
xml miner Documentation

Release 0.0.5

Chao Li

Apr 15, 2022

Contents:

1	XML/TRXML Selector	1
1.1	Description	1
1.2	Status	1
1.3	Requirements	1
1.4	Installation	1
1.5	Usage	1
1.6	Development	3
1.7	selector and output details:	3
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	xml_miner	7
3.1	xml_miner package	7
4	Credits	17
4.1	Development Lead	17
4.2	Contributors	17
5	0.0.5 (2019-10-14)	19
6	0.0.4 (2019-09-11)	21
7	0.0.3 (2019-08-11)	23
8	0.0.2 (2019-08-09)	25
9	0.0.1 (2019-08-09)	27
10	0.0.0 (2019-08-06)	29
11	Indices and tables	31
	Python Module Index	33
	Index	35

1.1 Description

This package provides two scripts: `mine-xml` and `mine-trxml`.

`mine-xml` selects tags from `xml/mxml` files, and save the selected values to file.

`mine-trxml` selects fields from `trxml/mtrxml` files, and save the selected values to file.

1.2 Status

1.3 Requirements

Python 3.6+

1.4 Installation

```
pip install xml-selector
```

1.5 Usage

1.5.1 Use xml selector script

The xml selector supports:

- one or more tagnames:
- selector could be one tagname name
- or comma separated tagnames langskill,compskill,softskills
- multiple sources:
- e.g. select from xml dir, xml files, mxml file, or directly from annotation server

examples:

```
#select from xml directory
mine-xml --source tests/xm1s/ --selector name --output_file name.tsv
mine-xml --source tests/xm1s/ --selector langskill,compskill,softskill --output_file_
↪skill.tsv --with_field_name

#select from xml file or mxml file
mine-xml --source tests/sample.mxml --selector experience --output_file experience.tsv

#select directly from annotation server
mine-xml --source localhost:50249 --selector name --output_file name.tsv --query "set_
↪Data2018"
```

1.5.2 Use trxml selector script

The trxml selector supports:

- one or more selectors:
- selector can be one field: name.0.name
- or comma separated fields: name.0.name,address.0.address
- single or multi item:
- can select field from one item, e.g. experienceitem.3.experience
- or select field value of all item, e.g. experienceitem.experience (or experienceitem.*.experience)
- multiple sources:
- e.g. select from trxml dir, trxml files, or mtrxml file

examples:

```
# one selector, single item
mine-trxml --source tests/trxm1s/ --selector name.0.name --output_file name.tsv

# one selector, multiple item
mine-trxml --source tests/sample.mxml --selector experienceitem.experience --output_
↪file experience.tsv

# more selectors, single item
```

(continues on next page)

(continued from previous page)

```

mine-trxml --source tests/trxmls/ --selector name.0.name,address.0.address,phone.0.
↳phone --output_file personal.tsv

# more selectors, multiple item
mine-trxml --source tests/sample.mxml --itemgroup experienceitem --fields experience,
↳experiencedate --output_file experience.tsv
mine-trxml --source tests/sample.mxml --selector experienceitem.*.experience,
↳experienceitem.*.experiencedate --output_file experience.tsv
mine-trxml --source tests/sample.mxml --selector experienceitem.experience,
↳experienceitem.experiencedate --output_file experience.tsv

```

1.6 Development

To install package and its dependencies, run the following from project root directory:

```
python setup.py install
```

To work the code and develop the package, run the following from project root directory:

```
python setup.py develop
```

To run unit tests, execute the following from the project root directory:

```
python setup.py test
```

1.7 selector and output details:

- mine-xml:

input: documents, selector(s), output

output:

- default (parameter with_field_name not set): filename, field_value

e.g. select all names with selector name

filename	value
xxxx	Chao Li

- parameter with_field_name set: filename, field_value, field_name

e.g. select skills with selector compskill,langskill,otherskill

filename	value	field
xxxx	java	compskill
xxxx	dutch	langskill

- mine-trxml

- input:

