

---

# **TranSMART loader Documentation**

***Release 0.1.3***

**Gijs Kant**

**May 15, 2019**



---

## Contents:

---

<b>1</b>	<b>Installation and usage</b>	<b>3</b>
1.1	Usage . . . . .	3
<b>2</b>	<b>Documentation</b>	<b>5</b>
<b>3</b>	<b>Known issues</b>	<b>7</b>
<b>4</b>	<b>Development</b>	<b>9</b>
4.1	Python versions . . . . .	9
4.2	Package management and dependencies . . . . .	9
4.3	Testing and code coverage . . . . .	9
4.4	Documentation . . . . .	10
4.5	Coding style conventions and code quality . . . . .	10
<b>5</b>	<b>License</b>	<b>11</b>
<b>6</b>	<b>Credits</b>	<b>13</b>
<b>7</b>	<b>Data model</b>	<b>15</b>
7.1	Data model . . . . .	15
<b>8</b>	<b>API Reference</b>	<b>19</b>
8.1	transmart_loader package . . . . .	19
<b>9</b>	<b>Index</b>	<b>27</b>
	<b>Python Module Index</b>	<b>29</b>



This package contains classes that represent the core domain objects stored in the [TranSMART](#) platform, an open source data sharing and analytics platform for translational biomedical research.

It also provides a utility that writes such objects to tab-separated files that can be loaded into a TranSMART database using the [transmart-copy](#) tool.

Note: this is a very preliminary version, still under development. Issues can be reported at <https://github.com/thehyve/python-transmart-loader/issues>.



# CHAPTER 1

---

## Installation and usage

---

To install `transmart_loader`, do:

```
pip install transmart-loader
```

or from sources:

```
git clone https://github.com/thehyve/python_transmart_loader.git
cd python_transmart_loader
pip install .
```

## 1.1 Usage

Usage examples can be found in these projects:

- `fhir2transmart`: a tool that translates core [HL7 FHIR](#) resources to the TranSMART data model.
- `ontology2transmart`: a tool that translates ontologies available from [DIMDI](#) to TranSMART ontologies.





## CHAPTER 2

---

### Documentation

---

Full documentation of the package is available at [Read the Docs](#).



## CHAPTER 3

---

### Known issues

---

- Date values are not correctly translated



For a quick reference on software development, we refer to [the software guide checklist](#).

### 4.1 Python versions

This repository is set up with Python version 3.6

Add or remove Python versions based on project requirements. [The guide](#) contains more information about Python versions and writing Python 2 and 3 compatible code.

### 4.2 Package management and dependencies

This project uses *pip* for installing dependencies and package management.

- Dependencies should be added to *setup.py* in the *install\_requires* list.

### 4.3 Testing and code coverage

- Tests are in the `tests` folder.
- The `tests` folder contains:
  - A test if files for *transmart-copy* are generated for fake data (file: `test_transmart_loader`)
  - A test that checks whether your code conforms to the Python style guide (PEP 8) (file: `test_lint.py`)
- The testing framework used is [PyTest](#)
  - [PyTest introduction](#)
- Tests can be run with `python setup.py test`

## 4.4 Documentation

- Documentation should be put in the `docs` folder.
- To generate html documentation run `python setup.py build_sphinx`

## 4.5 Coding style conventions and code quality

- Check your code style with `prospector`
- You may need run `pip install .[dev]` first, to install the required dependencies

## CHAPTER 5

---

### License

---

Copyright (c) 2019 The Hyve B.V.

The TranSMART loader is licensed under the MIT License. See the file [LICENSE](#).





## CHAPTER 6

---

### Credits

---

This package was created with [Cookiecutter](#) and the [NLeSC/python-template](#).



