LMF library

Generated by Doxygen 1.8.7

Fri Jul 24 2015 12:25:33

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

eol . . . ??
lmf . . . ??
lmf.src . . . ??
lmf.src.common . . . ??
lmf.src.common.defs . . . ??
lmf.src.common.range . . . ??
lmf.src.config . . . ??
lmf.src.config.mdf . . . ??
lmf.src.config.tex . . . ??
lmf.src.config.xml . . . ??
lmf.src.core . . . ??
lmf.src.core.definition . . . ??
lmf.src.core.form . . . ??
lmf.src.core.form_representation . . . ??
lmf.src.core.global_information . . . ??
lmf.src.core.lexical_entry . . . ??
lmf.src.core.lexical_resource . . . ??
lmf.src.core.lexicon . . . ??
lmf.src.core.representation . . . ??
lmf.src.core.sense . . . ??
lmf.src.core.statement . . . ??
lmf.src.core.text_representation . . . ??
lmf.src.input . . . ??
lmf.src.input.elan . . . ??
lmf.src.input.ite . . . ??
lmf.src.input.mdf . . . ??
lmf.src.input.toolbox_settings . . . ??
lmf.src.input.txt . . . ??
lmf.src.input.xls . . . ??
lmf.src.input.xml_lmf . . . ??
lmf.src.morphology . . . ??
lmf.src.morphology.component . . . ??
lmf.src.morphology.lemma . . . ??
lmf.src.morphology.list_of_components . . . ??
lmf.src.morphology.related_form . . . ??
lmf.src.morphology.stem . . . ??
lmf.src.morphology.word_form . . . ??
lmf.src.morphosyntax . . . ??

lmf.src.morphosyntax.paradigm . . . . . ??
lmf.src.mrd . . . . . ??
lmf.src.mrd.context . . . . . ??
lmf.src.mrd.equivalent . . . . . ??
lmf.src.mrd.subject_field . . . . . ??
lmf.src.output . . . . . ??
lmf.src.output.doc . . . . . ??
lmf.src.output.html . . . . . ??
lmf.src.output.mdf . . . . . ??
lmf.src.output.odt . . . . . ??
lmf.src.output.tex . . . . . ??
lmf.src.output.txt . . . . . ??
lmf.src.output.xls . . . . . ??
lmf.src.output.xml_ite . . . . . ??
lmf.src.output.xml_lift . . . . . ??
lmf.src.output.xml_lmf . . . . . ??
lmf.src.output.xml_lp . . . . . ??
lmf.src.output.xml_olif . . . . . ??
lmf.src.output.xml_tb . . . . . ??
lmf.src.output.xml_tei . . . . . ??
lmf.src.resources . . . . . ??
lmf.src.resources.audio . . . . . ??
lmf.src.resources.human_resource . . . . . ??
lmf.src.resources.material . . . . . ??
lmf.src.resources.picture . . . . . ??
lmf.src.resources.resource . . . . . ??
lmf.src.resources.speaker . . . . . ??
lmf.src.resources.video . . . . . ??
lmf.src.utils . . . . . ??
lmf.src.utils.attr . . . . . ??
lmf.src.utils.error_handling . . . . . ??
lmf.src.utils.io . . . . . ??
lmf.src.utils.ipa2sampa . . . . . ??
lmf.src.utils.ipa2sampa.ipa2sampa . . . . . ??
lmf.src.utils.log . . . . . ??
lmf.src.utils.xml_format . . . . . ??
lmf.src.wrapper . . . . . ??
tables . . . . . ??
uid . . . . . ??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

lmf.src.morphology.component.Component . . . ??
lmf.src.mrd.context.Context . . . ??
lmf.src.core.definition.Definition . . . ??
lmf.src.mrd.equivalent.Equivalent . . . ??
Exception
  lmf.src.utils.error_handling.Error . . . ??
    lmf.src.utils.error_handling.InputError . . . ??
    lmf.src.utils.error_handling.OutputError . . . ??
lmf.src.core.form.Form . . . ??
lmf.src.core.global_information.GlobalInformation . . . ??
lmf.src.core.lexical_entry.LexicalEntry . . . ??
lmf.src.core.lexical_resource.LexicalResource . . . ??
lmf.src.core.lexicon.Lexicon . . . ??
lmf.src.morphology.list_of_components.ListOfComponents . . . ??
lmf.src.morphosyntax.paradigm.Paradigm . . . ??
lmf.src.core.representation.Representation . . . ??
lmf.src.resources.resource.Resource . . . ??
lmf.src.core.sense.Sense . . . ??
lmf.src.core.statement.Statement . . . ??
lmf.src.mrd.subject_field.SubjectField . . . ??
UserWarning
  lmf.src.utils.error_handling.Warning . . . ??
Form
  lmf.src.morphology.lemma.Lemma . . . ??
  lmf.src.morphology.related_form.RelatedForm . . . ??
  lmf.src.morphology.stem.Stem . . . ??
  lmf.src.morphology.word_form.WordForm . . . ??
HumanResource
  lmf.src.resources.speaker.Speaker . . . ??
Material
  lmf.src.resources.audio.Audio . . . ??
  lmf.src.resources.picture.Picture . . . ??
  lmf.src.resources.video.Video . . . ??
Representation
  lmf.src.core.form_representation.FormRepresentation . . . ??
Representation
  lmf.src.core.text_representation.TextRepresentation . . . ??
Resource

lmf.src.resources.human_resource.HumanResource . . . . . ??
lmf.src.resources.material.Material . . . . . ??

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

lmf.src.resources.audio.Audio
: Audio is a Material subclass representing an audio recording . . . . . ??

lmf.src.morphology.component.Component . . . . . ??

lmf.src.mrd.context.Context
: "Context is a class representing a text string that provides authentic context for the use of the word form managed by the Lemma. This class is to be distinguished from Sense Example." (LMF) . . . . . ??

lmf.src.core.definition.Definition
: "Definition is a class representing a narrative description of a sense. It is provided to help human users understand the meaning of a lexical entry. A Sense instance can have zero to many definitions. Each Definition instance may be associated with zero to many Text Representation instances in order to manage the text defintion in more than one language or script. In addition, the narrative description can be expressed in a different language or script than the one in the Lexical Entry instance." (LMF) . . . . . ??

lmf.src.mrd.equivalent.Equivalent
: "Equivalent is a class representing the translation equivalent of the word form managed by the Lemma class." (LMF) . . . . . ??

lmf.src.utils.error_handling.Error
: Base class for exceptions in this library . . . . . ??

lmf.src.core.form.Form
: "Form is an abstract class representing a lexeme, a morphological variant of a lexeme or a morph. The Form class allows subclasses." (LMF) . . . . . ??

lmf.src.core.form_representation.FormRepresentation
: "Form Representation is a class representing one variant orthography of a Form." (LMF) . . . ??

lmf.src.core.global_information.GlobalInformation
: "Global Information is a class for administrative information and other general attributes, such as /language coding/ or /script coding/, which are valid for the entire lexical resource." (LMF) . . . ??

lmf.src.resources.human_resource.HumanResource
: HumanResource is a Resource subclass . . . . . ??

lmf.src.utils.error_handling.InputError
: Exception raised for errors in the input . . . . . ??

lmf.src.morphology.lemma.Lemma
: "Lemma is a Form subclass representing a form chosen by convention to designate the Lexical Entry. The lemma is usually equivalent to one of the inflected forms, the root, stem or compound phrase." (LMF) . . . . . ??

lmf.src.core.lexical_entry.LexicalEntry
"Lexical Entry is a class representing a lexeme in a given language and is a container for managing the Form and Sense classes. A Lexical Entry instance can contain one to many different forms and can have from zero to many different senses." (LMF) . . . . . **??**
lmf.src.core.lexical_resource.LexicalResource
"Lexical Resource is a class representing the entire resource and is a container for one or more lexicons. There is only one Lexical Resource instance." (LMF) . . . . . **??**
lmf.src.core.lexicon.Lexicon
"Lexicon is a class containing all the lexical entries of a given language within the entire resource." (LMF) . . . . . **??**
lmf.src.morphology.list_of_components.ListOfComponents . . . . . **??**
lmf.src.resources.material.Material
Material is a Resource subclass . . . . . **??**
lmf.src.utils.error_handling.OutputError
Exception raised for errors in the output . . . . . **??**
lmf.src.morphosyntax.paradigm.Paradigm
Paradigm is a class representing a morphological paradigm . . . . . **??**
lmf.src.resources.picture.Picture
Picture is a Material subclass representing a picture . . . . . **??**
lmf.src.morphology.related_form.RelatedForm
"Related Form is a Form subclass representing a word form or a morph that can be related to the Lexical Entry. There is no asumption that the Related Form is associated with the Sense class in the Lexical Entry." (LMF) . . . . . **??**
lmf.src.core.representation.Representation
"Representation class is an abstract class representing a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script and orthography." (LMF) **??**
lmf.src.resources.resource.Resource
Resource is an abstract class representing a material or a human resource . . . . . **??**
lmf.src.core.sense.Sense
"Sense is a class representing one meaning of a lexical entry. The Sense class allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry." (LMF) . . . . . **??**
lmf.src.resources.speaker.Speaker
Speaker is a HumanResource subclass . . . . . **??**
lmf.src.core.statement.Statement
"Statement is a class representating a narrative description that refines or complements Definition." (LMF) . . . . . **??**
lmf.src.morphology.stem.Stem
"Stem is a Form subclass representing a morph, thus manages the sublexme parts" (LMF) . . **??**
lmf.src.mrd.subject_field.SubjectField
"Subject Field is a class representing a text string that provides domain or status information." (LMF) . . . . . **??**
lmf.src.core.text_representation.TextRepresentation
"Text Representation is a class representing the textual content of definition or statement. When there is more than one variant orthography, the Text Representation instance contains a Unicode string representing the textual content as well as unique attribute-value pairs that describe the specific language, script and orthography." (LMF) . . . . . **??**
lmf.src.resources.video.Video
Video is a Material subclass representing a video . . . . . **??**
lmf.src.utils.error_handling.Warning
Base class for warnings in this library . . . . . **??**
lmf.src.morphology.word_form.WordForm
"Word Form is a Form subclass representing a form that a lexeme can take when used in a sentence or a phrase." (LMF) . . . . . **??**

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/wrapper.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/defs.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/range.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/mdf.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/tex.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/xml.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/definition.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form_representation.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/global_information.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_entry.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_resource.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexicon.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/representation.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/sense.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/statement.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/text_representation.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/elan.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/ite.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/mdf.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/toolbox_settings.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/txt.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/xls.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/xml_lmf.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/component.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/lemma.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/list_of_components.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/related_form.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/stem.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/word_form.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphosyntax/__init__.py . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphosyntax/paradigm.py . . . ??

/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/__init__.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/context.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/equivalent.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/subject_field.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/__init__.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/doc.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/html.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/mdf.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/odt.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/tex.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/txt.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xls.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_ite.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lift.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lmf.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lp.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_olif.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_tb.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_tei.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/__init__.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/audio.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/human_resource.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/material.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/picture.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/resource.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/speaker.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/video.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/__init__.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/attr.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/io.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/log.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/xml_format.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/eol/eol.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/ipa2sampa/__init__.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/ipa2sampa/ipa2sampa.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/tables/tables.py . . . . . ??
/Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/uid/uid.py . . . . . ??

Chapter 5

Namespace Documentation

5.1 eol Namespace Reference

Variables

- tuple parser = OptionParser()
- tuple options = parser.parse_args()
- tuple in_file = open(options.input, "r", encoding='utf-8')
- tuple out_file = open(options.output, "w", encoding='utf-8')
- string EOL = '\n'
- list lines = []
- tuple previous_line = lines.pop()
- tuple line = previous_line.replace(EOL, " ")

5.1.1 Variable Documentation

5.1.1.1 string eol.EOL = '\n'

Definition at line 22 of file eol.py.

5.1.1.2 tuple eol.in_file = open(options.input, "r", encoding='utf-8')

Definition at line 12 of file eol.py.

5.1.1.3 tuple eol.line = previous_line.replace(EOL, " ")

Definition at line 33 of file eol.py.

5.1.1.4 list eol.lines = []

Definition at line 28 of file eol.py.

5.1.1.5 tuple eol.options = parser.parse_args()

Definition at line 8 of file eol.py.

5.1.1.6 tuple eol.out_file = open(options.output, "w", encoding='utf-8')

Definition at line 13 of file eol.py.

5.1.1.7 tuple eol.parser = OptionParser()

Definition at line 5 of file eol.py.

5.1.1.8 tuple eol.previous_line = lines.pop()

Definition at line 32 of file eol.py.

5.2 lmf Namespace Reference

Namespaces

- src

5.3 lmf.src Namespace Reference

Namespaces

- common
- config
- core
- input
- morphology
- morphosyntax
- mrd
- output
- resources
- utils
- wrapper

5.4 lmf.src.common Namespace Reference

Namespaces

- defs
- range

5.5 lmf.src.common.defs Namespace Reference

Variables

- string VERNACULAR = "vernacular"

  Define languages.
- string ENGLISH = "English"
- string NATIONAL = "national"
- string REGIONAL = "regional"

5.5.1 Variable Documentation

5.5.1.1 string lmf.src.common.defs.ENGLISH = "English"

Definition at line 5 of file defs.py.

5.5.1.2 string lmf.src.common.defs.NATIONAL = "national"

Definition at line 6 of file defs.py.

5.5.1.3 string lmf.src.common.defs.REGIONAL = "regional"

Definition at line 7 of file defs.py.

5.5.1.4 string lmf.src.common.defs.VERNACULAR = "vernacular"

Define languages.

Definition at line 4 of file defs.py.

5.6 lmf.src.common.range Namespace Reference

Variables

- tuple partOfSpeech_range

  Possible values allowed for LMF part of speech LexicalEntry attribute.
- tuple type_variant_range

  Possible values allowed for LMF variant type FormRepresentation attribute.
- tuple noteType_range

  Possible values allowed for LMF note type Statement attribute.
- tuple grammaticalNumber_range

  Possible values allowed for LMF grammatical number WordForm attribute.
- tuple grammaticalGender_range

  Possible values allowed for LMF grammatical gender WordForm attribute.
- tuple person_range

  Possible values allowed for LMF grammatical person WordForm attribute.
- tuple anymacy_range

  Possible values allowed for LMF anymacy WordForm attribute.
- tuple clusivity_range

  Possible values allowed for LMF clusivity WordForm attribute.
- tuple tense_range

  Possible values allowed for LMF grammatical tense WordForm attribute.
- tuple voice_range

  Possible values allowed for LMF voice WordForm attribute.
- tuple verbFormMood_range

  Possible values allowed for LMF verb form mood WordForm attribute.
- tuple degree_range

  Possible values allowed for LMF degree WordForm attribute.
- tuple semanticRelation_range

  Possible values allowed for semantic relation RelatedForm attribute.

- tuple paradigmLabel_range

  Possible values allowed for paradigm label Paradigm attribute.
- tuple type_example_range

  Possible values allowed for example type Context attribute.
- tuple mediaType_range

  Possible values allowed for media type Material attribute.
- tuple quality_range

  Possible values allowed for quality Audio attribute.

5.6.1 Variable Documentation

5.6.1.1 tuple lmf.src.common.range.anymacy_range

Initial value:

```
1 = set([
2     "animate",
3     "inanimate",
4     "other anymacy"
5 ])
```

Possible values allowed for LMF anymacy WordForm attribute.

Definition at line 110 of file range.py.

5.6.1.2 tuple lmf.src.common.range.clusivity_range

Initial value:

```
1 = set([
2     "inclusive",
3     "exclusive"
4 ])
```

Possible values allowed for LMF clusivity WordForm attribute.

Definition at line 117 of file range.py.

5.6.1.3 tuple lmf.src.common.range.degree_range

Initial value:

```
1 = set([
2     "comparative degree",
3     "positive degree",
4     "superlative degree"
5 ])
```

Possible values allowed for LMF degree WordForm attribute.

Definition at line 152 of file range.py.

5.6.1.4 tuple lmf.src.common.range.grammaticalGender_range

Initial value:

```
1 = set([
2     "common gender",
3     "feminine",
4     "masculine",
5     "neuter"
6 ])
```

Possible values allowed for LMF grammatical gender WordForm attribute.

Definition at line 95 of file range.py.

5.6.1.5 tuple lmf.src.common.range.grammaticalNumber_range

Initial value:

```
1 = set([
2     "collective",
3     "dual",
4     "paucal",
5     "plural",
6     "quadrial",
7     "singular",
8     "trial"
9 ])
```

Possible values allowed for LMF grammatical number WordForm attribute.

Definition at line 84 of file range.py.

5.6.1.6 tuple lmf.src.common.range.mediaType_range

Initial value:

```
1 = set([
2     "unspecified",
3     "unknown",
4     "audio",
5     "video",
6     "document",
7     "text",
8     "image",
9     "drawing"
10 ])
```

Possible values allowed for media type Material attribute.

Definition at line 195 of file range.py.

5.6.1.7 tuple lmf.src.common.range.noteType_range

Initial value:

```
1 = set([
2     "comparison",
3     "history",
4     "semantics",
5     "tone",
6     "derivation",
7     "case",
8     "subord",
9     "usage",
10     "comment",
11     "legend",
12     "restriction",
13     "encyclopedic",
14     "anthropology",
15     "discourse",
16     "grammar",
17     "phonology",
18     "question",
19     "sociolinguistics",
20     "general"
21 ])
```

Possible values allowed for LMF note type Statement attribute.

Definition at line 61 of file range.py.

5.6.1.8 tuple lmf.src.common.range.paradigmLabel_range

Initial value:

```
1 = set([
2     "lexicalized affix",
3     "conjugation class",
4     "past stem",
5     "comitative", "COM", # comitative (Leipzig)
6     "construction",
7     "directional",
8     "irregularity",
9     "classifier"
10 ])
```

Possible values allowed for paradigm label Paradigm attribute.

Definition at line 175 of file range.py.

5.6.1.9 tuple lmf.src.common.range.partOfSpeech_range

Possible values allowed for LMF part of speech LexicalEntry attribute.

Definition at line 4 of file range.py.

5.6.1.10 tuple lmf.src.common.range.person_range

Initial value:

```
1 = set([
2     "first person",
3     "second person",
4     "third person"
5 ])
```

Possible values allowed for LMF grammatical person WordForm attribute.

Definition at line 103 of file range.py.

5.6.1.11 tuple lmf.src.common.range.quality_range

Initial value:

```
1 = set([
2     "very low",
3     "low",
4     "normal",
5     "good",
6     "very good" # high
7 ])
```

Possible values allowed for quality Audio attribute.

Definition at line 207 of file range.py.

5.6.1.12 tuple lmf.src.common.range.semanticRelation_range

Initial value:

```
1 = set([
2     "synonym",
3     "antonym",
4     "homonym",
5     "etymology",
6     "subentry",
```

```
7     "main entry",
8     "simple link",
9     "complex predicate",
10     "derived form",
11     "root",
12     "stem",
13     "collocation"
14 ])
```

Possible values allowed for semantic relation RelatedForm attribute.

Definition at line 159 of file range.py.

5.6.1.13 tuple lmf.src.common.range.tense_range

Initial value:

```
1 = set([
2     "future",
3     "imperfect",
4     "past",
5     "present"
6 ])
```

Possible values allowed for LMF grammatical tense WordForm attribute.

Definition at line 123 of file range.py.

5.6.1.14 tuple lmf.src.common.range.type_example_range

Initial value:

```
1 = set([
2     "proverb",
3     "locution",
4     "example",
5     "combination"
6 ])
```

Possible values allowed for example type Context attribute.

Definition at line 187 of file range.py.

5.6.1.15 tuple lmf.src.common.range.type_variant_range

Initial value:

```
1 = set([
2     "unspecified",
3     "orthography",
4     "phonetics",
5     "archaic"
6 ])
```

Possible values allowed for LMF variant type FormRepresentation attribute.