- documents, selector(s), output,

- documents, itemgroup, fields, output
- single selector:
- single item (`name.0.name`): filename field

filename	name.0.name
xxxx	Chao Li

- multi items (`skill.*.skill`): filename item_index field

filename	item_index	field
xxxx	0	java
xxxx	1	dutch

- multiple selectors
- single item: filename, field1, field2 ...

each selector points to a field of a specific item with a digital index, e.g. `name.0.lastname`, `name.0.firstname`, `address.0.country`

filename	name.0.lastname	name.0.firstname	address.0.country
xxxx	Li	Chao	China
xxxx	Lee	Richard	USA

- multi items: filename, item_index, field1, field2 ...

each selector points to a field from all items in an itemgroup, e.g. `skill.skill`, `skill.type`, `skill.date`

filename	skill	skill	type	date
xxxx	0	java	compskill	2001-2005
xxxx	1	dutch	langskill	2002-

2.1 Stable release

To install xml-miner, run this command in your terminal:

```
$ pip install xml-miner
```

This is the preferred method to install xml-miner, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for xml-miner can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/tilaboy/xml-miner
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/tilaboy/xml-miner/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


3.1 xml_miner package

3.1.1 Subpackages

xml_miner.data_utils package

Submodules

xml_miner.data_utils.asclient module

A module to communicate the TK annotation server

```
class xml_miner.data_utils.asclient.ASClient (host: str, port: str, as_user: str = "",  
                                              as_pass: str = "")
```

Bases: object

Python version of annotation server client

BUFFER_SIZE = 2048

ENCODING = 'utf-8'

check_user_password ()

send server the username and password and confirm the login

close_socket ()

shutdown the connect

get_docs (*query=""*) → Iterator[str]

get all queried documents

get_ids (*query: str = ""*) → List[str]

get all document ids of all the queried documents

make_connection () → bool
Build connect to the Annotation Server

prepare_message (*message: str*) → str
prepare the query message: - add new line - encode with utf8

send_and_receive (*query: str*) → str
send the query to AS and decode the received response

socket_output () → str
Receive response from Annotation Server and decode

xml_miner.data_utils.data_loader module

A module to load input xml/trxml from different source

```
class xml_miner.data_utils.data_loader.DataLoader (data_generator=None)
    Bases: object

    DataLoader: - load data from different resources - generate xml input for downstream tasks

    TRXML_HEADER = '<TexttractorResult '

    XML_HEADER = '<begin '

    classmethod load_from_as (host, port, query, as_user="", as_pass="")
        create the document loader object from AnnotationServer

        params: host (string): hostname of the annotationserver port (int): port of the AnnotationServer query
            (string): query to select documents as_user: AnnotationServer username as_pass: AnnotationServer
            password

        output: DataLoader object: a iterator object to generate xml string

    classmethod load_from_dir (input_dir)
        create the document loader object from dir

        params: input_dir (string): director contains xml files

        output: DataLoader object: a iterator object to generate xml string

    classmethod load_from_mtrxml (input_mxml)
        create the document loader object from mxml

        params: mxml (string): a mxml files

        output: DataLoader object: a iterator object to generate xml string

    classmethod load_from_mxml (input_mxml)
        create the document loader object from mxml

        params: mxml (string): a mxml files

        output: DataLoader object: a iterator object to generate xml string

xml_miner.data_utils.data_loader.load_from_file (input_dir, files)
    load document from a list files read from a directory

    params: input_dir (string): directory contains all files files (list): a list files

    output: xml string: a iterator object to generate xml string

xml_miner.data_utils.data_loader.load_from_string (xml_string, header_line)
    load document string, the string might contain multiple xml files
```

params: xml_string (string): input xml strings

output: xml string: a iterator object to generate xml string

xml_miner.data_utils.data_saver module

A Module to output the selected values to the chosen format

class xml_miner.data_utils.data_saver.DataSaver(*output_file*)

Bases: object

DataLoader: - open/create targed output file - save the selected values to file with corresponding format

close_stream()

close the file

store(*row*)

Store one row at a time

Module contents

DataLoader and DataSaver classes

class xml_miner.data_utils.DataLoader(*data_generator=None*)

Bases: object

DataLoader: - load data from different resources - generate xml input for downstream tasks

TRXML_HEADER = '<TexttractorResult '