#### 7.1 Data model

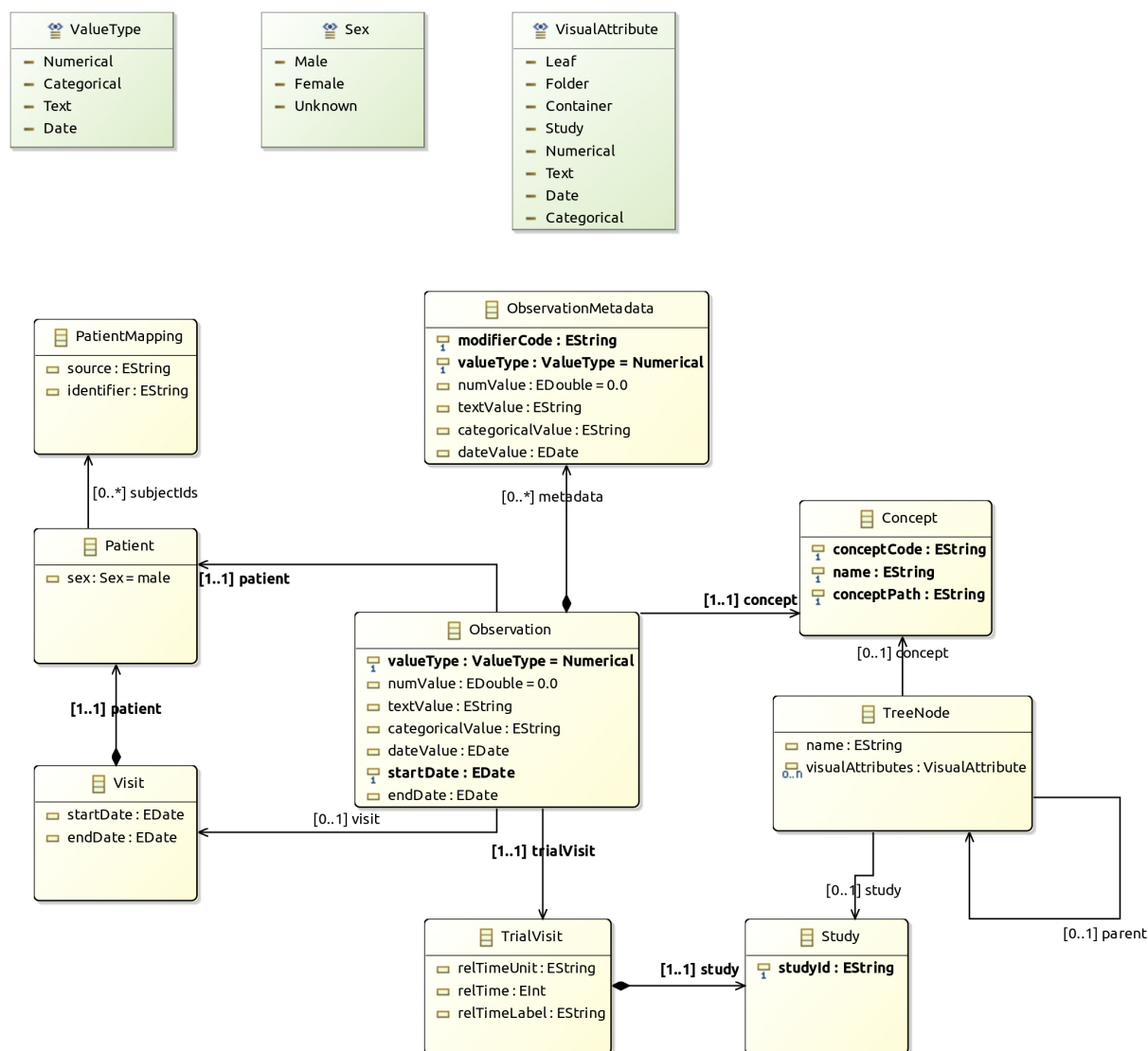


Fig. 1: TranSMART data model

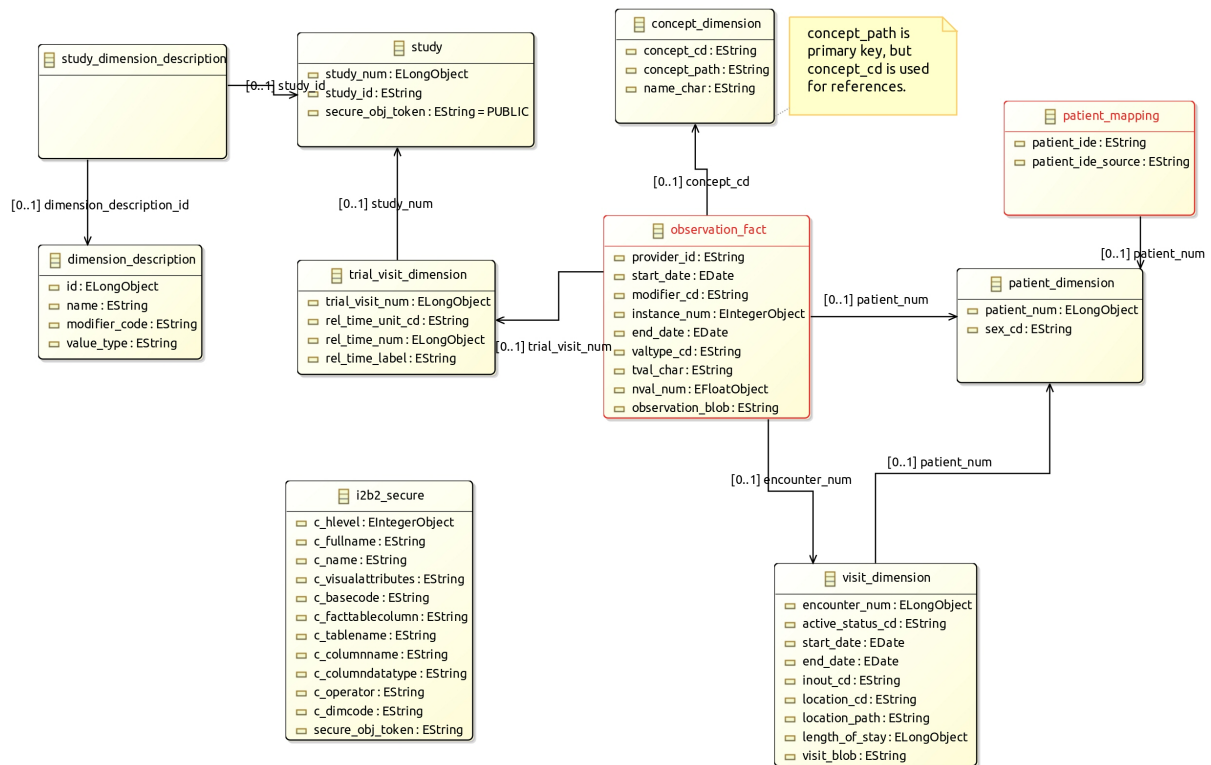


Fig. 2: TranSMART database tables



## 8.1 transmart\_loader package

Documentation about TranSMART loader

### 8.1.1 Submodules

#### transmart\_loader.collection\_validator module

```
class transmart_loader.collection_validator.CollectionValidator
    Bases: transmart_loader.collection_visitor.CollectionVisitor
    Validation class for TranSMART data collections.

    static validate (collection: transmart_loader.transmart.DataCollection)
    visit_concept (concept: transmart_loader.transmart.Concept) → None
    visit_node (node: transmart_loader.transmart.TreeNode) → None
    visit_observation (observation: transmart_loader.transmart.Observation) → None
    visit_patient (patient: transmart_loader.transmart.Patient) → None
    visit_study (study: transmart_loader.transmart.Study) → None
    visit_trial_visit (trial_visit: transmart_loader.transmart.TrialVisit) → None
    visit_visit (visit: transmart_loader.transmart.Visit) → None
```