Definition at line 53 of file range.py.

5.6.1.16 tuple lmf.src.common.range.verbFormMood_range

Initial value:

```
1 = set([
2     "gerundive",
3     "imperative",
4     "indicative",
5     "infinitive",
6     "participle",
7     "subjunctive",
8     "conditional",
9     "relative mood",
10     "prohibitive mood",
11     "debitive mood"
12 ])
```

Possible values allowed for LMF verb form mood WordForm attribute.

Definition at line 138 of file range.py.

5.6.1.17 tuple lmf.src.common.range.voice_range

Initial value:

```
1 = set([
2     "active voice",
3     "middle voice",
4     "passive voice"
5 ])
```

Possible values allowed for LMF voice WordForm attribute.

Definition at line 131 of file range.py.

5.7 lmf.src.config Namespace Reference

Namespaces

- mdf
- tex
- xml

5.8 lmf.src.config.mdf Namespace Reference

Functions

- def set_bw

  Functions to process some MDF fields (input)
- def get_bw

  Functions to process some MDF fields (output)

Variables

- tuple mdf_lmf

  Mapping between MDF markers and LMF representation (input)
- list mdf_order

  Order in which MDF markers must be written (output) This is the standard order defined in Appendix B of "Making Dictionaries. A guide to lexicography and the Multi-Dictionary Formatter", Software version 1.0, David F.
- tuple lmf_mdf

  Mapping between LMF representation and MDF markers (output)

- tuple ps_range

  Possible values allowed for 'ps' MDF marker.
- tuple ps_partOfSpeech

  Mapping between 'ps' MDF marker value and LMF part of speech LexicalEntry attribute value (input) Source: http://www.isocat.org/rest/dcs/119.
- tuple mdf_semanticRelation

  Mapping between MDF markers and LMF semantic relation RelatedForm attribute value (input)
- tuple pd_person

  Mapping between 'pd' MDF markers and LMF person WordForm attribute value (input)
- tuple pd_anymacy

  Mapping between 'pd' MDF markers and LMF anymacy WordForm attribute value (input)
- tuple pd_grammaticalNumber

  Mapping between 'pd' MDF markers and LMF grammatical number WordForm attribute value (input)
- tuple pd_clusivity

  Mapping between 'pd' MDF markers and LMF clusivity WordForm attribute value (input)
- tuple pdl_paradigmLabel

  Mapping between 'pdl' MDF marker value and LMF paradigm label Paradigm attribute value (input)
- tuple sd_range

  Possible values allowed for 'sd' MDF marker.
- tuple lf_range

  Possible values allowed for 'lf' MDF marker.

5.8.1 Function Documentation

5.8.1.1 def lmf.src.config.mdf.get_bw (*lexical_entry*)

Functions to process some MDF fields (output)

Definition at line 245 of file mdf.py.

5.8.1.2 def lmf.src.config.mdf.set_bw (*bw, lexical_entry*)

Functions to process some MDF fields (input)

Definition at line 7 of file mdf.py.

5.8.2 Variable Documentation

5.8.2.1 tuple lmf.src.config.mdf.lf_range

Possible values allowed for 'lf' MDF marker.

Definition at line 632 of file mdf.py.

5.8.2.2 tuple lmf.src.config.mdf.lmf_mdf

Mapping between LMF representation and MDF markers (output)

Definition at line 252 of file mdf.py.

5.8.2.3 tuple lmf.src.config.mdf.mdf_lmf

Mapping between MDF markers and LMF representation (input)

Definition at line 19 of file mdf.py.

5.8.2.4 list lmf.src.config.mdf.mdf_order

Order in which MDF markers must be written (output) This is the standard order defined in Appendix B of "Making Dictionaries. A guide to lexicography and the Multi-Dictionary Formatter", Software version 1.0, David F.

Coward, Charles E. Grimes, SIL International, Waxhaw, North Carolina, 2000

Definition at line 128 of file mdf.py.

5.8.2.5 tuple lmf.src.config.mdf.mdf_semanticRelation

Initial value:

```
1 = dict({
2     "sy" : "synonym",
3     "an" : "antonym",
4     "hm" : "homonym",
5     "et" : "etymology",
6     "se" : "subentry",
7     "mn" : "main entry",
8     "cf" : "simple link",
9     "cp" : "complex predicate",
10     "lf" : None,
11     "ev" : None,
12     "ee" : None,
13     "en" : None,
14     "er" : None
15     # "derived form",
16     # "root",
17     # "stem",
18     # "collocation"
19 })
```

Mapping between MDF markers and LMF semantic relation RelatedForm attribute value (input)

Definition at line 512 of file mdf.py.

5.8.2.6 tuple lmf.src.config.mdf.pd_anymacy

Initial value:

```
1 = dict({
2     4 : "inanimate"
3 })
```

Mapping between 'pd' MDF markers and LMF anymacy WordForm attribute value (input)

Definition at line 540 of file mdf.py.

5.8.2.7 tuple lmf.src.config.mdf.pd_clusivity

Initial value:

```
1 = dict({
2     'i'     : "inclusive",
3     'e'     : "exclusive"
4 })
```

Mapping between 'pd' MDF markers and LMF clusivity WordForm attribute value (input)

Definition at line 554 of file mdf.py.

5.8.2.8 tuple lmf.src.config.mdf.pd_grammaticalNumber

Initial value:

```
1 = dict({
2     'd'     : "dual",
3     'p'     : "plural",
4     "pl"    : "plural",
5     's'     : "singular",
6     "sg"    : "singular"
7 })
```

Mapping between 'pd' MDF markers and LMF grammatical number WordForm attribute value (input)

Definition at line 545 of file mdf.py.

5.8.2.9 tuple lmf.src.config.mdf.pd_person

Initial value:

```
1 = dict({
2     1 : "first person",
3     2 : "second person",
4     3 : "third person"
5 })
```

Mapping between 'pd' MDF markers and LMF person WordForm attribute value (input)

Definition at line 533 of file mdf.py.

5.8.2.10 tuple lmf.src.config.mdf.pdl_paradigmLabel

Initial value:

```
1 = dict({
2     "la"        : "lexicalized affix",
3     "cc"        : "conjugation class",
4     "past"      : "past stem",
5     "comit"     : "comitative",
6     "constr"    : "construction",
7     "dir"       : "directional",
8     "ir"        : "irregularity",
9     "cl"        : "classifier"
10 })
```

Mapping between 'pdl' MDF marker value and LMF paradigm label Paradigm attribute value (input)

Definition at line 560 of file mdf.py.

5.8.2.11 tuple lmf.src.config.mdf.ps_partOfSpeech

Mapping between 'ps' MDF marker value and LMF part of speech LexicalEntry attribute value (input) Source↩ : http://www.isocat.org/rest/dcs/119.

Definition at line 438 of file mdf.py.

5.8.2.12 tuple lmf.src.config.mdf.ps_range

Possible values allowed for 'ps' MDF marker.

Definition at line 363 of file mdf.py.

5.8.2.13 tuple lmf.src.config.mdf.sd_range

Possible values allowed for 'sd' MDF marker.

Definition at line 572 of file mdf.py.

5.9 lmf.src.config.tex Namespace Reference

Functions

- def lmf_to_tex

  Function giving order in which information must be written in LaTeX and mapping between LMF representation and LaTeX (output)

Variables

- tuple partOfSpeech_tex

  Mapping between LMF part of speech LexicalEntry attribute value and LaTeX layout (output)
- tuple paradigmLabel_tex

  Mapping between LMF paradigmLabel Paradigm attribute value and LaTeX layout (output)

5.9.1 Function Documentation

5.9.1.1 def lmf.src.config.tex.lmf_to_tex (*lexical_entry*, *font* = `None`, *partOfSpeech_mapping* = `partOfSpeech_tex`, *languages* = `[VERNACULAR`, *ENGLISH*, *NATIONAL*, *REGIONAL*)

Function giving order in which information must be written in LaTeX and mapping between LMF representation and LaTeX (output)

Function to convert LMF lexical entry information to be written into LaTeX commands.

Parameters

lexical_entry	The Lexical Entry LMF instance to display.
font	A Python dictionary describing fonts to use for different languages.
partOfSpeech_mapping	A Python dictionary giving abbreviations for LMF part of speech values.
languages	A list of languages to consider for LaTeX layout (all by default).

Returns

A string representing the lexical entry in LaTeX format.

Definition at line 59 of file tex.py.

5.9.2 Variable Documentation

5.9.2.1 tuple lmf.src.config.tex.paradigmLabel_tex

Initial value:

```
1 = dict({
2 })
```

Mapping between LMF paradigmLabel Paradigm attribute value and LaTeX layout (output)

Definition at line 55 of file tex.py.

5.9.2.2 tuple lmf.src.config.tex.partOfSpeech_tex

Mapping between LMF part of speech LexicalEntry attribute value and LaTeX layout (output)

Definition at line 8 of file tex.py.

5.10 lmf.src.config.xml Namespace Reference

Functions

- def sort_order_read

  Read an XML file giving sort order.
- def config_read

  Read an XML file giving the user configuration.

5.10.1 Function Documentation

5.10.1.1 def lmf.src.config.xml.config_read (*filename*)

Read an XML file giving the user configuration.

Parameters

filename	The name of the XML file to read with full path, for instance 'user/default/config.xml'.

Returns

A Lexical Resource.

Definition at line 45 of file xml.py.

5.10.1.2 def lmf.src.config.xml.sort_order_read (*filename*)

Read an XML file giving sort order.

Parameters

filename	The name of the XML file to read with full path, for instance 'user/default/sort_order.xml'.

Returns

A Python dictionary of ordered characters.

Definition at line 19 of file xml.py.

5.11 lmf.src.core Namespace Reference

Namespaces

- definition
- form
- form_representation
- global_information
- lexical_entry
- lexical_resource
- lexicon
- representation
- sense
- statement
- text_representation

5.12 lmf.src.core.definition Namespace Reference

Classes

- class Definition

  "Definition is a class representing a narrative description of a sense. It is provided to help human users understand the meaning of a lexical entry. A Sense instance can have zero to many definitions. Each Definition instance may be associated with zero to many Text Representation instances in order to manage the text defintion in more than one language or script. In addition, the narrative description can be expressed in a different language or script than the one in the Lexical Entry instance." (LMF)

5.13 lmf.src.core.form Namespace Reference

Classes

- class Form

  "Form is an abstract class representing a lexeme, a morphological variant of a lexeme or a morph. The Form class allows subclasses." (LMF)

5.14 lmf.src.core.form_representation Namespace Reference

Classes

- class FormRepresentation

  "Form Representation is a class representing one variant orthography of a Form." (LMF)

5.15 lmf.src.core.global_information Namespace Reference

Classes

- class GlobalInformation

  "Global Information is a class for administrative information and other general attributes, such as /language coding/ or /script coding/, which are valid for the entire lexical resource." (LMF)

5.16 lmf.src.core.lexical_entry Namespace Reference

Classes

- class LexicalEntry

  "Lexical Entry is a class representing a lexeme in a given language and is a container for managing the Form and Sense classes. A Lexical Entry instance can contain one to many different forms and can have from zero to many different senses." (LMF)

5.17 lmf.src.core.lexical_resource Namespace Reference

Classes

- class LexicalResource

"Lexical Resource is a class representing the entire resource and is a container for one or more lexicons. There is only one Lexical Resource instance." (LMF)

5.18 lmf.src.core.lexicon Namespace Reference

Classes

- class Lexicon

  "Lexicon is a class containing all the lexical entries of a given language within the entire resource." (LMF)

5.19 lmf.src.core.representation Namespace Reference

Classes

- class Representation

  "Representation class is an abstract class representing a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

5.20 lmf.src.core.sense Namespace Reference

Classes

- class Sense

  "Sense is a class representing one meaning of a lexical entry. The Sense class allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry." (LMF)

5.21 lmf.src.core.statement Namespace Reference

Classes

- class Statement

  "Statement is a class representating a narrative description that refines or complements Definition." (LMF)

5.22 lmf.src.core.text_representation Namespace Reference

Classes

- class TextRepresentation

  "Text Representation is a class representing the textual content of definition or statement. When there is more than one variant orthography, the Text Representation instance contains a Unicode string representing the textual content as well as unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

5.23 lmf.src.input Namespace Reference

Namespaces

- elan

- ite
- mdf
- toolbox_settings
- txt
- xls
- xml_lmf

5.24 lmf.src.input.elan Namespace Reference

5.25 lmf.src.input.ite Namespace Reference

5.26 lmf.src.input.mdf Namespace Reference

Functions

- def mdf_read

  Read an MDF file.

5.26.1 Function Documentation

5.26.1.1 def lmf.src.input.mdf.mdf_read (*filename* = None, *mdf2lmf* = mdf_lmf, *lexicon* = None, *id* = None, *encoding* = ENCODING)

Read an MDF file.

Parameters

filename	The name of the MDF file to read with full path, for instance 'user/input.txt'.
mdf2lmf	A Python dictionary describing the mapping between MDF markers and LMF representation. Default value is 'mdf_lmf' dictionary defined in 'src/config/mdf.py'. Please refer to it as an example.
lexicon	An existing Lexicon to fill with lexical entries to read.
id	A Python string identifying the lexicon to create.
encoding	Use 'utf-8' encoding by default. Otherwise, user has to precise the native encoding of its document.

Returns

A Lexicon instance containing all lexical entries.

Definition at line 13 of file mdf.py.

5.27 lmf.src.input.toolbox_settings Namespace Reference

5.28 lmf.src.input.txt Namespace Reference

5.29 lmf.src.input.xls Namespace Reference

5.30 lmf.src.input.xml_lmf Namespace Reference

Functions

- def compute_name

  Compute attribute/module name from object name as follows: 'ObjectName' attribute/module name is 'object_name'.
- def factory

  This function is an object factory.
- def xml_lmf_read

  Read an XML LMF file.
- def get_sub_elements

  This function recursively parses the given XML element and creates corresponding LMF instances with their attributes.

5.30.1 Function Documentation

5.30.1.1 def lmf.src.input.xml_lmf.compute_name (*object_name*)

Compute attribute/module name from object name as follows: 'ObjectName' attribute/module name is 'object_↩ name'.

Parameters

object_name	String containing name of the object, e.g. 'LexicalEntry'.

Returns

The corresponding attribute/module name, e.g. 'lexical_entry'.

Definition at line 6 of file xml_lmf.py.

5.30.1.2 def lmf.src.input.xml_lmf.factory (*object_name, attributes*)

This function is an object factory.

Indeed, from an object name and its attributes, it creates a Python object and sets its attributes.

Parameters

object_name	A Python string containing the object name, for instance 'LexicalEntry'.
attributes	A Python dictionary containing pairs of attribute name (as a Python string) and value, for instance {'partOfSpeech': 'n'}.

Definition at line 21 of file xml_lmf.py.

5.30.1.3 def lmf.src.input.xml_lmf.get_sub_elements (*instance, element*)

This function recursively parses the given XML element and creates corresponding LMF instances with their attributes.

Parameters

instance	An LMF object instance.
element	An XML element.

Definition at line 69 of file xml_lmf.py.

5.30.1.4 def lmf.src.input.xml_lmf.xml_lmf_read (*filename*)

Read an XML LMF file.

Parameters

filename	The name of the XML LMF file to read with full path, for instance 'user/input.xml'.

Returns

A Lexical Resource instance containing all lexicons.

Definition at line 57 of file xml_lmf.py.

5.31 lmf.src.morphology Namespace Reference

Namespaces

- component
- lemma
- list_of_components
- related_form
- stem
- word_form

5.32 lmf.src.morphology.component Namespace Reference

Classes

- class Component

5.33 lmf.src.morphology.lemma Namespace Reference

Classes

- class Lemma

  "Lemma is a Form subclass representing a form chosen by convention to designate the Lexical Entry. The lemma is usually equivalent to one of the inflected forms, the root, stem or compound phrase." (LMF).

5.34 lmf.src.morphology.list_of_components Namespace Reference

Classes

- class ListOfComponents

5.35 lmf.src.morphology.related_form Namespace Reference

Classes

- class RelatedForm

  "Related Form is a Form subclass representing a word form or a morph that can be related to the Lexical Entry. There is no asumption that the Related Form is associated with the Sense class in the Lexical Entry." (LMF)

5.36 lmf.src.morphology.stem Namespace Reference

Classes

- class Stem

  "Stem is a Form subclass representing a morph, thus manages the sublexme parts" (LMF)

5.37 lmf.src.morphology.word_form Namespace Reference

Classes

- class WordForm

  "Word Form is a Form subclass representing a form that a lexeme can take when used in a sentence or a phrase." (LMF)

5.38 lmf.src.morphosyntax Namespace Reference

Namespaces

- paradigm

5.39 lmf.src.morphosyntax.paradigm Namespace Reference

Classes

- class Paradigm

  Paradigm is a class representing a morphological paradigm.

5.40 lmf.src.mrd Namespace Reference

Namespaces

- context
- equivalent
- subject_field

5.41 lmf.src.mrd.context Namespace Reference

Classes

- class Context

  "Context is a class representing a text string that provides authentic context for the use of the word form managed by the Lemma. This class is to be distinguished from Sense Example." (LMF)

5.42 lmf.src.mrd.equivalent Namespace Reference

Classes

- class Equivalent

  "Equivalent is a class representing the translation equivalent of the word form managed by the Lemma class." (LMF)

5.43 lmf.src.mrd.subject_field Namespace Reference

Classes

- class SubjectField

  "Subject Field is a class representing a text string that provides domain or status information." (LMF)

5.44 lmf.src.output Namespace Reference

Namespaces

- doc
- html
- mdf
- odt
- tex
- txt
- xls
- xml_ite
- xml_lift
- xml_lmf
- xml_lp
- xml_olif
- xml_tb
- xml_tei

5.45 lmf.src.output.doc Namespace Reference

Functions

- def doc_write

  Write a document file.

5.45.1 Function Documentation

5.45.1.1 def lmf.src.output.doc.doc_write (*object*, *filename*, *items* = `lambda lexical_entry: lexical_entry.get_lexeme()`, *sort_order* = `None`, *paradigms* = `False`, *reverse* = `False`)

Write a document file.

Parameters

object	The LMF instance to convert into document output format.
filename	The name of the document file to write with full path, for instance 'user/output.doc'.
items	Lambda function giving the item to sort. Default value is 'lambda lexical_entry: lexical_↩ entry.get_lexeme()', which means that the items to sort are lexemes.
sort_order	Python list. Default value is 'None', which means that the document output is alphabetically ordered.

Definition at line 14 of file doc.py.

5.46 lmf.src.output.html Namespace Reference

5.47 lmf.src.output.mdf Namespace Reference

Functions

- def mdf_write

  Write an MDF file.
- def parse_list

  Parse a group of markers and write them into an MDF file.
- def write_line

  Write a line into an MDF file.

5.47.1 Function Documentation

5.47.1.1 def lmf.src.output.mdf.mdf_write (*object, filename, lmf2mdf =* `lmf_mdf`*, order =* `mdf_order`)

Write an MDF file.

Parameters

object	The LMF instance to convert into MDF output format.
filename	The name of the MDF file to write with full path, for instance 'user/output.txt'.
lmf2mdf	A Python dictionary describing the mapping between LMF representation and MDF markers. Default value is 'lmf_mdf' dictionary defined in 'src/config/mdf.py'. Please refer to it as an example.
order	A Python list defining the order in which MDF markers must be written, for instance ["lx", "ps"]. Default value is 'mdf_order' list defined in 'src/config/mdf.py'.

Definition at line 7 of file mdf.py.

5.47.1.2 def lmf.src.output.mdf.parse_list (*mdf_file, lmf2mdf, marker, object*)

Parse a group of markers and write them into an MDF file.

Parameters

mdf_file	The file to write in.
lmf2mdf	A Python dictionary describing the mapping between LMF representation and MDF markers.
marker	The MDF marker.

object	The current processed object.

Definition at line 31 of file mdf.py.