XML_HEADER = '<begin '

classmethod load_from_as(*host, port, query, as_user="", as_pass=""*)

create the document loader object from AnnotationServer

params: host (string): hostname of the annotationserver port (int): port of the AnnotationServer query (string): query to select documents as_user: AnnotationServer username as_pass: AnnotationServer password

output: DataLoader object: a iterator object to generate xml string

classmethod load_from_dir(*input_dir*)

create the document loader object from dir

params: input_dir (string): director contains xml files

output: DataLoader object: a iterator object to generate xml string

classmethod load_from_mtrxml(*input_mxml*)

create the document loader object from mxml

params: mxml (string): a mxml files

output: DataLoader object: a iterator object to generate xml string

classmethod load_from_mxml(*input_mxml*)

create the document loader object from mxml

params: mxml (string): a mxml files

output: DataLoader object: a iterator object to generate xml string

```
class xml_miner.data_utils.DataSaver(output_file)
    Bases: object

    DataLoader: - open/create targeted output file - save the selected values to file with corresponding format

    close_stream()
        close the file

    store(row)
        Store one row at a time
```

xml_miner.selectors package

Submodules

xml_miner.selectors.selector_utils module

utils and constants functions used by the selector and selectors class

```
xml_miner.selectors.selector_utils.selector_attribute(selectors, attribute_name) → str
    fetch the selector attribute, and check the consistency of all selectors

    params: - selectors: a list of selector object - attribute_name: name of the attribute
    output: attribute_value: string

xml_miner.selectors.selector_utils.valid_field_name(tag_name: str = "") → bool
    simple validation function:

    params: - tag_name: string
    output: - True/False
```

xml_miner.selectors.trxml_selector module

selector class for trxml

```
class xml_miner.selectors.trxml_selector.TRXMLSelector(selector: str)
    Bases: xml_miner.selectors.xml_selector.XMLSelector

    trxml selector:
        • subclass of XMLSelector
        • method to select values from trxml

    field_value_from_item(item) → str
        given an item and a field_name, get the value of that field

    parse_trxml_selector()
        converting the trxml selector to (itemgroup, index, field):

        params:
            • selector: string

        output:
            • itemgroup, index, field

        conversion rules:
```

- ig.index.field	->	(ig, index, field)
- ig.*.field	->	(ig, *, field)
- ig.field	->	(ig, *, field)

select_field_with_xpath (*xml_tree*)

select the field using the selector xpath

select_value_with_xpath (*xml_tree*) → str

get the value of the field where the selector matches

xml_miner.selectors.trxml_selectors module

TRXML Selectors class

```
class xml_miner.selectors.trxml_selectors.TRXMLSelectors (selectors:      List[str],
                                                         trxml_selector_type=None,
                                                         shared_itemgroup_name=None)
```

Bases: object

TRXMLSelectors: - array of TRXMLSelector class - method to select values on trxml doc level or from each items

classmethod from_itemgroup_and_fields (*itemgroup: str, fields: str*)

construct from itemgroup and fields, only for trxml

input:

- ItemGroup, e.g. experienceitem
- Fields, e.g. jobtitle,startdate,enddate

classmethod from_selector_string (*selector_string: str*)

construct the selectors from string

input:

- selector string

select_trxml_fields (*trxml*)

select values from all fields matching selectors

xml_miner.selectors.xml_selector module

XML selector

```
class xml_miner.selectors.xml_selector.XMLSelector (selector: str)
```

Bases: object

XMLSelector: - select all values of nodes matches selector

select_all_fields (*xml_tree*)

select all fields match selectors

select_all_values (*xml_tree*) → List[str]

select all values match selectors

xml_miner.selectors.xml_selectors module

XML Selectors class

```
class xml_miner.selectors.xml_selectors.XMLSelectors (selectors: List[str])
    Bases: object

    XMLSelectors: - array of XMLSelector class - method to select values from xml object

    classmethod from_selector_string (selector_string: str)
        construct xml selector from input string

    select_xml_fields (xml_tree)
        select all values matches the selector
```

