedTest method), 111
- getRange() (OwnedUsage method), 114
- getRange() (OwnedVariableComponent method), 116
- getSampleRoster() (Document method), 34
- getSequence() (Document method), 35
- getTest() (Document method), 35

getTypeURI() (ActivityProperty method), 21
 getTypeURI() (AgentProperty method), 22
 getTypeURI() (AnalysisProperty method), 22
 getTypeURI() (AssociationProperty method), 23
 getTypeURI() (AttachmentProperty method), 24
 getTypeURI() (BuildProperty method), 24
 getTypeURI() (CollectionProperty method), 25
 getTypeURI() (CombinatorialDerivationProperty method), 26
 getTypeURI() (ComponentDefinitionProperty method), 28
 getTypeURI() (ComponentProperty method), 28
 getTypeURI() (DesignProperty method), 32
 getTypeURI() (FunctionalComponentProperty method), 36
 getTypeURI() (ImplementationProperty method), 37
 getTypeURI() (InteractionProperty method), 38
 getTypeURI() (LocationProperty method), 38
 getTypeURI() (MapsToProperty method), 39
 getTypeURI() (ModelProperty method), 40
 getTypeURI() (ModuleDefinitionProperty method), 41
 getTypeURI() (ModuleProperty method), 42
 getTypeURI() (ParticipationProperty method), 117
 getTypeURI() (PlanProperty method), 118
 getTypeURI() (SampleRosterProperty method), 120
 getTypeURI() (SBOLObject method), 119
 getTypeURI() (SequenceAnnotationProperty method), 121
 getTypeURI() (SequenceConstraintProperty method), 122
 getTypeURI() (SequenceProperty method), 122
 getTypeURI() (TestProperty method), 123
 getTypeURI() (UsageProperty method), 124
 getTypeURI() (VariableComponentProperty method), 124
 getUpstreamComponent() (ComponentDefinition method), 27

H

hasDownstreamComponent() (ComponentDefinition method), 27
 hasHomepage() (in module sbol.libsbol), 125
 hasUpstreamComponent() (ComponentDefinition method), 27

I

Identified (class in sbol.libsbol), 36
 ImplementationProperty (class in sbol.libsbol), 36
 incrementMajor() (VersionProperty method), 125
 incrementMinor() (VersionProperty method), 125
 incrementPatch() (VersionProperty method), 125
 Interaction (class in sbol.libsbol), 37
 InteractionProperty (class in sbol.libsbol), 37
 IntProperty (class in sbol.libsbol), 37

isMasked() (FunctionalComponent method), 35

L

Location (class in sbol.libsbol), 38
 LocationProperty (class in sbol.libsbol), 38

M

major() (VersionProperty method), 125
 MapsTo (class in sbol.libsbol), 38
 MapsToProperty (class in sbol.libsbol), 39
 mask() (FunctionalComponent method), 35
 minor() (VersionProperty method), 125
 Model (class in sbol.libsbol), 39
 ModelProperty (class in sbol.libsbol), 40
 Module (class in sbol.libsbol), 40
 ModuleDefinition (class in sbol.libsbol), 40
 ModuleDefinitionProperty (class in sbol.libsbol), 41
 ModuleProperty (class in sbol.libsbol), 42

O

OwnedActivity (class in sbol.libsbol), 42
 OwnedAgent (class in sbol.libsbol), 45
 OwnedAnalysis (class in sbol.libsbol), 47
 OwnedAssociation (class in sbol.libsbol), 50
 OwnedAttachment (class in sbol.libsbol), 53
 OwnedBuild (class in sbol.libsbol), 55
 OwnedCollection (class in sbol.libsbol), 58
 OwnedCombinatorialDerivation (class in sbol.libsbol), 61
 OwnedComponent (class in sbol.libsbol), 63
 OwnedComponentDefinition (class in sbol.libsbol), 66
 OwnedDesign (class in sbol.libsbol), 69
 OwnedFunctionalComponent (class in sbol.libsbol), 71
 OwnedImplementation (class in sbol.libsbol), 74
 OwnedInteraction (class in sbol.libsbol), 77
 OwnedLocation (class in sbol.libsbol), 79
 OwnedMapsTo (class in sbol.libsbol), 82
 OwnedModel (class in sbol.libsbol), 85
 OwnedModule (class in sbol.libsbol), 87
 OwnedModuleDefinition (class in sbol.libsbol), 90
 OwnedParticipation (class in sbol.libsbol), 93
 OwnedPlan (class in sbol.libsbol), 95
 OwnedSampleRoster (class in sbol.libsbol), 98
 OwnedSequence (class in sbol.libsbol), 101
 OwnedSequenceAnnotation (class in sbol.libsbol), 103
 OwnedSequenceConstraint (class in sbol.libsbol), 106
 OwnedTest (class in sbol.libsbol), 109
 OwnedUsage (class in sbol.libsbol), 111
 OwnedVariableComponent (class in sbol.libsbol), 114