5.47.1.3 def lmf.src.output.mdf.write_line (*mdf_file, marker, value*)

Write a line into an MDF file.

Parameters

mdf_file	The file to write in.
marker	The MDF marker.
value	The corresponding value.

Definition at line 49 of file mdf.py.

5.48 lmf.src.output.odt Namespace Reference

Variables

- tuple textdoc = OpenDocumentText()
- s = textdoc.styles
- tuple h1style = Style(name="Heading 1", family="paragraph")
- tuple boldstyle = Style(name="Bold", family="text")
- tuple boldprop = TextProperties(fontweight="bold", fontname="Arial", fontsize="8pt")
- tuple h = H(outlinelevel=1, stylename=h1style, text="My first text")
- tuple p = P(text="Hello world. ")
- tuple boldpart = Span(stylename=boldstyle, text="This part is bold. ")

5.48.1 Variable Documentation

5.48.1.1 tuple lmf.src.output.odt.boldpart = Span(stylename=**boldstyle**, text="This part is bold. ")

Definition at line 22 of file odt.py.

5.48.1.2 tuple lmf.src.output.odt.boldprop = TextProperties(fontweight="bold", fontname="Arial", fontsize="8pt")

Definition at line 15 of file odt.py.

5.48.1.3 tuple lmf.src.output.odt.boldstyle = Style(name="Bold", family="text")

Definition at line 14 of file odt.py.

5.48.1.4 tuple lmf.src.output.odt.h = H(outlinelevel=1, stylename=**h1style**, text="My first text")

Definition at line 19 of file odt.py.

5.48.1.5 tuple lmf.src.output.odt.h1style = Style(name="Heading 1", family="paragraph")

Definition at line 10 of file odt.py.

5.48.1.6 tuple lmf.src.output.odt.p = P(text="Hello world. ")

Definition at line 21 of file odt.py.

5.48.1.7 lmf.src.output.odt.s = textdoc.styles

Definition at line 9 of file odt.py.

5.48.1.8 tuple lmf.src.output.odt.textdoc = OpenDocumentText()

Definition at line 7 of file odt.py.

5.49 lmf.src.output.tex Namespace Reference

Functions

- def file_read

  Read file contents.
- def insert_references

  Insert references to paradigms.
- def tex_write

  Write a LaTeX file.
- def handle_font

  Functions to process LaTeX layout.
- def handle_reserved
- def handle_fi
- def handle_fv
- def handle_fn
- def handle_pinyin
- def handle_caps
- def handle_quotes
- def format_uid

  Functions to process LaTeX fields (output)
- def format_link

  Display hyperlink to a lexical entry in LaTeX format.
- def format_lexeme

  'lx', 'hm' and 'lc' fields are flipped if 'lc' field has data.
- def format_audio

  Embed sound file into PDF.
- def format_part_of_speech

  Display part of speech in LaTeX format.
- def format_definitions

  Glosses are supplanted by definitions.
- def format_lt

  Display 'lt' in LaTeX format.
- def format_sc

  Display 'sc' in LaTeX format.
- def format_rf

  Display 'rf' in LaTeX format.
- def format_examples

Display examples in LaTeX format.

- def format_usage_notes

  Display usage notes in LaTeX format.
- def format_encyclopedic_informations

  Display encyclopedic informations in LaTeX format.
- def format_restrictions

  Display restrictions in LaTeX format.
- def format_lexical_functions

  Display lexical functions in LaTeX format.
- def format_related_forms

  Display related forms in LaTeX format.
- def format_variant_forms

  Display variant forms in LaTeX format.
- def format_borrowed_word

  Display borrowed word in LaTeX format.
- def format_etymology

  Display etymology in LaTeX format.
- def format_paradigms

  Display all paradigms in LaTeX format.
- def format_table

  Display a table in LaTeX format.
- def format_semantic_domains

  Display semantic domains in LaTeX format.
- def format_bibliography

  Display bibliography in LaTeX format.
- def format_picture

  Display a picture in LaTeX format.
- def format_notes

  Display all notes in LaTeX format.
- def format_source

  Display source in LaTeX format.
- def format_status

  Display status in LaTeX format.
- def format_date

  Do not display date in LaTeX format.

5.49.1 Function Documentation

5.49.1.1 def lmf.src.output.tex.file_read (*filename*)

Read file contents.

Parameters

filename	The name of the file with full path containing information to read, for instance the LaTeX header of the document: 'user/config/japhug.tex'.

Returns

A Python string containing read information.

Definition at line 13 of file tex.py.

5.49.1.2 def lmf.src.output.tex.format_audio (*lexical_entry, font*)

Embed sound file into PDF.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string embedding sound in LaTeX format.

Definition at line 346 of file tex.py.

5.49.1.3 def lmf.src.output.tex.format_bibliography (*lexical_entry, font*)

Display bibliography in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing bibliography in LaTeX format.

Definition at line 728 of file tex.py.

5.49.1.4 def lmf.src.output.tex.format_borrowed_word (*lexical_entry, font*)

Display borrowed word in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing borrowed word in LaTeX format.

Definition at line 634 of file tex.py.

5.49.1.5 def lmf.src.output.tex.format_date (*lexical_entry, font*)

Do not display date in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

An empty string.

Definition at line 791 of file tex.py.

5.49.1.6 def lmf.src.output.tex.format_definitions (*sense, font, languages* = None)

Glosses are supplanted by definitions.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.
languages	A list of languages to consider for definitions and glosses (all by default).

Returns

A string representing glosses and definitions in LaTeX format.

Definition at line 421 of file tex.py.

5.49.1.7 def lmf.src.output.tex.format_encyclopedic_informations (*sense, font*)

Display encyclopedic informations in LaTeX format.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing encyclopedic informations in LaTeX format.

Definition at line 533 of file tex.py.

5.49.1.8 def lmf.src.output.tex.format_etymology (*lexical_entry, font*)

Display etymology in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing etymology in LaTeX format.

Definition at line 648 of file tex.py.

5.49.1.9 def lmf.src.output.tex.format_examples (*sense, font, languages* = None)

Display examples in LaTeX format.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.
languages	A list of languages to consider for examples (all by default).

Returns

A string representing examples in LaTeX format.

Definition at line 489 of file tex.py.

5.49.1.10 def lmf.src.output.tex.format_lexeme (*lexical_entry, font*)

'lx', 'hm' and 'lc' fields are flipped if 'lc' field has data.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing lexeme in LaTeX format.

Definition at line 318 of file tex.py.

5.49.1.11 def lmf.src.output.tex.format_lexical_functions (*lexical_entry, font*)

Display lexical functions in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing lexical functions in LaTeX format.

Definition at line 567 of file tex.py.

5.49.1.12 def lmf.src.output.tex.format_link (*lexical_entry, font*)

Display hyperlink to a lexical entry in LaTeX format.

Parameters

lexical_entry	The targeted Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing the hyperlink in LaTeX format.

Definition at line 306 of file tex.py.

5.49.1.13 def lmf.src.output.tex.format_lt (*sense, font*)

Display 'lt' in LaTeX format.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing 'lt' in LaTeX format.

Definition at line 462 of file tex.py.

5.49.1.14 def lmf.src.output.tex.format_notes (*lexical_entry, font*)

Display all notes in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing all notes in LaTeX format.

Definition at line 748 of file tex.py.

5.49.1.15 def lmf.src.output.tex.format_paradigms (*lexical_entry, font*)

Display all paradigms in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing all paradigms in LaTeX format.

Definition at line 661 of file tex.py.

5.49.1.16 def lmf.src.output.tex.format_part_of_speech (*lexical_entry, font, mapping* = `partOfSpeech_tex`, *language* = `None`)

Display part of speech in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.
mapping	A Python dictionary giving the mapping between LMF part of speech LexicalEntry attribute value and LaTeX layout.
language	Language to consider to display part of speech.

Returns

A string representing part of speech in LaTeX format.

Definition at line 402 of file tex.py.

5.49.1.17 def lmf.src.output.tex.format_picture (*lexical_entry, font*)

Display a picture in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing a picture in LaTeX format.

Definition at line 739 of file tex.py.

5.49.1.18 def lmf.src.output.tex.format_related_forms (*lexical_entry, font*)

Display related forms in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing related forms in LaTeX format.

Definition at line 580 of file tex.py.

5.49.1.19 def lmf.src.output.tex.format_restrictions (*sense, font*)

Display restrictions in LaTeX format.

Parameters

sense	The current Sense LMF instance.

font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing restrictions in LaTeX format.

Definition at line 550 of file tex.py.

5.49.1.20 def lmf.src.output.tex.format_rf (*sense, font*)

Display 'rf' in LaTeX format.

Parameters

lexical_entry	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing 'rf' in LaTeX format.

Definition at line 480 of file tex.py.

5.49.1.21 def lmf.src.output.tex.format_sc (*sense, font*)

Display 'sc' in LaTeX format.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing 'sc' in LaTeX format.

Definition at line 471 of file tex.py.

5.49.1.22 def lmf.src.output.tex.format_semantic_domains (*lexical_entry, font*)

Display semantic domains in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing semantic domains in LaTeX format.

Definition at line 713 of file tex.py.

5.49.1.23 def lmf.src.output.tex.format_source (*lexical_entry, font*)

Display source in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing source in LaTeX format.

Definition at line 771 of file tex.py.

5.49.1.24 def lmf.src.output.tex.format_status (*lexical_entry, font*)

Display status in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing status in LaTeX format.

Definition at line 780 of file tex.py.

5.49.1.25 def lmf.src.output.tex.format_table (*lexical_entry, font*)

Display a table in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing a table in LaTeX format.

Definition at line 705 of file tex.py.

5.49.1.26 def lmf.src.output.tex.format_uid (*lexical_entry, font*)

Functions to process LaTeX fields (output)

Transform unique identifier of a lexical entry in ASCII format.

Parameters

lexical_entry	The targeted Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing the unique identifier in LaTeX format.

Definition at line 290 of file tex.py.

5.49.1.27 def lmf.src.output.tex.format_usage_notes (*sense, font*)

Display usage notes in LaTeX format.

Parameters

sense	The current Sense LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing usage notes in LaTeX format.

Definition at line 516 of file tex.py.

5.49.1.28 def lmf.src.output.tex.format_variant_forms (*lexical_entry, font*)

Display variant forms in LaTeX format.

Parameters

lexical_entry	The current Lexical Entry LMF instance.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.

Returns

A string representing variant forms in LaTeX format.

Definition at line 616 of file tex.py.

5.49.1.29 def lmf.src.output.tex.handle_caps (*text*)

```
Handle small caps.
Replace '°xxx' by '\textsc{xxx}' in translated examples.
```

Definition at line 262 of file tex.py.

5.49.1.30 def lmf.src.output.tex.handle_fi (*text*)

```
Replace 'fi:xxx' and '|fi{xxx}' by \\textit{xxx}.
```

Definition at line 217 of file tex.py.

5.49.1.31 def lmf.src.output.tex.handle_fn (*text, font*)

```
Replace 'fn:xxx' and '|fn{xxx}' by font[NATIONAL](xxx).
```

Definition at line 242 of file tex.py.

5.49.1.32 def lmf.src.output.tex.handle_font (*text*)

Functions to process LaTeX layout.

```
Replace '{xxx}' by '\ipa{xxx}' in 'un', 'xn', 'gn', 'dn', 'en'.
```

Definition at line 188 of file tex.py.

5.49.1.33 def lmf.src.output.tex.handle_fv (*text, font*)

```
Replace 'fv:xxx' and '|fv{xxx}' by font[VERNACULAR](xxx).
```

Definition at line 229 of file tex.py.

5.49.1.34 def lmf.src.output.tex.handle_pinyin (*text*)

```
Replace '@xxx' by '\\textcolor{gray}{xxx}' in 'lx', 'dv', 'xv' fields (already in API).
```

Definition at line 254 of file tex.py.

5.49.1.35 def lmf.src.output.tex.handle_quotes (*text*)

```
Hanlde quotation marks.
Replace each "xxx" by ``xxx".
```

Definition at line 272 of file tex.py.

5.49.1.36 def lmf.src.output.tex.handle_reserved (*text*)

```
Handle reserved characters $ & % # _ ^ except \ { }.
```

Definition at line 197 of file tex.py.

5.49.1.37 def lmf.src.output.tex.insert_references (*lexical_entry*)

Insert references to paradigms.

Parameters

lexical_entry	The targeted Lexical Entry LMF instance.

Returns

A string representing the references in LaTeX format.

Definition at line 25 of file tex.py.

5.49.1.38 def lmf.src.output.tex.tex_write (*object, filename, preamble* = None, *introduction* = None, *lmf2tex* = lmf_to_tex, *font* = None, *items* = lambda lexical_entry: lexical_entry.get_lexeme(), *sort_order* = None, *paradigms* = [], *tables* = [])

Write a LaTeX file.

Note that the lexicon must already be ordered at this point. Here, parameters 'items' and 'sort_order' are only used to define chapters.

Parameters

object	The LMF instance to convert into LaTeX output format.
filename	The name of the LaTeX file to write with full path, for instance 'user/output.tex'.
preamble	The name of the LaTeX file with full path containing the LaTeX header of the document, for instance 'user/config/japhug.tex'. Deafult value is None.
lmf2tex	A function giving the mapping from LMF representation information that must be written to LaTeX commands, in a defined order. Default value is 'lmf_to_tex' function defined in 'src/config/tex.py'. Please refer to it as an example.
font	A Python dictionary giving the vernacular, national, regional fonts to apply to a text in LaTeX format.
items	Lambda function giving the item to sort. Default value is 'lambda lexical_entry: lexical_↩ entry.get_lexeme()', which means that the items to sort are lexemes.
sort_order	Default value is 'None', which means that the LaTeX output is alphabetically ordered.
paradigms	A Python list of LaTeX filenames with full path containing the paradigms in LaTeX format. Default value is an empty list.

Definition at line 53 of file tex.py.

5.50 lmf.src.output.txt Namespace Reference

5.51 lmf.src.output.xls Namespace Reference

5.52 lmf.src.output.xml_ite Namespace Reference

5.53 lmf.src.output.xml_lift Namespace Reference

5.54 lmf.src.output.xml_lmf Namespace Reference

Functions

- def xml_lmf_write

  Write an XML LMF file.
- def build_sub_elements

  Create XML sub-elements to an existing XML element by parsing an LMF object instance.
- def add_link

  Functions to process XML/XHTML layout.
- def handle_reserved
- def handle_fv
- def handle_fn
- def handle_font
- def handle_pinyin
- def handle_caps
- def handle_tones

5.54.1 Function Documentation

5.54.1.1 def lmf.src.output.xml_lmf.add_link (*object, element*)

Functions to process XML/XHTML layout.

```
Insert an hyperlink <a href=xxx>xxx<a/> in XML.
```

Definition at line 66 of file xml_lmf.py.

5.54.1.2 def lmf.src.output.xml_lmf.build_sub_elements (*object, element*)

Create XML sub-elements to an existing XML element by parsing an LMF object instance.

Parameters

object	An LMF object instance.
element	XML element for which sub-elements have to be created according to LMF object attributes.

Definition at line 19 of file xml_lmf.py.

5.54.1.3 def lmf.src.output.xml_lmf.handle_caps (*element*)

```
Handle small caps.
Replace '°xxx' by '<span class="sc">xxx</span>'.
```

Definition at line 236 of file xml_lmf.py.

5.54.1.4 def lmf.src.output.xml_lmf.handle_fn (*element*)

```
Replace 'fn:xxx' and '|fn{xxx}' by '<span class="national">xxx</span>'.
```

Definition at line 131 of file xml_lmf.py.

5.54.1.5 def lmf.src.output.xml_lmf.handle_font (*element*)

```
Replace '{xxx}' by '<span class="ipa">xxx</span>'.
```

Definition at line 168 of file xml_lmf.py.

5.54.1.6 def lmf.src.output.xml_lmf.handle_fv (*element*)

```
Replace 'fv:xxx' and '|fv{xxx}' by '<span class="vernacular">xxx</span>'.
```

Definition at line 92 of file xml_lmf.py.

5.54.1.7 def lmf.src.output.xml_lmf.handle_pinyin (*element*)

```
Replace '@xxx' by '<span class="pinyin">xxx</span>'.
```

Definition at line 202 of file xml_lmf.py.

5.54.1.8 def lmf.src.output.xml_lmf.handle_reserved (*element*)

```
Handle reserved characters.
```

Definition at line 87 of file xml_lmf.py.

5.54.1.9 def lmf.src.output.xml_lmf.handle_tones (*element*)

```
Replace tones subscripts by '<sub>xxx</sub>'.
```

Definition at line 271 of file xml_lmf.py.

5.54.1.10 def lmf.src.output.xml_lmf.xml_lmf_write (*object, filename*)

Write an XML LMF file.

Parameters

object	The LMF instance to write as XML.
filename	The name of the XML LMF file to write with full path, for instance 'user/output.xml'.

Definition at line 7 of file xml_lmf.py.

5.55 lmf.src.output.xml_lp Namespace Reference

5.56 lmf.src.output.xml_olif Namespace Reference

5.57 lmf.src.output.xml_tb Namespace Reference

5.58 lmf.src.output.xml_tei Namespace Reference

5.59 lmf.src.resources Namespace Reference

Namespaces

- audio
- human_resource
- material
- picture
- resource
- speaker
- video

5.60 lmf.src.resources.audio Namespace Reference

Classes

- class Audio

  Audio is a Material subclass representing an audio recording.

5.61 lmf.src.resources.human_resource Namespace Reference

Classes

- class HumanResource

  HumanResource is a Resource subclass.

5.62 lmf.src.resources.material Namespace Reference

Classes

- class Material

  Material is a Resource subclass.

5.63 lmf.src.resources.picture Namespace Reference

Classes

- class Picture

  Picture is a Material subclass representing a picture.

5.64 lmf.src.resources.resource Namespace Reference

Classes

- class Resource

  Resource is an abstract class representing a material or a human resource.

5.65 lmf.src.resources.speaker Namespace Reference

Classes

- class Speaker

  Speaker is a HumanResource subclass.

5.66 lmf.src.resources.video Namespace Reference

Classes

- class Video

  Video is a Material subclass representing a video.

5.67 lmf.src.utils Namespace Reference

Namespaces

- attr
- error_handling
- io
- ipa2sampa
- log
- xml_format

5.68 lmf.src.utils.attr Namespace Reference

Functions

- def check_attr_type

  Check that attribute value is of specified type.
- def check_attr_range

  Check that attribute value is in specified range.
- def check_date_format

  Verify that date format is composed as follows: YYYY-MM-DD (ISO 8601).
- def check_time_format

  Verify that time format is composed as follows: THH:MM:SS,MSMS (ISO 8601: 'T' for Time).
- def check_duration_format

  Verify that duration format is composed as follows: PTxxHxxMxxS (ISO 8601: 'P' for Period).

5.68.1 Function Documentation

5.68.1.1 def lmf.src.utils.attr.check_attr_range (*value, range, msg, mapping* = None)

Check that attribute value is in specified range.

Parameters

value	The attribute value to check.
range	A Python set giving the range of allowed values.
msg	The message to display if value is out-of-range.
mapping	A Python dictionary giving mapping between values (i.e. from MDF to LMF)

Returns

The value to set, or None if out-of-range.

Definition at line 20 of file attr.py.

5.68.1.2 def lmf.src.utils.attr.check_attr_type (*val, typ, msg*)

Check that attribute value is of specified type.

Parameters

val	The attribute value to check.
typ	The allowed Python type(s): simple, or Python set or list.
msg	The message to display if value is not of correct type.

Definition at line 6 of file attr.py.

5.68.1.3 def lmf.src.utils.attr.check_date_format (*date*)

Verify that date format is composed as follows: YYYY-MM-DD (ISO 8601).

If not, display a Warning message.

Parameters

date	Date to check.

Definition at line 45 of file attr.py.