Module contents

xml selectors and trxml selectors classes

```
class xml_miner.selectors.XMLSelectors (selectors: List[str])
    Bases: object

    XMLSelectors: - array of XMLSelector class - method to select values from xml object

    classmethod from_selector_string (selector_string: str)
        construct xml selector from input string

    select_xml_fields (xml_tree)
        select all values matches the selector

class xml_miner.selectors.TRXMLSelectors (selectors: List[str], trxml_selector_type=None,
                                          shared_itemgroup_name=None)
    Bases: object

    TRXMLSelectors: - array of TRXMLSelector class - method to select values on trxml doc level or from each
    items

    classmethod from_itemgroup_and_fields (itemgroup: str, fields: str)
        construct from itemgroup and fields, only for trxml

    input:

        • ItemGroup, e.g. experienceitem

        • Fields, e.g. jobtitle,startdate,enddate

    classmethod from_selector_string (selector_string: str)
        construct the selectors from string

    input:

        • selector string

    select_trxml_fields (trxml)
        select values from all fields matching selectors
```

xml_miner.xml package

Submodules

xml_miner.xml.base_xml module

XML class: render xml file or xml strings to xml tree object

class xml_miner.xml.base_xml.**XML** (*top_level_obj=None*)

Bases: object

XML: general xml class, xml tree object can be generated from - xml file - xml string

classmethod from_file (*xml_file: str*)

create xml object from filename

params: xml_file (string): xml file

output: xml object: ElementTree object

classmethod from_string (*xml_string: str*)

create xml object from xml_string

params: xml_string (string): xml string

output: xml object: ElementTree object

static text_from_element (*element*)

the text value of an xml element

xml_miner.xml.tk_trxml module

TRXML class: render field or strings to trxml class, and select using xpath

class xml_miner.xml.tk_trxml.**TKTRXML** (*top_level_obj=None*)

Bases: *xml_miner.xml.base_xml.XML*

TRXML: - render field or strings to trxml class - and select using xpath

filename

filename of the original file

normally stored as an attribute of the top level tag

working_entity

the xml element to apply searching

xml_entity

the xml part of the tree

xml_miner.xml.tk_xml module

TRXML class: render field or strings to xml class, and select values

class xml_miner.xml.tk_xml.**TKXML** (*top_level_obj=None*)

Bases: *xml_miner.xml.base_xml.XML*

TKXML class: xml tree object generated from - xml file - xml string

working_entity

the xml element to apply searching

Module contents

xml and trxml classes

class `xml_miner.xml.TKXML` (*top_level_obj=None*)

Bases: `xml_miner.xml.base_xml.XML`

TKXML class: xml tree object generated from - xml file - xml string

working_entity

the xml element to apply searching

class `xml_miner.xml.TKTRXML` (*top_level_obj=None*)

Bases: `xml_miner.xml.base_xml.XML`

TRXML: - render field or strings to trxml class - and select using xpath

filename

filename of the original file

normally stored as an attribute of the top level tag

working_entity

the xml element to apply searching

xml_entity

the xml part of the tree

3.1.2 Submodules

3.1.3 `xml_miner.mine_trxml` module

the trxml selector script

`xml_miner.mine_trxml.get_args()`

get arguments

`xml_miner.mine_trxml.main()`

apply selectors to trxml files

3.1.4 `xml_miner.mine_xml` module

the xml selector script

`xml_miner.mine_xml.get_args()`

get arguments

`xml_miner.mine_xml.main()`

apply selectors to xml files

3.1.5 `xml_miner.miner` module

apply selector on input data, and output it to a csv file

class `xml_miner.miner.CommonMiner` (*selectors*)

Bases: `object`

CommonMiner:

shared class for both xml and trxml

static `normalize_string` (*line: str*) → str

normalization selected values: - replace

with ‘__NEWLINE__’

- replace with 4 space ‘ ‘

class `xml_miner.miner.TRXMLMiner` (*selectors, itemgroup=None, fields=None*)

Bases: `xml_miner.miner.CommonMiner`

TRXMLPorcessor: - iterate over the trxml files and select values - output selected values to a file, and print summary

load_data (*source*)

load the data into a data generator

params:

- source: data source

output:

- yeild trxml

mine (*source*)

iterate the input data (trxml obj), apply selector on each trxml, and output the selected values to a csv file

params: source: data source

output: generate selected values per doc

mine_and_save (*source: str, output_file: str*)

iterate the input data (trxml obj), apply selector on each trxml, and output the selected values to a csv file

params: source (string): data source output_file (string): the output filename

read_selectors (*selector: str, itemgroup: str = "", fields: str = ""*)

read selector strings and construct selector object

params:

- selector: input selector strings
- itemgroup: input itemgroup strings
- fields: input fields strings

output:

- selectors: TRXMLSelectors object

class `xml_miner.miner.XMLMiner` (*selectors, with_field_name=False*)

Bases: `xml_miner.miner.CommonMiner`

XMLPorcessor: - iterate over the xml files and select values - output selected values to a file, and print summary

load_data (*source: str, query: str = None, as_user: str = None, as_pass: str = None*)

load the data into a data generator

params:

- source: data source

- annotation server parameters: query, as_user, as_pass

output:

- yeild xml

mine (*source: str, query: str = None, as_user: str = None, as_pass: str = None*)
iterate the input data (xml obj), apply selector on each xml, and yield the selected values

params:

- source: data source
- annotation server parameters: query, as_user, as_pass

output:

- iterate over selected fields per doc

mine_and_save (*source: str, output_file: str, query: str = None, as_user: str = None, as_pass: str = None*)
iterate the selected values and save/print to ouput

params:

- source: data source
- output_file (string): the output filename
- annotation server parameters: query, as_user, as_pass

output file format:

- no field name: filename value
- with field name: filename, value, field_name

read_selectors (*selector: str*)
read selector strings and construct selectors object

params:

- selector: input selector strings

output:

- selectors: XMLSelectors object

3.1.6 Module contents

Top-level package for xml-miner

`xml_miner.define_logger(mod_name)`
Set the default logging configuration

`xml_miner.set_logging_level(level=30)`
Change logging level

4.1 Development Lead

- Chao Li <chaoli.job@google.com>

4.2 Contributors

- Chao Li

CHAPTER 5

0.0.5 (2019-10-14)

- bug fix: ElementTree xpath find will return a None if value is an empty string, restore to empty string

CHAPTER 6

0.0.4 (2019-09-11)

- bug fix: reading always use utf8, and not continue reading if failed on encoding of one document

CHAPTER 7

0.0.3 (2019-08-11)

- expand miner.py module to generate matched phrases per doc

CHAPTER 8

0.0.2 (2019-08-09)

- added support for CI

CHAPTER 9

0.0.1 (2019-08-09)

- make two script: mine-xml and mine-trxml

CHAPTER 10

0.0.0 (2019-08-06)

- Add the first version of the mine_xml and mine_trxml

CHAPTER 11

Indices and tables

- `genindex`
- `modindex`
- `search`

X

- `xml_miner`, [16](#)
- `xml_miner.data_utils`, [9](#)
- `xml_miner.data_utils.asclient`, [7](#)
- `xml_miner.data_utils.data_loader`, [8](#)
- `xml_miner.data_utils.data_saver`, [9](#)
- `xml_miner.mine_trxml`, [14](#)
- `xml_miner.mine_xml`, [14](#)
- `xml_miner.miner`, [14](#)
- `xml_miner.selectors`, [12](#)
- `xml_miner.selectors.selector_utils`, [10](#)
- `xml_miner.selectors.trxml_selector`, [10](#)
- `xml_miner.selectors.trxml_selectors`, [11](#)
- `xml_miner.selectors.xml_selector`, [11](#)
- `xml_miner.selectors.xml_selectors`, [12](#)
- `xml_miner.xml`, [14](#)
- `xml_miner.xml.base_xml`, [13](#)
- `xml_miner.xml.tk_trxml`, [13](#)
- `xml_miner.xml.tk_xml`, [13](#)