#### transmart\_loader.collection\_visitor module

```
class transmart_loader.collection_visitor.CollectionVisitor
    Bases: object
    Visitor class for TranSMART data collections
```

**visit** (*collection*: *Optional[transmart\_loader.transmart.DataCollection]*) → None

**visit\_concept** (*concept*: *transmart\_loader.transmart.Concept*) → None

**visit\_node** (*node*: *transmart\_loader.transmart.TreeNode*) → None

**visit\_observation** (*observation*: *transmart\_loader.transmart.Observation*) → None

**visit\_patient** (*patient*: *transmart\_loader.transmart.Patient*) → None

**visit\_study** (*study*: *transmart\_loader.transmart.Study*) → None

**visit\_trial\_visit** (*trial\_visit*: *transmart\_loader.transmart.TrialVisit*) → None

**visit\_visit** (*visit*: *transmart\_loader.transmart.Visit*) → None

### transmart\_loader.console module

```
class transmart_loader.console.Console
    Bases: object

    A helper class for displaying messages on the console (stderr).

    Black = '\x1b[30m'
    BlackBackground = '\x1b[40m'
    Blue = '\x1b[94m'
    Green = '\x1b[92m'
    GreenBackground = '\x1b[42m'
    Grey = '\x1b[37m'
    Red = '\x1b[91m'
    RedBackground = '\x1b[41m'
    Reset = '\x1b[0m'
    Yellow = '\x1b[93m'
    YellowBackground = '\x1b[103m'

    static error(message)
    static info(message)
    static success(message)
    static title(title)
    static warning(message)
```

### transmart\_loader.copy\_writer module

```
class transmart_loader.copy_writer.TransmartCopyWriter(output_dir: str)
    Bases: transmart_loader.collection_visitor.CollectionVisitor

    Writes TranSMART data collections to a folder with files that can be loaded into a TranSMART database using
    transmart-copy.

    concepts_header = ['concept_cd', 'concept_path', 'name_char']
    dimensions_header = ['id', 'name', 'modifier_code', 'value_type']
```



**init\_writers** () → None

Creates files and initialises writers for the output files in transmart-copy format.

**observations\_header** = ['encounter\_num', 'patient\_num', 'concept\_cd', 'provider\_id', 's

**patient\_mappings\_header** = ['patient\_id', 'patient\_id\_source', 'patient\_num']

**patients\_header** = ['patient\_num', 'sex\_cd']

**prepare\_output\_dir** () → None

Creates an output directory if it does not exist. Fails if the output directory exists and is not empty.

**studies\_header** = ['study\_num', 'study\_id', 'secure\_obj\_token']

**study\_dimensions\_header** = ['study\_id', 'dimension\_description\_id']

**tree\_nodes\_header** = ['c\_hlevel', 'c\_fullname', 'c\_name', 'c\_visualattributes', 'c\_base

**trial\_visits\_header** = ['trial\_visit\_num', 'study\_num', 'rel\_time\_unit\_cd', 'rel\_time\_n

**value\_type\_codes** = {<ValueType.Numeric: 1>: 'N', <ValueType.Categorical: 2>: 'T',

**visit\_concept** (concept: transmart\_loader.transmart.Concept) → None

Serialises a Concept entity to a TSV file.

**Parameters concept** – the Concept entity

**visit\_node** (node: transmart\_loader.transmart.TreeNode) → None

**visit\_observation** (observation: transmart\_loader.transmart.Observation) → None

Serialises an Observation entity to a TSV file.

FIXME: fix date value serialisation

**Parameters observation** – the Observation entity

**visit\_patient** (patient: transmart\_loader.transmart.Patient) → None

Serialises an Patient entity and related PatientMapping entities to TSV files.

**Parameters patient** – the Patient entity

**visit\_study** (study: transmart\_loader.transmart.Study) → None

Serialises a Study entity to a TSV file.