5.68.1.4 def lmf.src.utils.attr.check_duration_format (*duration*)

Verify that duration format is composed as follows: PTxxHxxMxxS (ISO 8601: 'P' for Period).

If not, display a Warning message.

Parameters

duration	Duration to check.

Definition at line 63 of file attr.py.

5.68.1.5 def lmf.src.utils.attr.check_time_format (*time*)

Verify that time format is composed as follows: THH:MM:SS,MSMS (ISO 8601: 'T' for Time).

If not, display a Warning message.

Parameters

time	Time to check.

Definition at line 54 of file attr.py.

5.69 lmf.src.utils.error_handling Namespace Reference

Classes

- class Error

  Base class for exceptions in this library.
- class InputError

  Exception raised for errors in the input.
- class OutputError

  Exception raised for errors in the output.
- class Warning

  Base class for warnings in this library.

5.70 lmf.src.utils.io Namespace Reference

Functions

- def open_file

  Open file in specified mode (automatically decode file in unicode).
- def open_read

  Open file in read mode (automatically decode file in unicode).
- def open_write

  Open file in write mode (automatically decode file in unicode).

Variables

- string EOL = '\n'
- string ENCODING = 'utf-8'

5.70.1 Function Documentation

5.70.1.1 def lmf.src.utils.io.open_file (*filename, mode, encoding* = ENCODING)

Open file in specified mode (automatically decode file in unicode).

Parameters

filename	Full path to file to open.
mode	Read or write mode.
encoding	Encoding mode. Default value is 'utf-8'.

Returns

File handler.

Definition at line 17 of file io.py.

5.70.1.2 def lmf.src.utils.io.open_read (*filename, encoding* = None)

Open file in read mode (automatically decode file in unicode).

Parameters

filename	Full path to file to open.
encoding	Encoding mode. Default value is None.

Returns

File handler.

Definition at line 33 of file io.py.

5.70.1.3 def lmf.src.utils.io.open_write (*filename, encoding* = None)

Open file in write mode (automatically decode file in unicode).

Parameters

filename	Full path to file to open.
encoding	Encoding mode. Default value is None.

Returns

File handler.

Definition at line 44 of file io.py.

5.70.2 Variable Documentation

5.70.2.1 string lmf.src.utils.io.ENCODING = 'utf-8'

Definition at line 15 of file io.py.

5.70.2.2 string lmf.src.utils.io.EOL = '\n'

Definition at line 9 of file io.py.

5.71 lmf.src.utils.ipa2sampa Namespace Reference

Namespaces

- ipa2sampa

5.72 lmf.src.utils.ipa2sampa.ipa2sampa Namespace Reference

Functions

- def uni2sampa

Variables

- tuple data = codecs.open('./src/utils/ipa2sampa/sampa.csv', 'r', 'utf-8')
- list sota = []
- tuple ta = eval('"""'+ta+'"""')
- tuple seq = line.strip()

5.72.1 Function Documentation

5.72.1.1 def lmf.src.utils.ipa2sampa.ipa2sampa.uni2sampa (*sequence*)

```
Convert sequence in unicode-ipa to ascii-sampa.

Notes
-----
Forked from LingPy's version for ipa2sampa, which is based on code
taken from Peter Kleiweg
(http://www.let.rug.nl/~kleiweg/L04/devel/python/xsampa.html).
```

Definition at line 30 of file ipa2sampa.py.

5.72.2 Variable Documentation

5.72.2.1 tuple lmf.src.utils.ipa2sampa.ipa2sampa.data = codecs.open('./src/utils/ipa2sampa/sampa.csv', 'r', 'utf-8')

Definition at line 13 of file ipa2sampa.py.

5.72.2.2 tuple lmf.src.utils.ipa2sampa.ipa2sampa.seq = line.strip()

Definition at line 59 of file ipa2sampa.py.

5.72.2.3 tuple lmf.src.utils.ipa2sampa.ipa2sampa.sota = []

Definition at line 16 of file ipa2sampa.py.

5.72.2.4 tuple lmf.src.utils.ipa2sampa.ipa2sampa.ta = eval('"""'+ta+'"""')

Definition at line 24 of file ipa2sampa.py.

5.73 lmf.src.utils.log Namespace Reference

Functions

- def log

  Write message into log file if any, or to standard output if verbose mode is on.

5.73.1 Function Documentation

5.73.1.1 def lmf.src.utils.log.log (*msg*, *options* = None)

Write message into log file if any, or to standard output if verbose mode is on.

Parameters

msg	String to log.
options	User options.

Definition at line 6 of file log.py.

5.74 lmf.src.utils.xml_format Namespace Reference

Functions

- def prettify

  Return a pretty-printed XML string for the given XML element.
- def write_result

  Write an XML element into a pretty XML output file.
- def parse_xml

  Parse an XML file.

5.74.1 Function Documentation

5.74.1.1 def lmf.src.utils.xml_format.parse_xml (*filename*)

Parse an XML file.

Parameters

filename	The name of the XML file to parse with full path, for instance 'user/input.xml'.

Returns

The root XML element.

Definition at line 32 of file xml_format.py.

5.74.1.2 def lmf.src.utils.xml_format.prettify (*element*, *encoding* = ENCODING)

Return a pretty-printed XML string for the given XML element.

Parameters

element	An XML element.
encoding	Encoding mode. Default value is 'utf-8'.

Returns

A Python string containing the printed version of the XML element.

Definition at line 10 of file xml_format.py.

5.74.1.3 def lmf.src.utils.xml_format.write_result (*element, filename, encoding* = ENCODING)

Write an XML element into a pretty XML output file.

Parameters

element	An XML element.
filename	The name of the XML file to write with full path, for instance 'user/output.xml'.
encoding	Encoding mode. Default value is 'utf-8'.

Definition at line 21 of file xml_format.py.

5.75 lmf.src.wrapper Namespace Reference

Functions

- def wrapper_rw

  Wrapper function that calls another function, restoring normal behavior on error.
- def read_mdf
- def read_xml_lmf
- def read_sort_order
- def read_config
- def write_mdf
- def write_xml_lmf
- def write_tex
- def write_doc

Variables

- lexical_resource = None

  Module variable.

5.75.1 Function Documentation

5.75.1.1 def lmf.src.wrapper.read_config (*args, kwds*)

Definition at line 113 of file wrapper.py.

5.75.1.2 def lmf.src.wrapper.read_mdf (*args, kwds*)

Definition at line 69 of file wrapper.py.

5.75.1.3 def lmf.src.wrapper.read_sort_order (*args, kwds*)

Definition at line 108 of file wrapper.py.

5.75.1.4 def lmf.src.wrapper.read_xml_lmf (*args, kwds*)

Definition at line 92 of file wrapper.py.

5.75.1.5 def lmf.src.wrapper.wrapper_rw (*func, args, kwds*)

Wrapper function that calls another function, restoring normal behavior on error.

Parameters

func	Callable object.
args	Arguments passed to 'func' as its first argument.
kwds	Other arguments passed to 'func'.

Definition at line 27 of file wrapper.py.

5.75.1.6 def lmf.src.wrapper.write_doc (*args, kwds*)

Definition at line 142 of file wrapper.py.

5.75.1.7 def lmf.src.wrapper.write_mdf (*args, kwds*)

Definition at line 118 of file wrapper.py.

5.75.1.8 def lmf.src.wrapper.write_tex (*args, kwds*)

Definition at line 133 of file wrapper.py.

5.75.1.9 def lmf.src.wrapper.write_xml_lmf (*args, kwds*)

Definition at line 124 of file wrapper.py.

5.75.2 Variable Documentation

5.75.2.1 lmf.src.wrapper.lexical_resource = None

Module variable.

Definition at line 25 of file wrapper.py.

5.76 tables Namespace Reference

Variables

- tuple parser = OptionParser()
- tuple options = parser.parse_args()
- tuple in_file = open(options.input, "r", encoding='utf-8')
- tuple out_eng = open(options.output_eng, "w", encoding='utf-8')

- tuple out_fra = open(options.output_fra, "w", encoding='utf-8')
- string EOL = '\n'
- string title_eng = """Words for which no close equivalent could be found"""
- string introduction_eng = """The list that follows groups words for which no close equivalents could be found. These negative pieces of information contain hints about the consultants' Na vocabulary and its 'soft shoulders'."""
- string title_fra = """Mots dont aucun équivalent n'a été trouvé"""
- string introduction_fra = """Cette liste regroupe les mots dont aucun équivalent n'a été trouvé. Même s'il ne s'agit que d'informations négatives, elles éclairent les limites du vocabulaire na des consultants."""
- string pattern = r"^\\(\w{2,3}) ?(.*)$"
- string lx = ""
- string ge = ""
- string gn = ""
- string gf = ""
- tuple result = re.search(pattern, line)

5.76.1 Variable Documentation

5.76.1.1 string tables.EOL = '\n'

Definition at line 26 of file tables.py.

5.76.1.2 string tables.ge = ""

Definition at line 52 of file tables.py.

5.76.1.3 string tables.gf = ""

Definition at line 54 of file tables.py.

5.76.1.4 string tables.gn = ""

Definition at line 53 of file tables.py.

5.76.1.5 tuple tables.in_file = open(options.input, "r", encoding='utf-8')

Definition at line 14 of file tables.py.

5.76.1.6 string tables.introduction_eng = """The list that follows groups words for which no close equivalents could be found. These negative pieces of information contain hints about the consultants' Na vocabulary and its 'soft shoulders'."""

Definition at line 32 of file tables.py.

5.76.1.7 string tables.introduction_fra = """Cette liste regroupe les mots dont aucun équivalent n'a été trouvé. Même s'il ne s'agit que d'informations négatives, elles éclairent les limites du vocabulaire na des consultants."""

Definition at line 34 of file tables.py.

5.76.1.8 string tables.lx = ""

Definition at line 51 of file tables.py.

5.76.1.9 tuple tables.options = parser.parse_args()

Definition at line 10 of file tables.py.

5.76.1.10 tuple tables.out_eng = open(options.output_eng, "w", encoding='utf-8')

Definition at line 15 of file tables.py.

5.76.1.11 tuple tables.out_fra = open(options.output_fra, "w", encoding='utf-8')

Definition at line 16 of file tables.py.

5.76.1.12 tuple tables.parser = OptionParser()

Definition at line 6 of file tables.py.

5.76.1.13 string tables.pattern = r"^\\(\w{2,3}) ?(.*)$"

Definition at line 50 of file tables.py.

5.76.1.14 tuple tables.result = re.search(pattern, line)

Definition at line 56 of file tables.py.

5.76.1.15 string tables.title_eng = """Words for which no close equivalent could be found"""

Definition at line 31 of file tables.py.

5.76.1.16 string tables.title_fra = """Mots dont aucun équivalent n'a été trouvé"""

Definition at line 33 of file tables.py.

5.77 uid Namespace Reference

Variables

- tuple parser = OptionParser()
- tuple options = parser.parse_args()
- tuple in_file = open(options.input, "r", encoding='utf-8')
- tuple out_file = open(options.output, "w", encoding='utf-8')
- string EOL = '\n'
- string pattern = r"^\\(\w{2,3}) ?(.*)$"
- string lx = ""
- string mkr = "lx"
- list sf = []
- string hm = ""
- tuple result = re.search(pattern, line)
- tuple uid = uni2sampa(lx)

5.77.1 Variable Documentation

5.77.1.1 string uid.EOL = '\n'

Definition at line 23 of file uid.py.

5.77.1.2 string uid.hm = ""

Definition at line 38 of file uid.py.

5.77.1.3 tuple uid.in_file = open(options.input, "r", encoding='utf-8')

Definition at line 13 of file uid.py.

5.77.1.4 string uid.lx = ""

Definition at line 35 of file uid.py.

5.77.1.5 string uid.mkr = "lx"

Definition at line 36 of file uid.py.

5.77.1.6 tuple uid.options = parser.parse_args()

Definition at line 9 of file uid.py.

5.77.1.7 tuple uid.out_file = open(options.output, "w", encoding='utf-8')

Definition at line 14 of file uid.py.

5.77.1.8 tuple uid.parser = OptionParser()

Definition at line 6 of file uid.py.

5.77.1.9 string uid.pattern = r"^\\(\w{2,3}) ?(.*)$"

Definition at line 34 of file uid.py.

5.77.1.10 tuple uid.result = re.search(pattern, line)

Definition at line 40 of file uid.py.

5.77.1.11 list uid.sf = [ ]

Definition at line 37 of file uid.py.

5.77.1.12 tuple uid.uid = uni2sampa(lx)

Definition at line 52 of file uid.py.

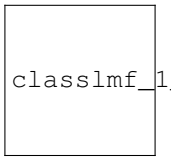
Chapter 6

Class Documentation

6.1 lmf.src.resources.audio.Audio Class Reference

Audio is a Material subclass representing an audio recording.

Inheritance diagram for lmf.src.resources.audio.Audio:



Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_mediaType
  Set media type.
- def get_mediaType
  Get media type.
- def set_fileName
  Set file name.
- def get_fileName
  Get file name.
- def set_author
  Set author of the material resource.
- def get_author
  Get author of the material resource.
- def set_quality
  Set audio recording quality.
- def get_quality
  Get audio recording quality.
- def set_sound
  Set sound.
- def get_sound

Get sound.

- def set_transcription

  Set transcription of the audio recording.
- def get_transcription

  Get transcription of the audio recording.
- def set_startPosition

  Set start position.
- def get_startPosition

  Get start position.
- def set_durationOfEffectiveSpeech

  Set duration of effective speech.
- def get_durationOfEffectiveSpeech

  Get duration of effective speech.
- def set_externalReference

  Set external reference.
- def get_externalReference

  Get external reference.
- def set_audioFileFormat

  Set audio file format.
- def get_audioFileFormat

  Get audio file formay.

Public Attributes

- quality
- sound
- startPosition
- durationOfEffectiveSpeech
- externalReference
- audioFileFormat
- transcription
- mediaType
- fileName
- author

6.1.1 Detailed Description

Audio is a Material subclass representing an audio recording.

Definition at line 11 of file audio.py.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 def lmf.src.resources.audio.Audio.__init__ (*self*)

Constructor.

Audio instances are owned by FormRepresentation.

Returns

An Audio instance.

Definition at line 14 of file audio.py.

6.1.2.2 def lmf.src.resources.audio.Audio.__del__ (*self*)

Destructor.

Definition at line 29 of file audio.py.

6.1.3 Member Function Documentation

6.1.3.1 def lmf.src.resources.audio.Audio.get_audioFileFormat (*self*)

Get audio file formay.

Returns

Audio attribute 'audioFileFormat'.

Definition at line 199 of file audio.py.

6.1.3.2 def lmf.src.resources.audio.Audio.get_author (*self*)

Get author of the material resource.

Returns

Audio attribute 'author'.

Definition at line 76 of file audio.py.

6.1.3.3 def lmf.src.resources.audio.Audio.get_durationOfEffectiveSpeech (*self*)

Get duration of effective speech.

Returns

Audio attribute 'durationOfEffectiveSpeech'.

Definition at line 167 of file audio.py.

6.1.3.4 def lmf.src.resources.audio.Audio.get_externalReference (*self*)

Get external reference.

Returns

Audio attribute 'externalReference'.

Definition at line 183 of file audio.py.

6.1.3.5 def lmf.src.resources.audio.Audio.get_fileName (*self*)

Get file name.

Returns

Audio attribute 'fileName'.

Definition at line 60 of file audio.py.

6.1.3.6 def lmf.src.resources.audio.Audio.get_mediaType (*self*)

Get media type.

Returns

Audio attribute 'mediaType'.

Definition at line 44 of file audio.py.

6.1.3.7 def lmf.src.resources.audio.Audio.get_quality (*self*)

Get audio recording quality.

Returns

Audio attribute 'quality'.

Definition at line 92 of file audio.py.

6.1.3.8 def lmf.src.resources.audio.Audio.get_sound (*self*)

Get sound.

Returns

Audio attribute 'sound'.

Definition at line 108 of file audio.py.

6.1.3.9 def lmf.src.resources.audio.Audio.get_startPosition (*self*)

Get start position.

Returns

Audio attribute 'startPosition'.

Definition at line 144 of file audio.py.

6.1.3.10 def lmf.src.resources.audio.Audio.get_transcription (*self*)

Get transcription of the audio recording.

Returns

Audio attribute 'transcription'.

Definition at line 124 of file audio.py.

6.1.3.11 def lmf.src.resources.audio.Audio.set_audioFileFormat (*self, audio_file_format*)

Set audio file format.

Parameters

audio_file_↩ format	Audio file format to set.

Returns

Audio instance.

Definition at line 189 of file audio.py.

6.1.3.12 def lmf.src.resources.audio.Audio.set_author (*self, author*)

Set author of the material resource.

Parameters

author	Author to set.

Returns

Audio instance.

Definition at line 66 of file audio.py.

6.1.3.13 def lmf.src.resources.audio.Audio.set_durationOfEffectiveSpeech (*self, duration*)

Set duration of effective speech.

Parameters

duration	Duration of effective speech to set.

Returns

Audio instance.

Definition at line 150 of file audio.py.

6.1.3.14 def lmf.src.resources.audio.Audio.set_externalReference (*self, external_reference*)

Set external reference.

Parameters

external_↩ reference	External reference to set.

Returns

Audio instance.

Definition at line 173 of file audio.py.

6.1.3.15 def lmf.src.resources.audio.Audio.set_fileName (*self, file_name*)

Set file name.

Parameters

file_name	Name to set.

Returns

Audio instance.

Definition at line 50 of file audio.py.

6.1.3.16 def lmf.src.resources.audio.Audio.set_mediaType (*self, media_type*)

Set media type.

Parameters

media_type	Type to set.

Returns

Audio instance.

Definition at line 34 of file audio.py.

6.1.3.17 def lmf.src.resources.audio.Audio.set_quality (*self, quality*)

Set audio recording quality.

Parameters

quality	Quality to set.

Returns

Audio instance.

Definition at line 82 of file audio.py.

6.1.3.18 def lmf.src.resources.audio.Audio.set_sound (*self, sound*)

Set sound.

Parameters

sound	Sound to set.

Returns

Audio instance.

Definition at line 98 of file audio.py.

6.1.3.19 def lmf.src.resources.audio.Audio.set_startPosition (*self, start_position*)

Set start position.

Parameters

start_position	Start position to set.

Returns

Audio instance.

Definition at line 130 of file audio.py.

6.1.3.20 def lmf.src.resources.audio.Audio.set_transcription (*self, transcription*)

Set transcription of the audio recording.

Parameters

Transcription	to set.

Returns

Audio instance.

Definition at line 114 of file audio.py.

6.1.4 Member Data Documentation

6.1.4.1 lmf.src.resources.audio.Audio.audioFileFormat

Definition at line 26 of file audio.py.

6.1.4.2 lmf.src.resources.audio.Audio.author

Definition at line 73 of file audio.py.

6.1.4.3 lmf.src.resources.audio.Audio.durationOfEffectiveSpeech

Definition at line 24 of file audio.py.

6.1.4.4 lmf.src.resources.audio.Audio.externalReference

Definition at line 25 of file audio.py.

6.1.4.5 lmf.src.resources.audio.Audio.fileName

Definition at line 57 of file audio.py.

6.1.4.6 lmf.src.resources.audio.Audio.mediaType

Definition at line 41 of file audio.py.

6.1.4.7 lmf.src.resources.audio.Audio.quality

Definition at line 21 of file audio.py.

6.1.4.8 lmf.src.resources.audio.Audio.sound

Definition at line 22 of file audio.py.

6.1.4.9 lmf.src.resources.audio.Audio.startPosition

Definition at line 23 of file audio.py.

6.1.4.10 lmf.src.resources.audio.Audio.transcription

Definition at line 27 of file audio.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/audio.py

6.2 lmf.src.morphology.component.Component Class Reference

Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_lexical_entry
  Set pointer to the component lexical entry instance.
- def get_lexical_entry
  Get pointed lexical entry.
- def get_lexeme
  Get component LexicalEntry lexeme.