A

ASClient (class in *xml_miner.data_utils.asclient*), 7

B

BUFFER_SIZE (*xml_miner.data_utils.asclient.ASClient* attribute), 7

C

check_user_password() (*xml_miner.data_utils.asclient.ASClient* method), 7

close_socket() (*xml_miner.data_utils.asclient.ASClient* method), 7

close_stream() (*xml_miner.data_utils.data_saver.DataSaver* method), 9

close_stream() (*xml_miner.data_utils.DataSaver* method), 10

CommonMiner (class in *xml_miner.miner*), 14

D

DataLoader (class in *xml_miner.data_utils*), 9

DataLoader (class in *xml_miner.data_utils.data_loader*), 8

DataSaver (class in *xml_miner.data_utils*), 9

DataSaver (class in *xml_miner.data_utils.data_saver*), 9

define_logger() (in module *xml_miner*), 16

E

ENCODING (*xml_miner.data_utils.asclient.ASClient* attribute), 7

F

field_value_from_item() (*xml_miner.selectors.trxml_selector.TRXMLSelector* method), 10

filename (*xml_miner.xml.tk_trxml.TKTRXML* attribute), 13

filename (*xml_miner.xml.TKTRXML* attribute), 14

from_file() (*xml_miner.xml.base_xml.XML* class method), 13

from_itemgroup_and_fields() (*xml_miner.selectors.trxml_selectors.TRXMLSelectors* class method), 11

from_itemgroup_and_fields() (*xml_miner.selectors.TRXMLSelectors* class method), 12

from_selector_string() (*xml_miner.selectors.trxml_selectors.TRXMLSelectors* class method), 11

from_selector_string() (*xml_miner.selectors.TRXMLSelectors* class method), 12

from_selector_string() (*xml_miner.selectors.xml_selectors.XMLSelectors* class method), 12

from_selector_string() (*xml_miner.selectors.XMLSelectors* class method), 12

from_string() (*xml_miner.xml.base_xml.XML* class method), 13

G

get_args() (in module *xml_miner.mine_trxml*), 14

get_args() (in module *xml_miner.mine_xml*), 14

get_docs() (*xml_miner.data_utils.asclient.ASClient* method), 7

get_ids() (*xml_miner.data_utils.asclient.ASClient* method), 7

L

load_data() (*xml_miner.miner.TRXMLMiner* method), 15

load_data() (*xml_miner.miner.XMLMiner* method), 15

load_from_as() (*xml_miner.data_utils.data_loader.DataLoader* class method), 8

load_from_as() (*xml_miner.data_utils.DataLoader* class method), 9

`load_from_dir()` (`xml_miner.data_utils.data_loader.DataLoader` `xml_miner.selectors.xml_selector.XMLSelector` class method), 8

`load_from_dir()` (`xml_miner.data_utils.DataLoader` class method), 9

`load_from_file()` (in module `xml_miner.data_utils.data_loader`), 8

`load_from_mtrxml()` (`xml_miner.data_utils.data_loader.DataLoader` class method), 8

`load_from_mtrxml()` (`xml_miner.data_utils.DataLoader` class method), 9

`load_from_mxml()` (`xml_miner.data_utils.data_loader.DataLoader` class method), 8

`load_from_mxml()` (`xml_miner.data_utils.DataLoader` class method), 9

`load_from_string()` (in module `xml_miner.data_utils.data_loader`), 8

M

`main()` (in module `xml_miner.mine_trxml`), 14

`main()` (in module `xml_miner.mine_xml`), 14

`make_connection()` (`xml_miner.data_utils.asclient.ASClient` method), 7

`mine()` (`xml_miner.miner.TRXMLMiner` method), 15

`mine()` (`xml_miner.miner.XMLMiner` method), 16

`mine_and_save()` (`xml_miner.miner.TRXMLMiner` method), 15

`mine_and_save()` (`xml_miner.miner.XMLMiner` method), 16

N

`normalize_string()` (`xml_miner.miner.CommonMiner` static method), 15

P

`parse_trxml_selector()` (`xml_miner.selectors.trxml_selector.TRXMLSelector` method), 10

`prepare_message()` (`xml_miner.data_utils.asclient.ASClient` method), 8

R

`read_selectors()` (`xml_miner.miner.TRXMLMiner` method), 15

`read_selectors()` (`xml_miner.miner.XMLMiner` method), 16

S

`select_all_fields()`

`select_all_values()` (`xml_miner.selectors.xml_selector.XMLSelector` method), 11