**Parameters study** – the Study entity

**visit\_tree\_node** (node: transmart\_loader.transmart.TreeNode, level=0, parent\_path='')

Serialises a TreeNode entity and its children to a TSV file.

**Parameters**

- **node** – the TreeNode entity
- **level** – the hierarchy level of the node
- **parent\_path** – the path of the parent node.

**visit\_trial\_visit** (trial\_visit: transmart\_loader.transmart.TrialVisit) → None

Serialises a TrialVisit entity to a TSV file.

**Parameters trial\_visit** – the TrialVisit entity

**visit\_visit** (visit: transmart\_loader.transmart.Visit) → None

Serialises a Visit entity to a TSV file. NB: this requires all patient visits to be cleared before loading new visits for the patient.

**Parameters visit** – the Visit entity

```
visits_header = ['encounter_num', 'patient_num', 'active_status_cd', 'start_date', 'en
```

```
write_collection (collection: transmart_loader.transmart.DataCollection) → None
```

```
write_dimension (dimension: transmart_loader.transmart.Dimension) → None
```

Serialises a Dimension entity to a TSV file.

Parameters **dimension** – the Dimension entity

```
write_dimensions () → None
```

Write dimensions metadata and link all studies to the dimensions

```
write_study_dimensions (study_index)
```

```
class transmart_loader.copy_writer.VisualAttribute
```

Bases: `enum.Enum`

Visual attribute of an ontology node

```
Categorical = 8
```

```
Container = 3
```

```
Date = 7
```

```
Folder = 2
```

```
Leaf = 1
```

```
Numerical = 5
```

```
Study = 4
```

```
Text = 6
```

```
transmart_loader.copy_writer.format_date (value: Optional[datetime.date]) → Optional[str]
```

```
transmart_loader.copy_writer.get_concept_node_row (node: transmart_loader.transmart.ConceptNode,
                                                    level, node_path)
```

```
transmart_loader.copy_writer.get_folder_node_row (node: transmart_loader.transmart.TreeNode,
                                                    level, node_path)
```

```
transmart_loader.copy_writer.get_study_node_row (node: transmart_loader.transmart.StudyNode,
                                                    level, node_path)
```

## transmart\_loader.csv\_types module

```
class transmart_loader.csv_types.CsvWriter
```

Bases: `object`

This is an abstract version of the return of `csv.writer()`.

```
writerow (row: Sequence[Any]) → None
```

```
writerows (rows: Sequence[Sequence[Any]]) → None
```

## transmart\_loader.loader\_exception module

```
exception transmart_loader.loader_exception.LoaderException
```

Bases: `Exception`

**transmart\_loader.transmart module**

**class** transmart\_loader.transmart.**CategoricalValue** (value: str)

Bases: *transmart\_loader.transmart.Value*

A categorical value

**value** ()

**value\_type** ()

**class** transmart\_loader.transmart.**Concept** (concept\_code: str, name: str, concept\_path: str, value\_type: *transmart\_loader.transmart.ValueType*)

Bases: object

Concepts to classify observations

**class** transmart\_loader.transmart.**ConceptNode** (concept: *transmart\_loader.transmart.Concept*)

Bases: *transmart\_loader.transmart.TreeNode*

Concept node

**class** transmart\_loader.transmart.**DataCollection** (concepts: *Iterable[transmart\_loader.transmart.Concept]*, studies: *Iterable[transmart\_loader.transmart.Study]*, trial\_visits: *Iterable[transmart\_loader.transmart.TrialVisit]*, visits: *Iterable[transmart\_loader.transmart.Visit]*, ontology: *Iterable[transmart\_loader.transmart.TreeNode]*, patients: *Iterable[transmart\_loader.transmart.Patient]*, observations: *Iterable[transmart\_loader.transmart.Observation]*)