Public Attributes

- position
- targets

6.2.1 Detailed Description

Definition at line 6 of file component.py.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 def lmf.src.morphology.component.Component.__init__ (*self*, *position* = `None`, *lexeme* = `None`)

Constructor.

Component instances are owned by ListOfComponents.

Parameters

position	The position of the component in the multiword expression.
targets	Related lexeme.

Returns

A Component instance.

Definition at line 7 of file component.py.

6.2.2.2 def lmf.src.morphology.component.Component.__del__ (*self*)

Destructor.

Definition at line 21 of file component.py.

6.2.3 Member Function Documentation

6.2.3.1 def lmf.src.morphology.component.Component.get_lexeme (*self*)

Get component LexicalEntry lexeme.

Returns

Component attribute 'targets'.

Definition at line 41 of file component.py.

6.2.3.2 def lmf.src.morphology.component.Component.get_lexical_entry (*self*)

Get pointed lexical entry.

Returns

Component private attribute '__lexical_entry'.

Definition at line 35 of file component.py.

6.2.3.3 def lmf.src.morphology.component.Component.set_lexical_entry (*self, lexical_entry*)

Set pointer to the component lexical entry instance.

This function can only be called once the full dictionary has been parsed.

Parameters

lexical_entry	The component LexicalEntry.

Returns

Component instance.

Definition at line 27 of file component.py.

6.2.4 Member Data Documentation

6.2.4.1 lmf.src.morphology.component.Component.position

Definition at line 14 of file component.py.

6.2.4.2 lmf.src.morphology.component.Component.targets

Definition at line 16 of file component.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/component.py

6.3 lmf.src.mrd.context.Context Class Reference

"Context is a class representing a text string that provides authentic context for the use of the word form managed by the Lemma. This class is to be distinguished from Sense Example." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_type

  Set context type.
- def get_type

  Get context type.
- def create_text_representation

  Create a text representation.
- def add_text_representation

  Add a text representation to the context.
- def get_text_representations

  Get all text representations maintained by the context.
- def get_last_text_representation

  Get the previously registered TextRepresentation instance.
- def find_written_forms

  Find written forms.
- def get_comments

  Get comments.
- def set_written_form

  Set text representation written form, language and script.
- def set_comment

  Set text representation comment.
- def get_speakerID

  Get related speaker identifier.
- def get_speaker

  Get speaker.

Public Attributes

- language
- type
- text_representation

  TextRepresentation instances are owned by Context There is zero to many TextRepresentation instances per Context.
- targets

6.3.1 Detailed Description

"Context is a class representing a text string that provides authentic context for the use of the word form managed by the Lemma. This class is to be distinguished from Sense Example." (LMF)

Definition at line 10 of file context.py.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 def lmf.src.mrd.context.Context.__init__ (*self, speakerID* = None)

Constructor.

Context instances are owned by Sense.

Parameters

speakerID	Related speaker identifier. If not provided, default value is None.

Returns

A Context instance.

Definition at line 13 of file context.py.

6.3.2.2 def lmf.src.mrd.context.Context.__del__ (*self*)

Destructor.

Release TextRepresentation instances.

Definition at line 30 of file context.py.

6.3.3 Member Function Documentation

6.3.3.1 def lmf.src.mrd.context.Context.add_text_representation (*self, text_representation*)

Add a text representation to the context.

Parameters

text_↩ representation	The TextRepresentation instance to add to the context.

Returns

Context instance.

Definition at line 65 of file context.py.

6.3.3.2 def lmf.src.mrd.context.Context.create_text_representation (*self*)

Create a text representation.

Returns

TextRepresentation instance.

Definition at line 59 of file context.py.

6.3.3.3 def lmf.src.mrd.context.Context.find_written_forms (*self*, *language* = None, *script_name* = None)

Find written forms.

This attribute is owned by TextRepresentation.

Parameters

language	If given, the language to consider to retrieve the written form.
script_name	If given, the script to consider to retrieve the written form.

Returns

A Python list of found TextRepresentation attributes 'writtenForm'.

Definition at line 86 of file context.py.

6.3.3.4 def lmf.src.mrd.context.Context.get_comments (*self*)

Get comments.

This attribute is owned by TextRepresentation.

Returns

A Python list of found TextRepresentation attributes 'comment'.

Definition at line 100 of file context.py.

6.3.3.5 def lmf.src.mrd.context.Context.get_last_text_representation (*self*)

Get the previously registered TextRepresentation instance.

Returns

The last element of Context attribute 'text_representation'.

Definition at line 79 of file context.py.

6.3.3.6 def lmf.src.mrd.context.Context.get_speaker (*self*)

Get speaker.

Returns

Context private attribute '__speaker'.

Definition at line 150 of file context.py.

6.3.3.7 def lmf.src.mrd.context.Context.get_speakerID (*self*)

Get related speaker identifier.

Returns

Context attribute 'targets'.

Definition at line 144 of file context.py.

6.3.3.8 def lmf.src.mrd.context.Context.get_text_representations (*self*)

Get all text representations maintained by the context.

Returns

A Python list of text representations.

Definition at line 73 of file context.py.

6.3.3.9 def lmf.src.mrd.context.Context.get_type (*self*)

Get context type.

Returns

Context attribute 'type'.

Definition at line 53 of file context.py.

6.3.3.10 def lmf.src.mrd.context.Context.set_comment (*self, comment*)

Set text representation comment.

Attribute 'comment' is owned by TextRepresentation.

Parameters

comment	The comment to set.

Returns

Context instance.

Definition at line 128 of file context.py.

6.3.3.11 def lmf.src.mrd.context.Context.set_type (*self, type*)

Set context type.

Parameters

type	Type of text representations, in range 'type_example_range' defined in 'common/range.py'.

Returns

Context instance.

Definition at line 40 of file context.py.

6.3.3.12 def lmf.src.mrd.context.Context.set_written_form (*self, written_form, language* = None, *script_name* = None)

Set text representation written form, language and script.

Attributes 'writtenForm', 'language' and 'scriptName' are owned by TextRepresentation.

Parameters

written_form	The written form to set.
language	Language of the written form.
script_name	The name of the script used to write the form, e.g. devanagari.

Returns

Context instance.

Definition at line 111 of file context.py.

6.3.4 Member Data Documentation

6.3.4.1 lmf.src.mrd.context.Context.language

Definition at line 19 of file context.py.

6.3.4.2 lmf.src.mrd.context.Context.targets

Definition at line 25 of file context.py.

6.3.4.3 lmf.src.mrd.context.Context.text_representation

TextRepresentation instances are owned by Context There is zero to many TextRepresentation instances per Context.

Definition at line 23 of file context.py.

6.3.4.4 lmf.src.mrd.context.Context.type

Definition at line 20 of file context.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/context.py

6.4 lmf.src.core.definition.Definition Class Reference

"Definition is a class representing a narrative description of a sense. It is provided to help human users understand the meaning of a lexical entry. A Sense instance can have zero to many definitions. Each Definition instance may

be associated with zero to many Text Representation instances in order to manage the text defintion in more than one language or script. In addition, the narrative description can be expressed in a different language or script than the one in the Lexical Entry instance." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_language

  Set language used for definition and gloss.
- def get_language

  Get language used for definition and gloss.
- def set_definition

  Set definition.
- def get_definition

  Get definition.
- def set_gloss

  Set gloss.
- def get_gloss

  Get gloss.
- def create_statement

  Create a Statement instance.
- def add_statement

  Add a Statement instance to this Definition instance.
- def get_statements

  Get all Statement instances maintained by this Definition instance.
- def get_first_statement

  Get the previously registered statement.
- def set_note

  Set note, note type and language.
- def find_notes

  Find notes.
- def set_usage_note

  Set usage note and language.
- def find_usage_notes

  Find usage notes.
- def set_encyclopedic_information

  Set encyclopedic information and language.
- def find_encyclopedic_informations

  Find encyclopedic informations.
- def set_restriction

  Set restriction and language.
- def find_restrictions

  Find restrictions.
- def set_borrowed_word

  Set source language (in English).
- def get_borrowed_word

  Get source language (in English).
- def set_written_form

Set loan word.

- def get_written_form

  Get loan word.

- def set_etymology

  Set etymology.

- def get_etymology

  Get etymology.

- def set_etymology_comment

  Set etymology comment and language.

- def get_etymology_comment

  Get etymology comment.

- def get_term_source_language

  Get language used for the etymology comment.

- def set_etymology_gloss

  Set etymology gloss.

- def get_etymology_gloss

  Get etymology gloss.

- def set_etymology_source

  Set etymology source.

- def get_etymology_source

  Get etymology source.

- def set_scientific_name

  Set scientific name.

- def get_scientific_name

  Get scientific name.

Public Attributes

- language
- definition
- gloss
- literally
- text_representation

  TextRepresentation instances are owned by Definition There is zero to many TextRepresentation instances per Definition.

- statement

  Statement instances are owned by Definition There is zero to many Statement instances per Definition.

6.4.1 Detailed Description

"Definition is a class representing a narrative description of a sense. It is provided to help human users understand the meaning of a lexical entry. A Sense instance can have zero to many definitions. Each Definition instance may be associated with zero to many Text Representation instances in order to manage the text defintion in more than one language or script. In addition, the narrative description can be expressed in a different language or script than the one in the Lexical Entry instance." (LMF)

Definition at line 9 of file definition.py.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 def lmf.src.core.definition.Definition.__init__ (*self*)

Constructor.

Definition instances are owned by Sense.

Returns

A Definition instance.

Definition at line 12 of file definition.py.

6.4.2.2 def lmf.src.core.definition.Definition.__del__ (*self*)

Destructor.

Release TextRepresentation and Statement instances.

Definition at line 28 of file definition.py.

6.4.3 Member Function Documentation

6.4.3.1 def lmf.src.core.definition.Definition.add_statement (*self, statement*)

Add a Statement instance to this Definition instance.

Parameters

statemement	The Statement instance to add to the Definition instance.

Returns

Definition instance.

Definition at line 104 of file definition.py.

6.4.3.2 def lmf.src.core.definition.Definition.create_statement (*self*)

Create a Statement instance.

Returns

Statement instance.

Definition at line 98 of file definition.py.

6.4.3.3 def lmf.src.core.definition.Definition.find_encyclopedic_informations (*self, language*)

Find encyclopedic informations.

This attribute is owned by Statement.

Parameters

language	The language to consider to retrieve the encyclopedic information.

Returns

A Python list of found Statement attributes 'encyclopedicInformation'.

Definition at line 241 of file definition.py.

6.4.3.4 def lmf.src.core.definition.Definition.find_notes (*self, type*)

Find notes.

This attribute is owned by Statement.

Parameters

type	The type to consider to retrieve the note.

Returns

A Python list of found Statement attributes 'note'.

Definition at line 161 of file definition.py.

6.4.3.5 def lmf.src.core.definition.Definition.find_restrictions (*self, language*)

Find restrictions.

This attribute is owned by Statement.

Parameters

language	The language to consider to retrieve the restriction.

Returns

A Python list of found Statement attributes 'restriction'.

Definition at line 281 of file definition.py.

6.4.3.6 def lmf.src.core.definition.Definition.find_usage_notes (*self, language*)

Find usage notes.

This attribute is owned by Statement.

Parameters

language	The language to consider to retrieve the usage note.

Returns

A Python list of found Statement attributes 'usageNote'.

Definition at line 201 of file definition.py.

6.4.3.7 def lmf.src.core.definition.Definition.get_borrowed_word (*self*)

Get source language (in English).

This attribute is owned by the first Statement.

Returns

Statement attribute 'borrowedWord'.

Definition at line 308 of file definition.py.

6.4.3.8 def lmf.src.core.definition.Definition.get_definition (*self, language* = None)

Get definition.

Parameters

language	If this argument is given, get definition only if written in this language.

Returns

The filtered Definition attribute 'definition'.

Definition at line 67 of file definition.py.

6.4.3.9 def lmf.src.core.definition.Definition.get_etymology (*self*)

Get etymology.

This attribute is owned by the first Statement.

Returns

Statement attribute 'etymology'.

Definition at line 360 of file definition.py.

6.4.3.10 def lmf.src.core.definition.Definition.get_etymology_comment (*self, term_source_language* = None)

Get etymology comment.

This attribute is owned by the first Statement.

Parameters

term_source_↩ language	The language of the etymology comment to retrieve.

Returns

Statement attribute 'etymologyComment'.

Definition at line 387 of file definition.py.

6.4.3.11 def lmf.src.core.definition.Definition.get_etymology_gloss (*self*)

Get etymology gloss.

This attribute is owned by the first Statement.

Returns

Statement attribute 'etymologyGloss'.

Definition at line 425 of file definition.py.

6.4.3.12 def lmf.src.core.definition.Definition.get_etymology_source (*self*)

Get etymology source.

This attribute is owned by the first Statement.

Returns

Statement attribute 'etymologySource'.

Definition at line 451 of file definition.py.

6.4.3.13 def lmf.src.core.definition.Definition.get_first_statement (*self*)

Get the previously registered statement.

Returns

The last element of Definition attribute 'statement'.

Definition at line 118 of file definition.py.

6.4.3.14 def lmf.src.core.definition.Definition.get_gloss (*self, language* = `None`)

Get gloss.

Parameters

language	If this argument is given, get gloss only if written in this language.

Returns

The filtered Definition attribute 'gloss'.

Definition at line 88 of file definition.py.

6.4.3.15 def lmf.src.core.definition.Definition.get_language (*self*)

Get language used for definition and gloss.

Returns

Definition attribute 'language'.

Definition at line 50 of file definition.py.

6.4.3.16 def lmf.src.core.definition.Definition.get_scientific_name (*self*)

Get scientific name.

This attribute is owned by the first Statement.

Returns

Statement attribute 'scientificName'.

Definition at line 477 of file definition.py.

6.4.3.17 def lmf.src.core.definition.Definition.get_statements (*self*)

Get all Statement instances maintained by this Definition instance.

Returns

A Python list of Statement instances.

Definition at line 112 of file definition.py.

6.4.3.18 def lmf.src.core.definition.Definition.get_term_source_language (*self*)

Get language used for the etymology comment.

This attribute is owned by the first Statement.

Returns

Statement attribute 'termSourceLanguage'.

Definition at line 399 of file definition.py.

6.4.3.19 def lmf.src.core.definition.Definition.get_written_form (*self*)

Get loan word.

This attribute is owned by the first Statement.

Returns

Statement attribute 'writtenForm'.

Definition at line 334 of file definition.py.

6.4.3.20 def lmf.src.core.definition.Definition.set_borrowed_word (*self, borrowed_word*)

Set source language (in English).

Attribute 'borrowedWord' is owned by the first Statement.

Parameters

borrowed_word	Source language.

Returns

Definition instance.

Definition at line 293 of file definition.py.

6.4.3.21 def lmf.src.core.definition.Definition.set_definition (*self, definition, language* = `None`)

Set definition.

Parameters

definition	Definition.
language	Language used for the definition.

Returns

Definition instance.

Definition at line 56 of file definition.py.

6.4.3.22 def lmf.src.core.definition.Definition.set_encyclopedic_information (*self, encyclopedic_information, language =* `None`)

Set encyclopedic information and language.

These attributes are owned by Statement.

Parameters

encyclopedic_information	Encyclopedic information to set.
language	Language used for the encyclopedic information.

Returns

Definition instance.

Definition at line 213 of file definition.py.

6.4.3.23 def lmf.src.core.definition.Definition.set_etymology (*self, etymology*)

Set etymology.

Attribute 'etymology' is owned by the first Statement.

Parameters

etymology	Etymology.

Returns

Definition instance.

Definition at line 345 of file definition.py.

6.4.3.24 def lmf.src.core.definition.Definition.set_etymology_comment (*self, etymology_comment, term_source_language =* `None`)

Set etymology comment and language.

Attributes 'etymologyComment' and 'termSourceLanguage' are owned by the first Statement.

Parameters

etymology_comment	Etymology comment.

term_source_↩ language	Language of the comment.

Returns

Definition instance.

Definition at line 371 of file definition.py.

6.4.3.25 def lmf.src.core.definition.Definition.set_etymology_gloss (*self, etymology_gloss*)

Set etymology gloss.

Attribute 'etymologyGloss' is owned by the first Statement.

Parameters

etymology_gloss	Etymology gloss.

Returns

Definition instance.

Definition at line 410 of file definition.py.

6.4.3.26 def lmf.src.core.definition.Definition.set_etymology_source (*self, etymology_source*)

Set etymology source.

Attribute 'etymologySource' is owned by the first Statement.

Parameters

etymology_↩ source	Etymology source.

Returns

Definition instance.

Definition at line 436 of file definition.py.

6.4.3.27 def lmf.src.core.definition.Definition.set_gloss (*self, gloss, language* = `None`)

Set gloss.

Parameters

gloss	Gloss.
language	Language used for the gloss.

Returns

Definition instance.

Definition at line 77 of file definition.py.

6.4.3.28 def lmf.src.core.definition.Definition.set_language (*self, language*)

Set language used for definition and gloss.

Parameters

language	Language used for definition and gloss.

Returns

Definition instance.

Definition at line 39 of file definition.py.

6.4.3.29 def lmf.src.core.definition.Definition.set_note (*self, note, type* = None, *language* = None)

Set note, note type and language.

These attributes are owned by Statement.

Parameters

note	Note to set.
type	Type of the note.
language	Language used for the note.

Returns

Definition instance.

Definition at line 125 of file definition.py.

6.4.3.30 def lmf.src.core.definition.Definition.set_restriction (*self, restriction, language* = None)

Set restriction and language.

These attributes are owned by Statement.

Parameters

restriction	Restriction to set.
language	Language used for the restriction.

Returns

Definition instance.

Definition at line 253 of file definition.py.

6.4.3.31 def lmf.src.core.definition.Definition.set_scientific_name (*self, scientific_name*)

Set scientific name.

Attribute 'scientficName' is owned by the first Statement.

Parameters

scientific_name	Scientific name.

Returns

Definition instance.

Definition at line 462 of file definition.py.

6.4.3.32 def lmf.src.core.definition.Definition.set_usage_note (*self, usage_note, language* = `None`)

Set usage note and language.

These attributes are owned by Statement.

Parameters

usage_note	Usage note to set.
language	Language used for the usage note.

Returns

Definition instance.

Definition at line 173 of file definition.py.

6.4.3.33 def lmf.src.core.definition.Definition.set_written_form (*self, written_form*)

Set loan word.

Attribute 'writtenForm' is owned by the first Statement.

Parameters

written_form	Loan word.

Returns

Definition instance.

Definition at line 319 of file definition.py.

6.4.4 Member Data Documentation

6.4.4.1 lmf.src.core.definition.Definition.definition

Definition at line 18 of file definition.py.

6.4.4.2 lmf.src.core.definition.Definition.gloss

Definition at line 19 of file definition.py.

6.4.4.3 lmf.src.core.definition.Definition.language

Definition at line 17 of file definition.py.

6.4.4.4 lmf.src.core.definition.Definition.literally

Definition at line 20 of file definition.py.

6.4.4.5 lmf.src.core.definition.Definition.statement

Statement instances are owned by Definition There is zero to many Statement instances per Definition.

Definition at line 26 of file definition.py.

6.4.4.6 lmf.src.core.definition.Definition.text_representation

TextRepresentation instances are owned by Definition There is zero to many TextRepresentation instances per Definition.

Definition at line 23 of file definition.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/definition.py

6.5 lmf.src.mrd.equivalent.Equivalent Class Reference

"Equivalent is a class representing the translation equivalent of the word form managed by the Lemma class." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_translation

  Set translation and language.
- def get_translation

  Get translation.
- def set_language

  Set language used for translation.
- def get_language

  Get language used for translation.

Public Attributes

- language
- translation
- text_representation

  TextRepresentation instances are owned by Equivalent There is zero to many TextRepresentation instances per Equivalent.