`select_field_with_xpath()` (`xml_miner.selectors.trxml_selector.TRXMLSelector` method), 11

`select_trxml_fields()` (`xml_miner.selectors.trxml_selectors.TRXMLSelectors` method), 11

`select_trxml_fields()` (`xml_miner.selectors.TRXMLSelectors` method), 12

`select_value_with_xpath()` (`xml_miner.selectors.trxml_selector.TRXMLSelector` method), 11

`select_xml_fields()` (`xml_miner.selectors.xml_selectors.XMLSelectors` method), 12

`select_xml_fields()` (`xml_miner.selectors.XMLSelectors` method), 12

`selector_attribute()` (in module `xml_miner.selectors.selector_utils`), 10

`send_and_receive()` (`xml_miner.data_utils.asclient.ASClient` method), 8

`set_logging_level()` (in module `xml_miner`), 16

`socket_output()` (`xml_miner.data_utils.asclient.ASClient` method), 8

`store()` (`xml_miner.data_utils.data_saver.DataSaver` method), 9

`store()` (`xml_miner.data_utils.DataSaver` method), 10

T

`text_from_element()` (`xml_miner.xml.base_xml.XML` static method), 13

`TKTRXML` (class in `xml_miner.xml`), 14

`TKTRXML` (class in `xml_miner.xml.tk_trxml`), 13

`TKXML` (class in `xml_miner.xml`), 14

`TKXML` (class in `xml_miner.xml.tk_xml`), 13

`TRXML_HEADER` (`xml_miner.data_utils.data_loader.DataLoader` attribute), 8

`TRXML_HEADER` (`xml_miner.data_utils.DataLoader` attribute), 9

`TRXMLMiner` (class in `xml_miner.miner`), 15

`TRXMLSelector` (class in `xml_miner.selectors.trxml_selector`), 10

`TRXMLSelectors` (class in `xml_miner.selectors`), 12

`TRXMLSelectors` (class in `xml_miner.selectors.trxml_selectors`), 11

V

`valid_field_name()` (in module `xml_miner.selectors.selector_utils`), 10

W

`working_entity(xml_miner.xml.tk_trxml.TKTRXML attribute)`, 13

`working_entity(xml_miner.xml.tk_xml.TKXML attribute)`, 13

`working_entity(xml_miner.xml.TKTRXML attribute)`, 14

`working_entity(xml_miner.xml.TKXML attribute)`, 14

X

`XML` (class in `xml_miner.xml.base_xml`), 13

`xml_entity(xml_miner.xml.tk_trxml.TKTRXML attribute)`, 13

`xml_entity(xml_miner.xml.TKTRXML attribute)`, 14

`XML_HEADER(xml_miner.data_utils.data_loader.DataLoader attribute)`, 8

`XML_HEADER(xml_miner.data_utils.DataLoader attribute)`, 9

`xml_miner` (module), 16

`xml_miner.data_utils` (module), 9

`xml_miner.data_utils.asclient` (module), 7

`xml_miner.data_utils.data_loader` (module), 8

`xml_miner.data_utils.data_saver` (module), 9

`xml_miner.mine_trxml` (module), 14

`xml_miner.mine_xml` (module), 14

`xml_miner.miner` (module), 14

`xml_miner.selectors` (module), 12

`xml_miner.selectors.selector_utils` (module), 10

`xml_miner.selectors.trxml_selector` (module), 10

`xml_miner.selectors.trxml_selectors` (module), 11

`xml_miner.selectors.xml_selector` (module), 11

`xml_miner.selectors.xml_selectors` (module), 12

`xml_miner.xml` (module), 14

`xml_miner.xml.base_xml` (module), 13

`xml_miner.xml.tk_trxml` (module), 13

`xml_miner.xml.tk_xml` (module), 13

`XMLMiner` (class in `xml_miner.miner`), 15

`XMLSelector` (class in `xml_miner.selectors.xml_selector`), 11

`XMLSelectors` (class in `xml_miner.selectors`), 12

`XMLSelectors` (class in `xml_miner.selectors.xml_selectors`), 12