Bases: object

A data collection that can be loaded into TranSMART

**class** transmart\_loader.transmart.**DateValue** (value: datetime.date)

Bases: *transmart\_loader.transmart.Value*

A date value

**value** ()

**value\_type** ()

**class** transmart\_loader.transmart.**Dimension** (name: str, modifier\_code: *Optional[str]* = None, value\_type: *Optional[transmart\_loader.transmart.ValueType]* = None)

Bases: object

Dimension metadata

**class** transmart\_loader.transmart.**NumericalValue** (value: float)

Bases: *transmart\_loader.transmart.Value*

A numerical value

```

    value()
    value_type()
class transmart_loader.transmart.Observation(patient: trans-
                                             mart_loader.transmart.Patient,
                                             concept: trans-
                                             mart_loader.transmart.Concept, visit: Op-
                                             tional[transmart_loader.transmart.Visit],
                                             trial_visit: trans-
                                             mart_loader.transmart.TrialVisit,
                                             start_date: Optional[datetime.date],
                                             end_date: Optional[datetime.date], value:
                                             transmart_loader.transmart.Value)

Bases: object

Data about an observed event or an attribute of a patient

class transmart_loader.transmart.ObservationMetadata
Bases: object

Metadata about an observation

class transmart_loader.transmart.Patient(identifier: str, sex: str, mappings: Se-
                                         quence[transmart_loader.transmart.PatientMapping])

Bases: object

Patient properties

class transmart_loader.transmart.PatientMapping(source: str, identifier: str)
Bases: object

Patient identifiers

class transmart_loader.transmart.Study(study_id: str, name: str)
Bases: object

class transmart_loader.transmart.StudyNode(study: transmart_loader.transmart.Study)
Bases: transmart_loader.transmart.TreeNode

Study node

class transmart_loader.transmart.TextValue(value: str)
Bases: transmart_loader.transmart.Value

A text value

    value()
    value_type()

class transmart_loader.transmart.TreeNode(name: str)
Bases: object

Ontology node

    add_child(child: transmart_loader.transmart.TreeNode)

class transmart_loader.transmart.TrialVisit(study: transmart_loader.transmart.Study,
                                             rel_time_label: str, rel_time_unit: Op-
                                             tional[str] = None, rel_time: Optional[int]
                                             = None)

Bases: object

Trial visit

```

```
class transmart_loader.transmart.Value
```

Bases: object

An observed value

**value**

**value\_type**

```
class transmart_loader.transmart.ValueType
```

Bases: enum.Enum

Type of an observed value

**Categorical** = 2

**Date** = 4

**Numeric** = 1

**Text** = 3

```
class transmart_loader.transmart.Visit (patient:      transmart_loader.transmart.Patient,  
                                         identifier:  str, active_status:  Optional[str],  
                                         start_date:  Optional[datetime.date], end_date:  
                                         Optional[datetime.date], inout: Optional[str], loca-  
                                         tion: Optional[str], length_of_stay: Optional[int])
```

Bases: object

Patient visit

### transmart\_loader.tsv\_writer module

```
class transmart_loader.tsv_writer.TsvWriter (path: str)
```

Bases: *transmart\_loader.csv\_types.CsvWriter*

Tab-separated values writer. Creates a new file when initialised and fails when the file already exists.

**close** () → None

**writerow** (row: Sequence[Any]) → None

**writerows** (rows: Sequence[Sequence[Any]]) → None



## CHAPTER 9

---

### Index

---

- `genindex`





### t

`transmart_loader`, [19](#)  
`transmart_loader.collection_validator`,  
    [19](#)  
`transmart_loader.collection_visitor`, [19](#)  
`transmart_loader.console`, [20](#)  
`transmart_loader.copy_writer`, [20](#)  
`transmart_loader.csv_types`, [22](#)  
`transmart_loader.loader_exception`, [22](#)  
`transmart_loader.transmart`, [23](#)  
`transmart_loader.tsv_writer`, [25](#)