P

participate() (ComponentDefinition method), 27
 Participation (class in sbol.libsbol), 117
 ParticipationProperty (class in sbol.libsbol), 117

patch() (VersionProperty method), 125
 PlanProperty (class in sbol.libsbol), 117

R

Range (class in sbol.libsbol), 118
 read() (Document method), 35
 ReferencedObject (class in sbol.libsbol), 118
 remove() (ActivityProperty method), 21
 remove() (AgentProperty method), 22
 remove() (AnalysisProperty method), 22
 remove() (AssociationProperty method), 23
 remove() (AttachmentProperty method), 24
 remove() (BuildProperty method), 24
 remove() (CollectionProperty method), 25
 remove() (CombinatorialDerivationProperty method), 26
 remove() (ComponentDefinitionProperty method), 28
 remove() (ComponentProperty method), 29
 remove() (DesignProperty method), 32
 remove() (FunctionalComponentProperty method), 36
 remove() (ImplementationProperty method), 37
 remove() (InteractionProperty method), 38
 remove() (LocationProperty method), 38
 remove() (MapsToProperty method), 39
 remove() (ModelProperty method), 40
 remove() (ModuleDefinitionProperty method), 41
 remove() (ModuleProperty method), 42
 remove() (OwnedActivity method), 44
 remove() (OwnedAgent method), 47
 remove() (OwnedAnalysis method), 50
 remove() (OwnedAssociation method), 52
 remove() (OwnedAttachment method), 55
 remove() (OwnedBuild method), 58
 remove() (OwnedCollection method), 60
 remove() (OwnedCombinatorialDerivation method), 63
 remove() (OwnedComponent method), 66
 remove() (OwnedComponentDefinition method), 68
 remove() (OwnedDesign method), 71
 remove() (OwnedFunctionalComponent method), 74
 remove() (OwnedImplementation method), 76
 remove() (OwnedInteraction method), 79
 remove() (OwnedLocation method), 82
 remove() (OwnedMapsTo method), 84
 remove() (OwnedModel method), 87
 remove() (OwnedModule method), 90
 remove() (OwnedModuleDefinition method), 92
 remove() (OwnedParticipation method), 95
 remove() (OwnedPlan method), 98
 remove() (OwnedSampleRoster method), 100
 remove() (OwnedSequence method), 103
 remove() (OwnedSequenceAnnotation method), 106
 remove() (OwnedSequenceConstraint method), 108
 remove() (OwnedTest method), 111
 remove() (OwnedUsage method), 114
 remove() (OwnedVariableComponent method), 116

remove() (ParticipationProperty method), 117
 remove() (PlanProperty method), 118
 remove() (SampleRosterProperty method), 120
 remove() (SequenceAnnotationProperty method), 121
 remove() (SequenceConstraintProperty method), 122
 remove() (SequenceProperty method), 122
 remove() (TestProperty method), 123
 remove() (UsageProperty method), 124
 remove() (VariableComponentProperty method), 124
 request_validation() (Document method), 35

S

SampleRosterProperty (class in sbol.libsbol), 119
 sbol.libsbol (module), 21
 SBOLObject (class in sbol.libsbol), 119
 Sequence (class in sbol.libsbol), 120
 SequenceAnnotation (class in sbol.libsbol), 121
 SequenceAnnotationProperty (class in sbol.libsbol), 121
 SequenceConstraint (class in sbol.libsbol), 121
 SequenceConstraintProperty (class in sbol.libsbol), 121
 SequenceProperty (class in sbol.libsbol), 122
 set() (ActivityProperty method), 21
 set() (AgentProperty method), 22
 set() (AnalysisProperty method), 23
 set() (AssociationProperty method), 23
 set() (AttachmentProperty method), 24
 set() (BuildProperty method), 24
 set() (CollectionProperty method), 25
 set() (CombinatorialDerivationProperty method), 26
 set() (ComponentDefinitionProperty method), 28
 set() (ComponentProperty method), 29
 set() (DesignProperty method), 32
 set() (FunctionalComponentProperty method), 36
 set() (ImplementationProperty method), 37
 set() (InteractionProperty method), 38
 set() (LocationProperty method), 38
 set() (MapsToProperty method), 39
 set() (ModelProperty method), 40
 set() (ModuleDefinitionProperty method), 41
 set() (ModuleProperty method), 42
 set() (OwnedActivity method), 45
 set() (OwnedAgent method), 47
 set() (OwnedAnalysis method), 50
 set() (OwnedAssociation method), 53
 set() (OwnedAttachment method), 55
 set() (OwnedBuild method), 58
 set() (OwnedCollection method), 61
 set() (OwnedCombinatorialDerivation method), 63
 set() (OwnedComponent method), 66
 set() (OwnedComponentDefinition method), 69
 set() (OwnedDesign method), 71
 set() (OwnedFunctionalComponent method), 74
 set() (OwnedImplementation method), 77
 set() (OwnedInteraction method), 79