6.5.1 Detailed Description

"Equivalent is a class representing the translation equivalent of the word form managed by the Lemma class." (LMF)

Definition at line 8 of file equivalent.py.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 def lmf.src.mrd.equivalent.Equivalent.__init__ (*self*)

Constructor.

Equivalent instances are owned by Sense.

Returns

An Equivalent instance.

Definition at line 11 of file equivalent.py.

6.5.2.2 def lmf.src.mrd.equivalent.Equivalent.__del__ (*self*)

Destructor.

Release TextRepresentation instances.

Definition at line 22 of file equivalent.py.

6.5.3 Member Function Documentation

6.5.3.1 def lmf.src.mrd.equivalent.Equivalent.get_language (*self*)

Get language used for translation.

Returns

Equivalent attribute 'language'.

Definition at line 61 of file equivalent.py.

6.5.3.2 def lmf.src.mrd.equivalent.Equivalent.get_translation (*self*, *language* = `None`)

Get translation.

Parameters

language	If this argument is given, get translation only if written in this language.

Returns

The filtered Equivalent attribute 'translation'.

Definition at line 43 of file equivalent.py.

6.5.3.3 def lmf.src.mrd.equivalent.Equivalent.set_language (*self*, *language*)

Set language used for translation.

Parameters

language	Language used for the translation.

Returns

Equivalent instance.

Definition at line 51 of file equivalent.py.

6.5.3.4 def lmf.src.mrd.equivalent.Equivalent.set_translation (*self*, *translation*, *language* = `None`)

Set translation and language.

Parameters

translation	The translation to set.
language	Language used for the translation.

Returns

Equivalent instance.

Definition at line 30 of file equivalent.py.

6.5.4 Member Data Documentation

6.5.4.1 lmf.src.mrd.equivalent.Equivalent.language

Definition at line 16 of file equivalent.py.

6.5.4.2 lmf.src.mrd.equivalent.Equivalent.text_representation

TextRepresentation instances are owned by Equivalent There is zero to many TextRepresentation instances per Equivalent.

Definition at line 20 of file equivalent.py.

6.5.4.3 lmf.src.mrd.equivalent.Equivalent.translation

Definition at line 17 of file equivalent.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/equivalent.py

6.6 lmf.src.utils.error_handling.Error Class Reference

Base class for exceptions in this library.

Inheritance diagram for lmf.src.utils.error_handling.Error:

Public Member Functions

- def __init__
  Constructor.
- def __str__
  Build the string to be displayed.
- def handle
  Define behavior to follow in case this error is caught: diplay error and exit program.

Public Attributes

- msg
- excp
- frame_info

6.6.1 Detailed Description

Base class for exceptions in this library.

Definition at line 3 of file error_handling.py.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 def lmf.src.utils.error_handling.Error.__init__ (*self, msg, excp* = None)

Constructor.

Parameters

msg	String to be reported to user.
excp	Raised system exception if any: IOError, KeyboardInterrupt, SystemExit, IndexError, KeyError, AttributeError, TypeError, NameError, UnboundLocalError, ValueError.

Returns

An Error instance.

Definition at line 6 of file error_handling.py.

6.6.3 Member Function Documentation

6.6.3.1 def lmf.src.utils.error_handling.Error.__str__ (*self*)

Build the string to be displayed.

Returns

A Python string.

Definition at line 18 of file error_handling.py.

6.6.3.2 def lmf.src.utils.error_handling.Error.handle (*self*)

Define behavior to follow in case this error is caught: diplay error and exit program.

Definition at line 30 of file error_handling.py.

6.6.4 Member Data Documentation

6.6.4.1 lmf.src.utils.error_handling.Error.excp

Definition at line 13 of file error_handling.py.

6.6.4.2 lmf.src.utils.error_handling.Error.frame_info

Definition at line 16 of file error_handling.py.

6.6.4.3 lmf.src.utils.error_handling.Error.msg

Definition at line 12 of file error_handling.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py

6.7 lmf.src.core.form.Form Class Reference

"Form is an abstract class representing a lexeme, a morphological variant of a lexeme or a morph. The Form class allows subclasses." (LMF)

Public Member Functions

- def __init__

  As Form is an abstract class, constructor raises an error.
- def __del__

  As Form is an abstract class, desctructor raises an error.
- def __new__

  Private initialization called from Form subclasses.

Public Attributes

- form_representation

  FormRepresentation instances are owned by Form subclasses There is zero to many FormRepresentation instances per Form subclass.

6.7.1 Detailed Description

"Form is an abstract class representing a lexeme, a morphological variant of a lexeme or a morph. The Form class allows subclasses." (LMF)

Definition at line 6 of file form.py.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 def lmf.src.core.form.Form.__init__ (*self*)

As Form is an abstract class, constructor raises an error.

Definition at line 9 of file form.py.

6.7.2.2 def lmf.src.core.form.Form.__del__ (*self*)

As Form is an abstract class, desctructor raises an error.

Definition at line 14 of file form.py.

6.7.3 Member Function Documentation

6.7.3.1 def lmf.src.core.form.Form.__new__ (*self*)

Private initialization called from Form subclasses.

Form subinstances are owned by LexicalEntry.

Definition at line 19 of file form.py.

6.7.4 Member Data Documentation

6.7.4.1 lmf.src.core.form.Form.form_representation

FormRepresentation instances are owned by Form subclasses There is zero to many FormRepresentation instances per Form subclass.

Definition at line 25 of file form.py.

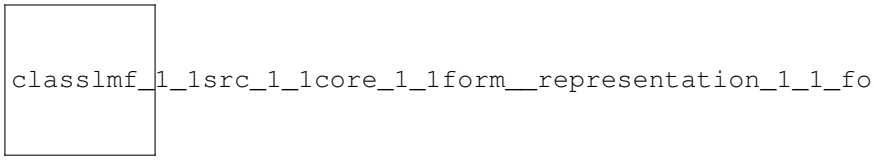
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form.py

6.8 lmf.src.core.form_representation.FormRepresentation Class Reference

"Form Representation is a class representing one variant orthography of a Form." (LMF)

Inheritance diagram for lmf.src.core.form_representation.FormRepresentation:



Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def get_speakers
  Get speakers.
- def set_writtenForm
  Set written form and script.
- def get_writtenForm
  Get written form.
- def set_variantForm
  Set variant form.
- def get_variantForm
  Get variant form.
- def set_type
  Set variant type.
- def get_type

Get variant type.

- def set_comment

  Set variant form comment.
- def get_comment

  Get variant form comment.
- def set_language

  Set language used for comment.
- def get_language

  Get language used for comment.
- def set_tone

  Set tone.
- def get_tone

  Get tone.
- def set_geographicalVariant

  Set geographical variant.
- def get_geographicalVariant

  Get geographical variant.
- def set_phoneticForm

  Set phonetic form.
- def get_phoneticForm

  Get phonetic form.
- def set_contextualVariation

  Set contextual variation.
- def get_contextualVariation

  Get contextual variation.
- def set_spellingVariant

  Set spelling variant.
- def get_spellingVariant

  Get spelling variant.
- def set_citationForm

  Set citation form.
- def get_citationForm

  Get citation form.
- def set_dialect

  Set dialect.
- def get_dialect

  Get dialect.
- def set_transliteration

  Set transliteration.
- def get_transliteration

  Get transliteration.
- def set_scriptName

  Set script name.
- def get_scriptName

  Get script name.
- def create_audio

  Create an Audio instance.
- def get_audio

  Get the audio resource maintained by the form representation.
- def set_audio

  Set audio resource.

Public Attributes

- variantForm
- type
- transliteration
- tone
- geographicalVariant
- phoneticForm
- contextualVariation
- spellingVariant
- citationForm
- dialect
- audio

  Audio instance is owned by FormRepresentation There is zero or one Audio instance per FormRepresentation.
- targets
- writtenForm
- comment
- language
- scriptName

6.8.1 Detailed Description

"Form Representation is a class representing one variant orthography of a Form." (LMF)

Definition at line 11 of file form_representation.py.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 def lmf.src.core.form_representation.FormRepresentation.__init__ (*self*)

Constructor.

FormRepresentation instances are owned by Form.

Returns

A FormRepresentation instance.

Definition at line 14 of file form_representation.py.

6.8.2.2 def lmf.src.core.form_representation.FormRepresentation.__del__ (*self*)

Destructor.

Release Audio instances.

Definition at line 40 of file form_representation.py.

6.8.3 Member Function Documentation

6.8.3.1 def lmf.src.core.form_representation.FormRepresentation.create_audio (*self*)

Create an Audio instance.

Returns

Audio instance.

Definition at line 297 of file form_representation.py.

6.8.3.2 def lmf.src.core.form_representation.FormRepresentation.get_audio (*self*)

Get the audio resource maintained by the form representation.

Returns

Audio instance.

Definition at line 303 of file form_representation.py.

6.8.3.3 def lmf.src.core.form_representation.FormRepresentation.get_citationForm (*self*)

Get citation form.

Returns

FormRepresentation attribute 'citationForm'.

Definition at line 243 of file form_representation.py.

6.8.3.4 def lmf.src.core.form_representation.FormRepresentation.get_comment (*self, language* = `None`)

Get variant form comment.

Parameters

language	If this argument is given, get comment only if written in this language.

Returns

The filtered Representation attribute 'comment'.

Definition at line 126 of file form_representation.py.

6.8.3.5 def lmf.src.core.form_representation.FormRepresentation.get_contextualVariation (*self*)

Get contextual variation.

Returns

FormRepresentation attribute 'contextualVariation'.

Definition at line 211 of file form_representation.py.

6.8.3.6 def lmf.src.core.form_representation.FormRepresentation.get_dialect (*self*)

Get dialect.

Returns

FormRepresentation attribute 'dialect'.

Definition at line 259 of file form_representation.py.

6.8.3.7 def lmf.src.core.form_representation.FormRepresentation.get_geographicalVariant (*self*)

Get geographical variant.

Returns

FormRepresentation attribute 'geographicalVariant'.

Definition at line 179 of file form_representation.py.

6.8.3.8 def lmf.src.core.form_representation.FormRepresentation.get_language (*self*)

Get language used for comment.

Returns

Representation attribute 'language'.

Definition at line 147 of file form_representation.py.

6.8.3.9 def lmf.src.core.form_representation.FormRepresentation.get_phoneticForm (*self*)

Get phonetic form.

Returns

FormRepresentation attribute 'phoneticForm'.

Definition at line 195 of file form_representation.py.

6.8.3.10 def lmf.src.core.form_representation.FormRepresentation.get_scriptName (*self*)

Get script name.

Returns

Representation attribute 'scriptName'.

Definition at line 291 of file form_representation.py.

6.8.3.11 def lmf.src.core.form_representation.FormRepresentation.get_speakers (*self*)

Get speakers.

Returns

FormRepresentation private attribute '__speaker', a Python list of Speaker instances.

Definition at line 49 of file form_representation.py.

6.8.3.12 def lmf.src.core.form_representation.FormRepresentation.get_spellingVariant (*self*)

Get spelling variant.

Returns

FormRepresentation attribute 'spellingVariant'.

Definition at line 227 of file form_representation.py.

6.8.3.13 def lmf.src.core.form_representation.FormRepresentation.get_tone (*self*)

Get tone.

Returns

FormRepresentation attribute 'tone'.

Definition at line 163 of file form_representation.py.

6.8.3.14 def lmf.src.core.form_representation.FormRepresentation.get_transliteration (*self*)

Get transliteration.

Returns

FormRepresentation attribute 'transliteration'.

Definition at line 275 of file form_representation.py.

6.8.3.15 def lmf.src.core.form_representation.FormRepresentation.get_type (*self*)

Get variant type.

Returns

FormRepresentation attribute 'type'.

Definition at line 106 of file form_representation.py.

6.8.3.16 def lmf.src.core.form_representation.FormRepresentation.get_variantForm (*self*)

Get variant form.

Returns

FormRepresentation attribute 'variantForm'.

Definition at line 86 of file form_representation.py.

6.8.3.17 def lmf.src.core.form_representation.FormRepresentation.get_writtenForm (*self, script_name* = `None`)

Get written form.

Parameters

script_name	If this argument is given, get written form only if written using this script.

Returns

The filtered Representation attribute 'writtenForm'.

Definition at line 68 of file form_representation.py.

6.8.3.18 def lmf.src.core.form_representation.FormRepresentation.set_audio (*self, media_type, file_name, author, quality, start_position, duration, external_reference, audio_file_format*)

Set audio resource.

Attributes 'mediaType', 'fileName', 'author', 'quality', 'startPosition', 'durationOfEffectiveSpeech', 'external↩ Reference', 'audioFileFormat' are owned by Material/Audio.

Parameters

media_type	The media type to set.
file_name	Name of the audio file.
author	Author of the recording.
quality	Quality of the recording, in range 'quality_range' defined in 'common/range.py'.
start_position	Start position of the form in the recording, in format 'Thh:mm:ss,msms', e.g. "T00:05:00".
duration	Duration of the effcetive speech, in format 'PThhHmmMssS', e.g. "PT00:05:00".
external_reference	Reference of the audio file, if not directly provided.
audio_file_format	Format of the audio file, e.g. "wav".

Returns

FormRepresentation instance.

Definition at line 309 of file form_representation.py.

6.8.3.19 def lmf.src.core.form_representation.FormRepresentation.set_citationForm (*self, citation_form*)

Set citation form.

Parameters

citation_form	The citation form to set.

Returns

FormRepresentation instance.

Definition at line 233 of file form_representation.py.

6.8.3.20 def lmf.src.core.form_representation.FormRepresentation.set_comment (*self, comment, language* = `None`)

Set variant form comment.

Parameters

comment	Comment about the variant form.
language	Language used for the comment.

Returns

FormRepresentation instance.

Definition at line 112 of file form_representation.py.

6.8.3.21 def lmf.src.core.form_representation.FormRepresentation.set_contextualVariation (*self, contextual_variation*)

Set contextual variation.

Parameters

contextual↩ Variation	The contextual variation to set.

Returns

FormRepresentation instance.

Definition at line 201 of file form_representation.py.

6.8.3.22 def lmf.src.core.form_representation.FormRepresentation.set_dialect (*self, dialect*)

Set dialect.

Parameters

dialect	The dialect to set.

Returns

FormRepresentation instance.

Definition at line 249 of file form_representation.py.

6.8.3.23 def lmf.src.core.form_representation.FormRepresentation.set_geographicalVariant (*self, geographical_variant*)

Set geographical variant.

Parameters

geographical_↩ variant	The geographical variant to set.

Returns

FormRepresentation instance.

Definition at line 169 of file form_representation.py.

6.8.3.24 def lmf.src.core.form_representation.FormRepresentation.set_language (*self, language*)

Set language used for comment.

Parameters

language	Language used for the comment.

Returns

FormRepresentation instance.

Definition at line 136 of file form_representation.py.

6.8.3.25 def lmf.src.core.form_representation.FormRepresentation.set_phoneticForm (*self, phonetic_form*)

Set phonetic form.

Parameters

phonetic_form	The phonetic form to set.

Returns

FormRepresentation instance.

Definition at line 185 of file form_representation.py.

6.8.3.26 def lmf.src.core.form_representation.FormRepresentation.set_scriptName (*self, script_name*)

Set script name.

Parameters

script_name	The script name to set.

Returns

FormRepresentation instance.

Definition at line 281 of file form_representation.py.

6.8.3.27 def lmf.src.core.form_representation.FormRepresentation.set_spellingVariant (*self, spelling_variant*)

Set spelling variant.

Parameters

spelling_variant	The spelling variant to set.

Returns

FormRepresentation instance.

Definition at line 217 of file form_representation.py.

6.8.3.28 def lmf.src.core.form_representation.FormRepresentation.set_tone (*self, tone*)

Set tone.

Parameters

tone	The tone to set.

Returns

FormRepresentation instance.

Definition at line 153 of file form_representation.py.

6.8.3.29 def lmf.src.core.form_representation.FormRepresentation.set_transliteration (*self, transliteration*)

Set transliteration.

Parameters

transliteration	The transliteration to set.

Returns

FormRepresentation instance.

Definition at line 265 of file form_representation.py.

6.8.3.30 def lmf.src.core.form_representation.FormRepresentation.set_type (*self, type*)

Set variant type.

Parameters

type	Type of variant, in range 'type_variant_range' defined in 'common/range.py'.

Returns

FormRepresentation instance.

Definition at line 92 of file form_representation.py.

6.8.3.31 def lmf.src.core.form_representation.FormRepresentation.set_variantForm (*self, variant_form*)

Set variant form.

Parameters

variant_form	The variant form to set.

Returns

FormRepresentation instance.

Definition at line 76 of file form_representation.py.

6.8.3.32 def lmf.src.core.form_representation.FormRepresentation.set_writtenForm (*self, written_form, script_name =* `None`)

Set written form and script.

Parameters

written_form	The written form to set.
script_name	Script used for the written form.

Returns

FormRepresentation instance.

Definition at line 55 of file form_representation.py.

6.8.4 Member Data Documentation

6.8.4.1 lmf.src.core.form_representation.FormRepresentation.audio

Audio instance is owned by FormRepresentation There is zero or one Audio instance per FormRepresentation.

Definition at line 33 of file form_representation.py.

6.8.4.2 lmf.src.core.form_representation.FormRepresentation.citationForm

Definition at line 29 of file form_representation.py.

6.8.4.3 lmf.src.core.form_representation.FormRepresentation.comment

Definition at line 121 of file form_representation.py.

6.8.4.4 lmf.src.core.form_representation.FormRepresentation.contextualVariation

Definition at line 27 of file form_representation.py.

6.8.4.5 lmf.src.core.form_representation.FormRepresentation.dialect

Definition at line 30 of file form_representation.py.

6.8.4.6 lmf.src.core.form_representation.FormRepresentation.geographicalVariant

Definition at line 25 of file form_representation.py.

6.8.4.7 lmf.src.core.form_representation.FormRepresentation.language

Definition at line 144 of file form_representation.py.

6.8.4.8 lmf.src.core.form_representation.FormRepresentation.phoneticForm

Definition at line 26 of file form_representation.py.

6.8.4.9 lmf.src.core.form_representation.FormRepresentation.scriptName

Definition at line 288 of file form_representation.py.

6.8.4.10 lmf.src.core.form_representation.FormRepresentation.spellingVariant

Definition at line 28 of file form_representation.py.

6.8.4.11 lmf.src.core.form_representation.FormRepresentation.targets

Definition at line 35 of file form_representation.py.

6.8.4.12 lmf.src.core.form_representation.FormRepresentation.tone

Definition at line 24 of file form_representation.py.

6.8.4.13 lmf.src.core.form_representation.FormRepresentation.transliteration

Definition at line 23 of file form_representation.py.

6.8.4.14 lmf.src.core.form_representation.FormRepresentation.type

Definition at line 22 of file form_representation.py.

6.8.4.15 lmf.src.core.form_representation.FormRepresentation.variantForm

Definition at line 21 of file form_representation.py.

6.8.4.16 lmf.src.core.form_representation.FormRepresentation.writtenForm

Definition at line 63 of file form_representation.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form_representation.py

6.9 lmf.src.core.global_information.GlobalInformation Class Reference

"Global Information is a class for administrative information and other general attributes, such as /language coding/ or /script coding/, which are valid for the entire lexical resource." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_languageCode

  Set global information language code.
- def get_languageCode

  Get global information language code.
- def set_version

  Set global information version.
- def get_version

  Get global information version.
- def set_license

  Set global information license.
- def get_license

  Get global information license.
- def set_characterEncoding

  Set global information character encoding.
- def get_characterEncoding

  Get global information character encoding.
- def set_dateCoding

  Set global information date coding.
- def get_dateCoding

  Get global information date coding.
- def set_projectName

  Set global information project name.
- def get_projectName

Get global information project name.