## A

`add_child()` (*transmart\_loader.transmart.TreeNode* method), 24

## B

`Black` (*transmart\_loader.console.Console* attribute), 20

`BlackBackground` (*transmart\_loader.console.Console* attribute), 20

`Blue` (*transmart\_loader.console.Console* attribute), 20

## C

`Categorical` (*transmart\_loader.copy\_writer.VisualAttribute* attribute), 22

`Categorical` (*transmart\_loader.transmart.ValueType* attribute), 25

`CategoricalValue` (class in *transmart\_loader.transmart*), 23

`close()` (*transmart\_loader.tsv\_writer.TsvWriter* method), 25

`CollectionValidator` (class in *transmart\_loader.collection\_validator*), 19

`CollectionVisitor` (class in *transmart\_loader.collection\_visitor*), 19

`Concept` (class in *transmart\_loader.transmart*), 23

`ConceptNode` (class in *transmart\_loader.transmart*), 23

`concepts_header` (*transmart\_loader.copy\_writer.TransmartCopyWriter* attribute), 20

`Console` (class in *transmart\_loader.console*), 20

`Container` (*transmart\_loader.copy\_writer.VisualAttribute* attribute), 22

`CsvWriter` (class in *transmart\_loader.csv\_types*), 22

## D

`DataCollection` (class in *transmart\_loader.transmart*), 23

`Date` (*transmart\_loader.copy\_writer.VisualAttribute* attribute), 22

`Date` (*transmart\_loader.transmart.ValueType* attribute), 25

`DateValue` (class in *transmart\_loader.transmart*), 23

`Dimension` (class in *transmart\_loader.transmart*), 23

`dimensions_header` (*transmart\_loader.copy\_writer.TransmartCopyWriter* attribute), 20

## E

`error()` (*transmart\_loader.console.Console* static method), 20

## F

`Folder` (*transmart\_loader.copy\_writer.VisualAttribute* attribute), 22

`format_date()` (in module *transmart\_loader.copy\_writer*), 22

## G

`get_concept_node_row()` (in module *transmart\_loader.copy\_writer*), 22

`get_folder_node_row()` (in module *transmart\_loader.copy\_writer*), 22

`get_study_node_row()` (in module *transmart\_loader.copy\_writer*), 22

`Green` (*transmart\_loader.console.Console* attribute), 20

`GreenBackground` (*transmart\_loader.console.Console* attribute), 20

`Grey` (*transmart\_loader.console.Console* attribute), 20

`info()` (*transmart\_loader.console.Console* static method), 20

`init_writers()` (*transmart\_loader.copy\_writer.TransmartCopyWriter* method), 20

## L

Leaf (*transmart\_loader.copy\_writer.VisualAttribute attribute*), 22  
 LoaderException, 22

## N

Numeric (*transmart\_loader.transmart.ValueType attribute*), 25  
 Numerical (*transmart\_loader.copy\_writer.VisualAttribute attribute*), 22  
 NumericalValue (*class in transmart\_loader.transmart*), 23

## O

Observation (*class in transmart\_loader.transmart*), 24  
 ObservationMetadata (*class in transmart\_loader.transmart*), 24  
 observations\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21

## P

Patient (*class in transmart\_loader.transmart*), 24  
 patient\_mappings\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 PatientMapping (*class in transmart\_loader.transmart*), 24  
 patients\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 prepare\_output\_dir() (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21

## R

Red (*transmart\_loader.console.Console attribute*), 20  
 RedBackground (*transmart\_loader.console.Console attribute*), 20  
 Reset (*transmart\_loader.console.Console attribute*), 20