set() (OwnedLocation method), 82
 set() (OwnedMapsTo method), 85
 set() (OwnedModel method), 87
 set() (OwnedModule method), 90
 set() (OwnedModuleDefinition method), 93
 set() (OwnedParticipation method), 95
 set() (OwnedPlan method), 98
 set() (OwnedSampleRoster method), 101
 set() (OwnedSequence method), 103
 set() (OwnedSequenceAnnotation method), 106
 set() (OwnedSequenceConstraint method), 109
 set() (OwnedTest method), 111
 set() (OwnedUsage method), 114
 set() (OwnedVariableComponent method), 117
 set() (ParticipationProperty method), 117
 set() (PlanProperty method), 118
 set() (ReferencedObject method), 118
 set() (SampleRosterProperty method), 120
 set() (SequenceAnnotationProperty method), 121
 set() (SequenceConstraintProperty method), 122
 set() (SequenceProperty method), 122
 set() (TestProperty method), 123
 set() (UsageProperty method), 124
 set() (VariableComponentProperty method), 125
 setFileFormat() (in module sbol.libsbol), 125
 setHomepage() (in module sbol.libsbol), 126
 setInput() (ModuleDefinition method), 41
 setOption() (Config static method), 29
 setOutput() (ModuleDefinition method), 41
 setReference() (ReferencedObject method), 119

T

TestProperty (class in sbol.libsbol), 122
 testSBOL() (in module sbol.libsbol), 126
 TextProperty (class in sbol.libsbol), 123
 TopLevel (class in sbol.libsbol), 123

U

updateSequence() (ComponentDefinition method), 27
 URIProperty (class in sbol.libsbol), 123
 UsageProperty (class in sbol.libsbol), 124

V

validate() (ActivityProperty method), 21
 validate() (AgentProperty method), 22
 validate() (AnalysisProperty method), 23
 validate() (AssociationProperty method), 23
 validate() (AttachmentProperty method), 24
 validate() (BuildProperty method), 24
 validate() (CollectionProperty method), 25
 validate() (CombinatorialDerivationProperty method), 26
 validate() (ComponentDefinitionProperty method), 28
 validate() (ComponentProperty method), 29
 validate() (DesignProperty method), 32

validate() (Document method), 35
 validate() (FunctionalComponentProperty method), 36
 validate() (ImplementationProperty method), 37
 validate() (InteractionProperty method), 38
 validate() (LocationProperty method), 38
 validate() (MapsToProperty method), 39
 validate() (ModelProperty method), 40
 validate() (ModuleDefinitionProperty method), 41
 validate() (ModuleProperty method), 42
 validate() (ParticipationProperty method), 117
 validate() (PlanProperty method), 118
 validate() (SampleRosterProperty method), 120
 validate() (SequenceAnnotationProperty method), 121
 validate() (SequenceConstraintProperty method), 122
 validate() (SequenceProperty method), 122
 validate() (TestProperty method), 123
 validate() (UsageProperty method), 124
 validate() (VariableComponentProperty method), 125
 VariableComponentProperty (class in sbol.libsbol), 124
 VersionProperty (class in sbol.libsbol), 125

W

write() (ActivityProperty method), 21
 write() (AgentProperty method), 22
 write() (AnalysisProperty method), 23
 write() (AssociationProperty method), 23
 write() (AttachmentProperty method), 24
 write() (BuildProperty method), 24
 write() (CollectionProperty method), 25
 write() (CombinatorialDerivationProperty method), 26
 write() (ComponentDefinitionProperty method), 28
 write() (ComponentProperty method), 29
 write() (DesignProperty method), 32
 write() (Document method), 35
 write() (FunctionalComponentProperty method), 36
 write() (ImplementationProperty method), 37
 write() (InteractionProperty method), 38
 write() (LocationProperty method), 38
 write() (MapsToProperty method), 39
 write() (ModelProperty method), 40
 write() (ModuleDefinitionProperty method), 42
 write() (ModuleProperty method), 42
 write() (ParticipationProperty method), 117
 write() (PlanProperty method), 118
 write() (SampleRosterProperty method), 120
 write() (SequenceAnnotationProperty method), 121
 write() (SequenceConstraintProperty method), 122
 write() (SequenceProperty method), 122
 write() (TestProperty method), 123
 write() (UsageProperty method), 124
 write() (VariableComponentProperty method), 125