- def set_creationDate

  Set global information creation date.
- def get_creationDate

  Get global information creation date.
- def set_lastUpdate

  Set global information last update.
- def get_lastUpdate

  Get global information last update.
- def set_author

  Set global information author.
- def get_author

  Get global information author.
- def set_description

  Set global information description.
- def get_description

  Get global information description.
- def compute_bibliographicCitation

  Compute bibliographic citation from date and author.
- def get_bibliographicCitation

  Get global information bibliographic citation.

Public Attributes

- languageCode
- author
- version
- lastUpdate
- license
- characterEncoding
- dateCoding
- creationDate
- projectName
- description
- bibliographicCitation

6.9.1 Detailed Description

"Global Information is a class for administrative information and other general attributes, such as /language coding/ or /script coding/, which are valid for the entire lexical resource." (LMF)

Definition at line 8 of file global_information.py.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 def lmf.src.core.global_information.GlobalInformation.__init__ (*self*)

Constructor.

GlobalInformation instance is owned by LexicalResource.

Returns

A GlobalInformation instance.

Definition at line 11 of file global_information.py.

6.9.2.2 def lmf.src.core.global_information.GlobalInformation.__del__ (*self*)

Destructor.

Definition at line 28 of file global_information.py.

6.9.3 Member Function Documentation

6.9.3.1 def lmf.src.core.global_information.GlobalInformation.compute_bibliographicCitation (*self*)

Compute bibliographic citation from date and author.

Set GlobalInformation attribute 'bibliographicCitation'.

Definition at line 175 of file global_information.py.

6.9.3.2 def lmf.src.core.global_information.GlobalInformation.get_author (*self*)

Get global information author.

Returns

GlobalInformation attribute 'author'.

Definition at line 155 of file global_information.py.

6.9.3.3 def lmf.src.core.global_information.GlobalInformation.get_bibliographicCitation (*self*)

Get global information bibliographic citation.

Returns

GlobalInformation attribute 'bibliographicCitation'.

Definition at line 185 of file global_information.py.

6.9.3.4 def lmf.src.core.global_information.GlobalInformation.get_characterEncoding (*self*)

Get global information character encoding.

Returns

GlobalInformation attribute 'characterEncoding'.

Definition at line 83 of file global_information.py.

6.9.3.5 def lmf.src.core.global_information.GlobalInformation.get_creationDate (*self*)

Get global information creation date.

Returns

GlobalInformation attribute 'creationDate'.

Definition at line 126 of file global_information.py.

6.9.3.6 def lmf.src.core.global_information.GlobalInformation.get_dateCoding (*self*)

Get global information date coding.

Returns

GlobalInformation attribute 'dateCoding'.

Definition at line 97 of file global_information.py.

6.9.3.7 def lmf.src.core.global_information.GlobalInformation.get_description (*self*)

Get global information description.

Returns

GlobalInformation attribute 'description'.

Definition at line 169 of file global_information.py.

6.9.3.8 def lmf.src.core.global_information.GlobalInformation.get_languageCode (*self*)

Get global information language code.

Returns

GlobalInformation attribute 'languageCode'.

Definition at line 41 of file global_information.py.

6.9.3.9 def lmf.src.core.global_information.GlobalInformation.get_lastUpdate (*self*)

Get global information last update.

Returns

GlobalInformation attribute 'lastUpdate'.

Definition at line 141 of file global_information.py.

6.9.3.10 def lmf.src.core.global_information.GlobalInformation.get_license (*self*)

Get global information license.

Returns

GlobalInformation attribute 'license'.

Definition at line 69 of file global_information.py.

6.9.3.11 def lmf.src.core.global_information.GlobalInformation.get_projectName (*self*)

Get global information project name.

Returns

GlobalInformation attribute 'projectName'.

Definition at line 111 of file global_information.py.

6.9.3.12 def lmf.src.core.global_information.GlobalInformation.get_version (*self*)

Get global information version.

Returns

GlobalInformation attribute 'version'.

Definition at line 55 of file global_information.py.

6.9.3.13 def lmf.src.core.global_information.GlobalInformation.set_author (*self, author*)

Set global information author.

Parameters

author	The author's name to set.

Returns

GlobalInformation instance.

Definition at line 147 of file global_information.py.

6.9.3.14 def lmf.src.core.global_information.GlobalInformation.set_characterEncoding (*self, character_encoding*)

Set global information character encoding.

Parameters

character_↩ encoding	The character encoding to use.

Returns

GlobalInformation instance.

Definition at line 75 of file global_information.py.

6.9.3.15 def lmf.src.core.global_information.GlobalInformation.set_creationDate (*self, date*)

Set global information creation date.

Parameters

date	The date to set.

Returns

GlobalInformation instance.

Definition at line 117 of file global_information.py.

6.9.3.16 def lmf.src.core.global_information.GlobalInformation.set_dateCoding (*self, date_coding*)

Set global information date coding.

Parameters

date_coding	The date coding to use.

Returns

GlobalInformation instance.

Definition at line 89 of file global_information.py.

6.9.3.17 def lmf.src.core.global_information.GlobalInformation.set_description (*self, description*)

Set global information description.

Parameters

description	The description to set.

Returns

GlobalInformation instance.

Definition at line 161 of file global_information.py.

6.9.3.18 def lmf.src.core.global_information.GlobalInformation.set_languageCode (*self, language_code*)

Set global information language code.

Parameters

language_code	The language code to use.

Returns

GlobalInformation instance.

Definition at line 33 of file global_information.py.

6.9.3.19 def lmf.src.core.global_information.GlobalInformation.set_lastUpdate (*self, date*)

Set global information last update.

Parameters

date	The date to set.

Returns

GlobalInformation instance.

Definition at line 132 of file global_information.py.

6.9.3.20 def lmf.src.core.global_information.GlobalInformation.set_license (*self, license*)

Set global information license.

Parameters

license	The license to set.

Returns

GlobalInformation instance.

Definition at line 61 of file global_information.py.

6.9.3.21 def lmf.src.core.global_information.GlobalInformation.set_projectName (*self, project_name*)

Set global information project name.

Parameters

project_name	The project name to set.

Returns

GlobalInformation instance.

Definition at line 103 of file global_information.py.

6.9.3.22 def lmf.src.core.global_information.GlobalInformation.set_version (*self, version*)

Set global information version.

Parameters

version	The version to set.

Returns

GlobalInformation version.

Definition at line 47 of file global_information.py.

6.9.4 Member Data Documentation

6.9.4.1 lmf.src.core.global_information.GlobalInformation.author

Definition at line 17 of file global_information.py.

6.9.4.2 lmf.src.core.global_information.GlobalInformation.bibliographicCitation

Definition at line 26 of file global_information.py.

6.9.4.3 lmf.src.core.global_information.GlobalInformation.characterEncoding

Definition at line 21 of file global_information.py.

6.9.4.4 lmf.src.core.global_information.GlobalInformation.creationDate

Definition at line 23 of file global_information.py.

6.9.4.5 lmf.src.core.global_information.GlobalInformation.dateCoding

Definition at line 22 of file global_information.py.

6.9.4.6 lmf.src.core.global_information.GlobalInformation.description

Definition at line 25 of file global_information.py.

6.9.4.7 lmf.src.core.global_information.GlobalInformation.languageCode

Definition at line 16 of file global_information.py.

6.9.4.8 lmf.src.core.global_information.GlobalInformation.lastUpdate

Definition at line 19 of file global_information.py.

6.9.4.9 lmf.src.core.global_information.GlobalInformation.license

Definition at line 20 of file global_information.py.

6.9.4.10 lmf.src.core.global_information.GlobalInformation.projectName

Definition at line 24 of file global_information.py.

6.9.4.11 lmf.src.core.global_information.GlobalInformation.version

Definition at line 18 of file global_information.py.

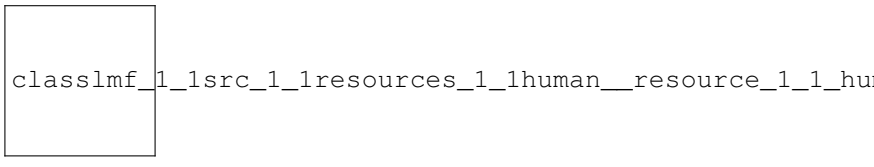
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/global_information.py

6.10 lmf.src.resources.human_resource.HumanResource Class Reference

HumanResource is a Resource subclass.

Inheritance diagram for lmf.src.resources.human_resource.HumanResource:



Public Member Functions

- def __init__

  As HumanResource is an abstract class, constructor raises an error.
- def __del__

  As HumanResource is an abstract class, desctructor raises an error.
- def __new__

  Private initialization called from HumanResource subclasses.

Public Attributes

- name
- anonymizationFlag
- reference
- source

6.10.1 Detailed Description

HumanResource is a Resource subclass.

HumanResource is an abstract class representing a speaker. The HumanResource class allows subclasses.

Definition at line 8 of file human_resource.py.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 def lmf.src.resources.human_resource.HumanResource.__init__ (*self*)

As HumanResource is an abstract class, constructor raises an error.

Definition at line 11 of file human_resource.py.

6.10.2.2 def lmf.src.resources.human_resource.HumanResource.__del__ (*self*)

As HumanResource is an abstract class, desctructor raises an error.

Definition at line 16 of file human_resource.py.

6.10.3 Member Function Documentation

6.10.3.1 def lmf.src.resources.human_resource.HumanResource.__new__ (*self*)

Private initialization called from HumanResource subclasses.

HumanResource subinstances are owned by LexicalResource.

Definition at line 21 of file human_resource.py.

6.10.4 Member Data Documentation

6.10.4.1 lmf.src.resources.human_resource.HumanResource.anonymizationFlag

Definition at line 26 of file human_resource.py.

6.10.4.2 lmf.src.resources.human_resource.HumanResource.name

Definition at line 25 of file human_resource.py.

6.10.4.3 lmf.src.resources.human_resource.HumanResource.reference

Definition at line 27 of file human_resource.py.

6.10.4.4 lmf.src.resources.human_resource.HumanResource.source

Definition at line 28 of file human_resource.py.

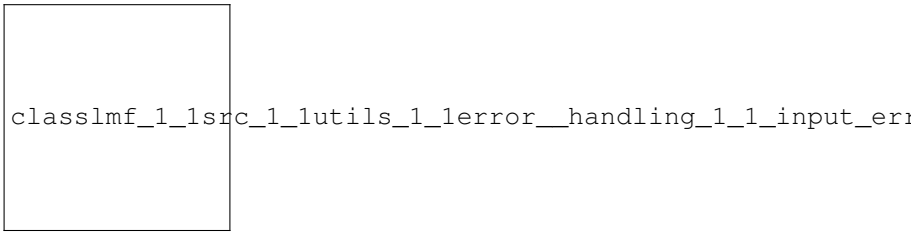
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/human_resource.py

6.11 lmf.src.utils.error_handling.InputError Class Reference

Exception raised for errors in the input.

Inheritance diagram for lmf.src.utils.error_handling.InputError:



Public Member Functions

- def __init__

  Constructor.
- def handle

  Define behavior to follow in case this error is caught: display error and exit program.

Public Attributes

- msg
- expr
- frame_info

6.11.1 Detailed Description

Exception raised for errors in the input.

Definition at line 41 of file error_handling.py.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 def lmf.src.utils.error_handling.InputError.__init__ (*self*, *msg*, *expr* = `None`)

Constructor.

Parameters

msg	Explanation of the error.
expr	Input expression in which the error occurred.

Returns

An InputError instance.

Definition at line 44 of file error_handling.py.

6.11.3 Member Function Documentation

6.11.3.1 def lmf.src.utils.error_handling.InputError.handle (*self*)

Define behavior to follow in case this error is caught: display error and exit program.

Definition at line 56 of file error_handling.py.

6.11.4 Member Data Documentation

6.11.4.1 lmf.src.utils.error_handling.InputError.expr

Definition at line 51 of file error_handling.py.

6.11.4.2 lmf.src.utils.error_handling.InputError.frame_info

Definition at line 54 of file error_handling.py.

6.11.4.3 lmf.src.utils.error_handling.InputError.msg

Definition at line 50 of file error_handling.py.

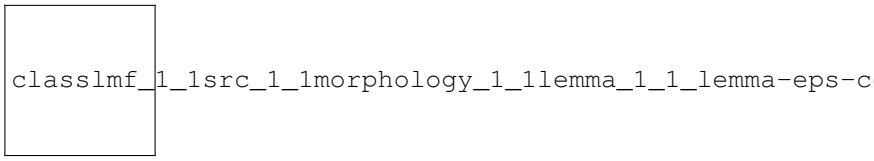
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py

6.12 lmf.src.morphology.lemma.Lemma Class Reference

"Lemma is a Form subclass representing a form chosen by convention to designate the Lexical Entry. The lemma is usually equivalent to one of the inflected forms, the root, stem or compound phrase." (LMF).

Inheritance diagram for lmf.src.morphology.lemma.Lemma:



Public Member Functions

- def __init__

  Constructor.

- def __del__
  Destructor.
- def set_lexeme
  Set lexeme.
- def get_lexeme
  Get lexeme.
- def create_form_representation
  Create a form representation.
- def add_form_representation
  Add a form representation to the lemma.
- def find_form_representations
  Find variant forms.
- def get_form_representations
  Get all form representations maintained by the lemma.
- def get_form_representation
  Get a given form representation maintained by the lemma.
- def set_variant_form
  Set variant form and type.
- def get_variant_forms
  Get all variant forms of specified type.
- def set_variant_comment
  Set variant comment and language.
- def set_tone
  Set tone.
- def get_tones
  Get all tones.
- def set_geographical_variant
  Set geographical variant.
- def set_phonetic_form
  Set phonetic form.
- def get_phonetic_forms
  Get all phonetic forms.
- def set_contextual_variation
  Set contextual variation.
- def get_contextual_variations
  Get all contextual variations.
- def set_spelling_variant
  Set spelling variant.
- def get_spelling_variants
  Get all spelling variants.
- def set_citation_form
  Set citation form.
- def get_citation_forms
  Get all citation forms.
- def set_dialect
  Set dialect.
- def set_transliteration
  Set transliteration.
- def get_transliterations
  Get all transliterations.
- def set_script_name
  Set script name.
- def set_audio
  Set audio resource.

Public Attributes

- hyphenation
- lexeme

6.12.1 Detailed Description

"Lemma is a Form subclass representing a form chosen by convention to designate the Lexical Entry. The lemma is usually equivalent to one of the inflected forms, the root, stem or compound phrase." (LMF).

Definition at line 9 of file lemma.py.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 def lmf.src.morphology.lemma.Lemma.__init__ (*self*)

Constructor.

Lemma instance is owned by LexicalEntry.

Returns

A Lemma instance.

Definition at line 12 of file lemma.py.

6.12.2.2 def lmf.src.morphology.lemma.Lemma.__del__ (*self*)

Destructor.

Definition at line 22 of file lemma.py.

6.12.3 Member Function Documentation

6.12.3.1 def lmf.src.morphology.lemma.Lemma.add_form_representation (*self, form_representation*)

Add a form representation to the lemma.

Parameters

form_↩ representation	The FormRepresentation instance to add to the lemma.

Returns

Lemma instance.

Definition at line 47 of file lemma.py.

6.12.3.2 def lmf.src.morphology.lemma.Lemma.create_form_representation (*self*)

Create a form representation.

Returns

FormRepresentation instance.

Definition at line 41 of file lemma.py.

6.12.3.3 def lmf.src.morphology.lemma.Lemma.find_form_representations (*self, type*)

Find variant forms.

This attribute is owned by FormRepresentation.

Parameters

type	The type to consider to retrieve the variant form.

Returns

A Python list of found FormRepresentation attributes 'variantForm'.

Definition at line 55 of file lemma.py.

6.12.3.4 def lmf.src.morphology.lemma.Lemma.get_citation_forms (*self, script_name* = `None`)

Get all citation forms.

This attribute is owned by FormRepresentation.

Parameters

script_name	If provided, get only citation forms that are written using this script.

Returns

A Python list of FormRepresentation attributes 'citationForm'.

Definition at line 299 of file lemma.py.

6.12.3.5 def lmf.src.morphology.lemma.Lemma.get_contextual_variations (*self*)

Get all contextual variations.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'contextualVariation'.

Definition at line 236 of file lemma.py.

6.12.3.6 def lmf.src.morphology.lemma.Lemma.get_form_representation (*self, index*)

Get a given form representation maintained by the lemma.

Parameters

index	The index of the wanted form representation.

Returns

The wanted FormRepresentation instance.

Definition at line 73 of file lemma.py.

6.12.3.7 def lmf.src.morphology.lemma.Lemma.get_form_representations (*self*)

Get all form representations maintained by the lemma.

Returns

A Python list of form representations.

Definition at line 67 of file lemma.py.

6.12.3.8 def lmf.src.morphology.lemma.Lemma.get_lexeme (*self*)

Get lexeme.

Returns

Lemma attribute 'lexeme'.

Definition at line 35 of file lemma.py.

6.12.3.9 def lmf.src.morphology.lemma.Lemma.get_phonetic_forms (*self, script_name* = `None`)

Get all phonetic forms.

This attribute is owned by FormRepresentation.

Parameters

script_name	If provided, get only phonetic forms that are written using this script.

Returns

A Python list of FormRepresentation attributes 'phoneticForm'.

Definition at line 205 of file lemma.py.

6.12.3.10 def lmf.src.morphology.lemma.Lemma.get_spelling_variants (*self*)

Get all spelling variants.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'spellingVariant'.

Definition at line 266 of file lemma.py.

6.12.3.11 def lmf.src.morphology.lemma.Lemma.get_tones (*self*)

Get all tones.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'tone'.

Definition at line 153 of file lemma.py.

6.12.3.12 def lmf.src.morphology.lemma.Lemma.get_transliterations (*self*)

Get all transliterations.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'transliteration'.

Definition at line 349 of file lemma.py.

6.12.3.13 def lmf.src.morphology.lemma.Lemma.get_variant_forms (*self, type* = "unspecified")

Get all variant forms of specified type.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'variantForm' if type matches.

Definition at line 103 of file lemma.py.

6.12.3.14 def lmf.src.morphology.lemma.Lemma.set_audio (*self, media_type, file_name, author, quality, start_position, duration, external_reference, audio_file_format*)

Set audio resource.

Attributes 'mediaType', 'fileName', 'author', 'quality', 'startPosition', 'durationOfEffectiveSpeech', 'external↩ Reference', 'audioFileFormat' are owned by Material/Audio, which is owned by FormRepresentation.

Parameters

media_type	The media type to set.
file_name	Name of the audio file.
author	Author of the recording.
quality	Quality of the recording, in range 'quality_range' defined in 'common/range.py'.
start_position	Start position of the form in the recording, in format 'Thh:mm:ss,msms', e.g. "T00:05:00".
duration	Duration of the effcetive speech, in format 'PThhHmmMssS', e.g. "PT00:05:00".
external_↩ reference	Reference of the audio file, if not directly provided.
audio_file_↩ format	Format of the audio file, e.g. "wav".

Returns

Lemma instance.

Definition at line 379 of file lemma.py.

6.12.3.15 def lmf.src.morphology.lemma.Lemma.set_citation_form (*self, citation_form, script_name* = None)

Set citation form.

This attribute is owned by FormRepresentation.

Parameters

citation_form	The citation form to set.
script_name	The name of the script used to write the citation form, e.g. devanagari.

Returns

Lemma instance.

Definition at line 277 of file lemma.py.

6.12.3.16 def lmf.src.morphology.lemma.Lemma.set_contextual_variation (*self, contextual_variation*)

Set contextual variation.

This attribute is owned by FormRepresentation.

Parameters

contextual_↩ variation	The contextual variation to set.

Returns

Lemma instance.

Definition at line 217 of file lemma.py.

6.12.3.17 def lmf.src.morphology.lemma.Lemma.set_dialect (*self, dialect*)

Set dialect.

This attribute is owned by FormRepresentation.

Parameters

dialect	The dialect to set.