## S

studies\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 Study (*class in transmart\_loader.transmart*), 24  
 Study (*transmart\_loader.copy\_writer.VisualAttribute attribute*), 22  
 study\_dimensions\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 StudyNode (*class in transmart\_loader.transmart*), 24

success() (*transmart\_loader.console.Console static method*), 20

## T

Text (*transmart\_loader.copy\_writer.VisualAttribute attribute*), 22  
 Text (*transmart\_loader.transmart.ValueType attribute*), 25  
 TextValue (*class in transmart\_loader.transmart*), 24  
 title() (*transmart\_loader.console.Console static method*), 20  
 transmart\_loader (*module*), 19  
 transmart\_loader.collection\_validator (*module*), 19  
 transmart\_loader.collection\_visitor (*module*), 19  
 transmart\_loader.console (*module*), 20  
 transmart\_loader.copy\_writer (*module*), 20  
 transmart\_loader.csv\_types (*module*), 22  
 transmart\_loader.loader\_exception (*module*), 22  
 transmart\_loader.transmart (*module*), 23  
 transmart\_loader.tsv\_writer (*module*), 25  
 TransmartCopyWriter (*class in transmart\_loader.copy\_writer*), 20  
 tree\_nodes\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 TreeNode (*class in transmart\_loader.transmart*), 24  
 trial\_visits\_header (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
 TrialVisit (*class in transmart\_loader.transmart*), 24  
 TsvWriter (*class in transmart\_loader.tsv\_writer*), 25

## V

validate() (*transmart\_loader.collection\_validator.CollectionValidator static method*), 19  
 Value (*class in transmart\_loader.transmart*), 24  
 value (*transmart\_loader.transmart.Value attribute*), 25  
 value() (*transmart\_loader.transmart.CategoricalValue method*), 23  
 value() (*transmart\_loader.transmart.DateValue method*), 23  
 value() (*transmart\_loader.transmart.NumericalValue method*), 23  
 value() (*transmart\_loader.transmart.TextValue method*), 24  
 value\_type (*transmart\_loader.transmart.Value attribute*), 25  
 value\_type() (*transmart\_loader.transmart.CategoricalValue method*), 23

[value\\_type\(\)](#) (*transmart\_loader.transmart.DateValue method*), 23  
[value\\_type\(\)](#) (*transmart\_loader.transmart.NumericalValue method*), 24  
[value\\_type\(\)](#) (*transmart\_loader.transmart.TextValue method*), 24  
[value\\_type\\_codes](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
[ValueType](#) (*class in transmart\_loader.transmart*), 25  
[Visit](#) (*class in transmart\_loader.transmart*), 25  
[visit\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 19  
[visit\\_concept\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_concept\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_concept\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_node\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_node\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_node\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_observation\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_observation\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_observation\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_patient\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_patient\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_patient\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_study\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_study\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_study\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_tree\\_node\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_trial\\_visit\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_trial\\_visit\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_trial\\_visit\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visit\\_visit\(\)](#) (*transmart\_loader.collection\_validator.CollectionValidator method*), 19  
[visit\\_visit\(\)](#) (*transmart\_loader.collection\_visitor.CollectionVisitor method*), 20  
[visit\\_visit\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 21  
[visits\\_header](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter attribute*), 21  
[VisualAttribute](#) (*class in transmart\_loader.copy\_writer*), 22

## W

[warning\(\)](#) (*transmart\_loader.console.Console static method*), 20  
[write\\_collection\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 22  
[write\\_dimension\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 22  
[write\\_dimensions\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 22  
[write\\_study\\_dimensions\(\)](#) (*transmart\_loader.copy\_writer.TransmartCopyWriter method*), 22  
[writerow\(\)](#) (*transmart\_loader.csv\_types.CsvWriter method*), 22  
[writerow\(\)](#) (*transmart\_loader.tsv\_writer.TsvWriter method*), 25  
[riterows\(\)](#) (*transmart\_loader.csv\_types.CsvWriter method*), 22

`writerows()` (*transmart\_loader.tsv\_writer.TsvWriter*  
*method*), [25](#)

## Y

`Yellow` (*transmart\_loader.console.Console* *attribute*),  
[20](#)

`YellowBackground` (*trans-*  
*mart\_loader.console.Console* *attribute*),  
[20](#)