Returns

Lemma instance.

Definition at line 311 of file lemma.py.

6.12.3.18 def lmf.src.morphology.lemma.Lemma.set_geographical_variant (*self, geographical_variant*)

Set geographical variant.

This attribute is owned by FormRepresentation.

Parameters

geographical_↩ variant	The geographical variant to set.

Returns

Lemma instance.

Definition at line 164 of file lemma.py.

6.12.3.19 def lmf.src.morphology.lemma.Lemma.set_lexeme (*self, lexeme*)

Set lexeme.

Parameters

lexeme	The lexeme to set.

Returns

Lemma instance.

Definition at line 27 of file lemma.py.

6.12.3.20 def lmf.src.morphology.lemma.Lemma.set_phonetic_form (*self, phonetic_form, script_name* = `None`)

Set phonetic form.

This attribute is owned by FormRepresentation.

Parameters

phonetic_form	The phonetic form to set.
script_name	The name of the script used to write the phonetic form, e.g. pinyin.

Returns

Lemma instance.

Definition at line 183 of file lemma.py.

6.12.3.21 def lmf.src.morphology.lemma.Lemma.set_script_name (*self, script_name*)

Set script name.

This attribute is owned by FormRepresentation.

Parameters

script_name	The script name to set.

Returns

Lemma instance.

Definition at line 360 of file lemma.py.

6.12.3.22 def lmf.src.morphology.lemma.Lemma.set_spelling_variant (*self, spelling_variant*)

Set spelling variant.

This attribute is owned by FormRepresentation.

Parameters

spelling_variant	The spelling variant to set.

Returns

Lemma instance.

Definition at line 247 of file lemma.py.

6.12.3.23 def lmf.src.morphology.lemma.Lemma.set_tone (*self, tone*)

Set tone.

This attribute is owned by FormRepresentation.

Parameters

tone	The tone to set.

Returns

Lemma instance.

Definition at line 134 of file lemma.py.

6.12.3.24 def lmf.src.morphology.lemma.Lemma.set_transliteration (*self, transliteration*)

Set transliteration.

This attribute is owned by FormRepresentation.

Parameters

transliteration	The transliteration to set.

Returns

Lemma instance.

Definition at line 330 of file lemma.py.

6.12.3.25 def lmf.src.morphology.lemma.Lemma.set_variant_comment (*self, comment, language* = `None`)

Set variant comment and language.

These attributes are owned by FormRepresentation.

Parameters

comment	Variant comment.
language	Language of comment.

Returns

Lemma instance.

Definition at line 114 of file lemma.py.

6.12.3.26 def lmf.src.morphology.lemma.Lemma.set_variant_form (*self, variant_form, type* = `"unspecified"`)

Set variant form and type.

These attributes are owned by FormRepresentation.

Parameters

variant_form	Variant form.
type	Type of variant, in range 'type_variant_range' defined in 'common/range.py'.

Returns

Lemma instance.

Definition at line 83 of file lemma.py.

6.12.4 Member Data Documentation

6.12.4.1 lmf.src.morphology.lemma.Lemma.hyphenation

Definition at line 19 of file lemma.py.

6.12.4.2 lmf.src.morphology.lemma.Lemma.lexeme

Definition at line 20 of file lemma.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/lemma.py

6.13 lmf.src.core.lexical_entry.LexicalEntry Class Reference

"Lexical Entry is a class representing a lexeme in a given language and is a container for managing the Form and Sense classes. A Lexical Entry instance can contain one to many different forms and can have from zero to many different senses." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_partOfSpeech

  Set grammatical category.
- def get_partOfSpeech

  Get grammatical category.
- def set_status

  Set lexical entry status.
- def get_status

  Get lexical entry status.
- def set_date

  Set lexical entry date.
- def get_date

  Get lexical entry date.
- def set_homonymNumber

  Set lexical entry homonym number.
- def get_homonymNumber

  Get lexical entry homonym number.
- def set_bibliography

  Set lexical entry bibliography.
- def get_bibliography

  Get lexical entry bibliography.
- def set_independentWord

  Set lexical entry independent word indication.
- def get_independentWord

  Get lexical entry independent word indication.
- def get_id

Get Unique IDentifier.

- def set_lexeme

  Set lexeme.
- def get_lexeme

  Get lexeme.
- def create_related_form

  Create a related form.
- def add_related_form

  Add a related form to the lexical entry.
- def create_and_add_related_form

  Create and add a related form to the lexical entry.
- def find_related_forms

  Find related lexemes.
- def get_related_forms

  Get all related forms maintained by the lexical entry.
- def get_form_representations

  Get all form representations maintained by the lemma.
- def set_variant_form

  Set variant form and type.
- def get_variant_forms

  Get all variant forms of specified type.
- def set_variant_comment

  Set variant comment and language.
- def set_tone

  Set tone.
- def get_tones

  Get all tones.
- def set_geographical_variant

  Set geographical variant.
- def set_phonetic_form

  Set phonetic form.
- def get_phonetic_forms

  Get all phonetic forms.
- def set_contextual_variation

  Set contextual variation.
- def get_contextual_variations

  Get all contextual variations.
- def set_spelling_variant

  Set spelling variant.
- def get_spelling_variants

  Get all spelling variants.
- def set_citation_form

  Set citation form.
- def get_citation_forms

  Get all citation forms.
- def set_dialect

  Set dialect.
- def set_transliteration

  Set transliteration.
- def get_transliterations

  Get all transliterations.

- def set_script_name

  Set script name.
- def create_sense

  Create a sense.
- def add_sense

  Add a sense to the lexical entry.
- def create_and_add_sense

  Create and add a sense to the lexical entry.
- def get_senses

  Get all senses maintained by the lexical entry.
- def get_last_sense

  Get the previously registered sense.
- def set_definition

  Set definition and language.
- def set_gloss

  Set gloss and language.
- def set_note

  Set note, type and language.
- def find_notes

  Find notes.
- def set_usage_note

  Set usage note and language.
- def set_encyclopedic_information

  Set encyclopedic information and language.
- def set_restriction

  Set restriction and language.
- def set_borrowed_word

  Set source language (in English).
- def get_borrowed_word

  Get source language (in English).
- def set_written_form

  Set loan word.
- def get_written_form

  Get loan word.
- def set_etymology

  Set etymology.
- def get_etymology

  Get etymology.
- def set_etymology_comment

  Set etymology comment and language.
- def get_etymology_comment

  Get etymology comment.
- def get_term_source_language

  Get language used for the etymology comment.
- def set_etymology_gloss

  Set etymology gloss.
- def get_etymology_gloss

  Get etymology gloss.
- def set_etymology_source

  Set etymology source.
- def get_etymology_source

Get etymology source.

- def set_scientific_name

  Set scientific_name.
- def get_scientific_name

  Get scientific name.
- def create_word_form

  Create a word form.
- def add_word_form

  Add a word form to the lexical entry.
- def get_word_forms

  Get all word forms maintained by the lexical entry.
- def set_paradigm

  Set paradigm.
- def find_paradigms

  Find paradigms.
- def set_paradigm_label

  Set paradigm label.
- def set_paradigm_form

  Set paradigm form.
- def set_morphology

  Set morphology.
- def get_paradigms

  Get all paradigms.
- def get_morphologies

  Get all morphologies.
- def create_example

  Create a context.
- def create_and_add_example

  Add an example to a new context and set its written form, language and script.
- def add_example

  Add an example to an existing context and set its written form, language and script.
- def set_example_comment

  Set comment of an existing example.
- def set_semantic_domain

  Set semantic domain and language.
- def get_semantic_domains

  Get all semantic domains.
- def set_translation

  Set translation and language.
- def set_audio

  Set audio resource.
- def is_subentry

  Check if this lexical entry is a subentry.
- def has_subentries

  Check if this lexical entry has subentries.
- def get_subentries

  Get subentries of this lexical entry.
- def get_main_entry

  If this lexical entry is a subentry, get its main entry.
- def create_and_add_component

  Create and add a component to the lexical entry.

- def get_components

  If this lexical entry is a multiword expression, get its components.
- def is_component

  Check if this lexical entry is a component.
- def get_speaker

  Get speaker.

Public Attributes

- homonymNumber
- status
- date
- partOfSpeech
- independentWord
- bibliography
- id

  UID is managed at the Lexicon level.
- sense

  Sense instances are owned by LexicalEntry There is zero to many Sense instances per LexicalEntry.
- lemma

  Lemma instance is owned by LexicalEntry There is one Lemma instance by LexicalEntry instance.
- related_form

  RelatedForm instances are owned by LexicalEntry There is zero to many RelatedForm instances per LexicalEntry.
- word_form

  WordForm instances are owned by LexicalEntry There is zero to many WordForm instances per LexicalEntry.
- stem

  Stem instances are owned by LexicalEntry There is zero to many Stem instances per LexicalEntry.
- list_of_components

  ListOfComponents instance is owned by LexicalEntry There is zero or one ListOfComponents instance per LexicalEntry.
- targets

6.13.1 Detailed Description

"Lexical Entry is a class representing a lexeme in a given language and is a container for managing the Form and Sense classes. A Lexical Entry instance can contain one to many different forms and can have from zero to many different senses." (LMF)

Definition at line 15 of file lexical_entry.py.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 def lmf.src.core.lexical_entry.LexicalEntry.__init__ (*self*, *id* = '0')

Constructor.

LexicalEntry instances are owned by Lexicon.

Parameters

id	Unique IDentifier. If not provided, default value is 0.

Returns

A LexicalEntry instance.

Definition at line 18 of file lexical_entry.py.

6.13.2.2 def lmf.src.core.lexical_entry.LexicalEntry.__del__ (*self*)

Destructor.

Release Sense, Lemma, RelatedForm, WordForm, Stem, ListOfComponents instances.

Definition at line 56 of file lexical_entry.py.

6.13.3 Member Function Documentation

6.13.3.1 def lmf.src.core.lexical_entry.LexicalEntry.add_example (*self, written_form, language* = `None`, *script_name* = `None`)

Add an example to an existing context and set its written form, language and script.

Attributes 'writtenForm', 'language' and 'scriptName' are owned by TextRepresentation, which is owned by Context, itself owend by Sense.

Parameters

written_form	The written form to set.
language	Language used for the written form.
script_name	The name of the script used to write the example, e.g. devanagari.

Returns

LexicalEntry instance.

Definition at line 967 of file lexical_entry.py.

6.13.3.2 def lmf.src.core.lexical_entry.LexicalEntry.add_related_form (*self, related_form*)

Add a related form to the lexical entry.

Parameters

related_form	The RelatedForm instance to add to the lexical entry.

Returns

LexicalEntry instance.

Definition at line 209 of file lexical_entry.py.

6.13.3.3 def lmf.src.core.lexical_entry.LexicalEntry.add_sense (*self, sense*)

Add a sense to the lexical entry.

Parameters

sense	The Sense instance to add to the lexical entry.

Returns

LexicalEntry instance.

Definition at line 465 of file lexical_entry.py.

6.13.3.4 def lmf.src.core.lexical_entry.LexicalEntry.add_word_form (*self, word_form*)

Add a word form to the lexical entry.

Parameters

word_form	The WordForm instance to add to the lexical entry.

Returns

LexicalEntry instance.

Definition at line 800 of file lexical_entry.py.

6.13.3.5 def lmf.src.core.lexical_entry.LexicalEntry.create_and_add_component (*self, position, lexeme*)

Create and add a component to the lexical entry.

Parameters

position	The position of the component in the multiword expression.
lexeme	Related lexeme.

Returns

LexicalEntry instance.

Definition at line 1102 of file lexical_entry.py.

6.13.3.6 def lmf.src.core.lexical_entry.LexicalEntry.create_and_add_example (*self, written_form, language* = `None`, *script_name* = `None`)

Add an example to a new context and set its written form, language and script.

Attributes 'writtenForm', 'language' and 'scriptName' are owned by TextRepresentation, which is owned by Context, itself owend by Sense.

Parameters

written_form	The written form to set.
language	Language used for the written form.
script_name	The name of the script used to write the example, e.g. devanagari.

Returns

LexicalEntry instance.

Definition at line 950 of file lexical_entry.py.

6.13.3.7 def lmf.src.core.lexical_entry.LexicalEntry.create_and_add_related_form (*self, lexeme, semantic_relation*)

Create and add a related form to the lexical entry.

Parameters

lexeme	Related lexeme.
semantic_↩ relation	The semantic relation existing between this lexical entry and the related lexeme to create.

Returns

LexicalEntry instance.

Definition at line 217 of file lexical_entry.py.

6.13.3.8 def lmf.src.core.lexical_entry.LexicalEntry.create_and_add_sense (*self, sense_number*)

Create and add a sense to the lexical entry.

Parameters

sense_number	Number of the sense to add.

Returns

LexicalEntry instance.

Definition at line 473 of file lexical_entry.py.

6.13.3.9 def lmf.src.core.lexical_entry.LexicalEntry.create_example (*self, reference* = `None`)

Create a context.

Attribute 'targets' is owned by Context, itself owend by Sense.

Parameters

reference	The example reference to set. If not provided, default value is None.

Returns

LexicalEntry instance.

Definition at line 935 of file lexical_entry.py.

6.13.3.10 def lmf.src.core.lexical_entry.LexicalEntry.create_related_form (*self, lexeme, semantic_relation*)

Create a related form.

Parameters

lexeme	Related lexeme.
semantic_↩ relation	The semantic relation existing between this lexical entry and the related lexeme to create.

Returns

RelatedForm instance.

Definition at line 201 of file lexical_entry.py.

6.13.3.11 def lmf.src.core.lexical_entry.LexicalEntry.create_sense (*self, id* = `0`)

Create a sense.

Parameters

id	Identifier.

Returns

Sense instance.

Definition at line 458 of file lexical_entry.py.

6.13.3.12 def lmf.src.core.lexical_entry.LexicalEntry.create_word_form (*self*)

Create a word form.

Returns

WordForm instance.

Definition at line 794 of file lexical_entry.py.

6.13.3.13 def lmf.src.core.lexical_entry.LexicalEntry.find_notes (*self, type*)

Find notes.

This attribute is owned by Statement, which owned by Definition, itself owned by Sense.

Parameters

type	Type of the note to consider to retrieve the note.

Returns

A Python list of found Statement attributes 'notes'.

Definition at line 544 of file lexical_entry.py.

6.13.3.14 def lmf.src.core.lexical_entry.LexicalEntry.find_paradigms (*self, script_name* = `None`, *person* = `None`, *anymacy* = `None`, *grammatical_number* = `None`, *clusivity* = `None`)

Find paradigms.

Attribute 'scriptName' is owned by FormRepresentation, wich is owned by WordForm. Attributes 'person', 'anymacy', 'grammaticalNumber' and 'clusivity' are owned by WordForm. Attribute 'writtenForm' to retrieve is owned by Form↩ Representation, wich is owned by WordForm.

Parameters

script_name	If this argument is given, get paradigm written form only if written using this script.
person	Person, e.g. first person.
anymacy	Anymacy, e.g. animate or inanimate.
grammatical_↩ number	Grammatical number, e.g. singular or plural.
clusivity	Clusivity, e.g. inclusive or exclusive.

Returns

A Python list of FormRepresentation attributes 'writtenForm'.

Definition at line 848 of file lexical_entry.py.

6.13.3.15 def lmf.src.core.lexical_entry.LexicalEntry.find_related_forms (*self, semantic_relation*)

Find related lexemes.

This attribute is owned by RelatedForm.

Parameters

semantic_↩ relation	The semantic relation to consider to retrieve the related form.

Returns

A Python list of found RelatedForm attributes 'targets'.

Definition at line 230 of file lexical_entry.py.

6.13.3.16 def lmf.src.core.lexical_entry.LexicalEntry.get_bibliography (*self*)

Get lexical entry bibliography.

Returns

LexicalEntry attribute 'bibliography'.

Definition at line 150 of file lexical_entry.py.

6.13.3.17 def lmf.src.core.lexical_entry.LexicalEntry.get_borrowed_word (*self*)

Get source language (in English).

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'borrowedWord'.

Definition at line 618 of file lexical_entry.py.

6.13.3.18 def lmf.src.core.lexical_entry.LexicalEntry.get_citation_forms (*self, script_name* = `None`)

Get all citation forms.

Attribute 'citationForm' is owned by FormRepresentation, which is owned by Lemma.

Parameters

script_name	If provided, get only citation forms that are written using this script.

Returns

A Python list of FormRepresentation attributes 'citationForm' if any.

Definition at line 405 of file lexical_entry.py.

6.13.3.19 def lmf.src.core.lexical_entry.LexicalEntry.get_components (*self*)

If this lexical entry is a multiword expression, get its components.

Returns

A list of components if any, an empty list otherwise.

Definition at line 1113 of file lexical_entry.py.

6.13.3.20 def lmf.src.core.lexical_entry.LexicalEntry.get_contextual_variations (*self*)

Get all contextual variations.

Attribute 'contextualVariation' is owned by FormRepresentation, which is owned by Lemma.

Returns

A Python list of FormRepresentation attributes 'contextualVariation' if any.

Definition at line 364 of file lexical_entry.py.

6.13.3.21 def lmf.src.core.lexical_entry.LexicalEntry.get_date (*self*)

Get lexical entry date.

Returns

LexicalEntry attribute 'date'.

Definition at line 122 of file lexical_entry.py.

6.13.3.22 def lmf.src.core.lexical_entry.LexicalEntry.get_etymology (*self*)

Get etymology.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

The first found Statement attribute 'etymology'.

Definition at line 670 of file lexical_entry.py.

6.13.3.23 def lmf.src.core.lexical_entry.LexicalEntry.get_etymology_comment (*self, term_source_language* = None)

Get etymology comment.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

term_source_language	The language of the etymology comment to retrieve.

Returns

The first found Statement attribute 'etymologyComment'.

Definition at line 695 of file lexical_entry.py.

6.13.3.24 def lmf.src.core.lexical_entry.LexicalEntry.get_etymology_gloss (*self*)

Get etymology gloss.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'etymologyGloss'.

Definition at line 731 of file lexical_entry.py.

6.13.3.25 def lmf.src.core.lexical_entry.LexicalEntry.get_etymology_source (*self*)

Get etymology source.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'etymologySource'.

Definition at line 757 of file lexical_entry.py.

6.13.3.26 def lmf.src.core.lexical_entry.LexicalEntry.get_form_representations (*self*)

Get all form representations maintained by the lemma.

Attribute 'form_representation' is owned by Lemma.

Returns

Lemma attribute 'form_representation' if any.

Definition at line 256 of file lexical_entry.py.

6.13.3.27 def lmf.src.core.lexical_entry.LexicalEntry.get_homonymNumber (*self*)

Get lexical entry homonym number.

Returns

LexicalEntry attribute 'homonymNumber'.

Definition at line 136 of file lexical_entry.py.

6.13.3.28 def lmf.src.core.lexical_entry.LexicalEntry.get_id (*self*)

Get Unique IDentifier.

Returns

LexicalEntry attribute 'id' followed by the homonym number.

Definition at line 172 of file lexical_entry.py.

6.13.3.29 def lmf.src.core.lexical_entry.LexicalEntry.get_independentWord (*self*)

Get lexical entry independent word indication.

Returns

LexicalEntry attribute 'independentWord'.

Definition at line 166 of file lexical_entry.py.

6.13.3.30 def lmf.src.core.lexical_entry.LexicalEntry.get_last_sense (*self*)

Get the previously registered sense.

Returns

The last element of LexicalEntry attribute 'sense'.

Definition at line 488 of file lexical_entry.py.

6.13.3.31 def lmf.src.core.lexical_entry.LexicalEntry.get_lexeme (*self*)

Get lexeme.

Attribute 'lexeme' is owned by Lemma.

Returns

Lemma attribute 'lexeme' if any.

Definition at line 193 of file lexical_entry.py.

6.13.3.32 def lmf.src.core.lexical_entry.LexicalEntry.get_main_entry (*self*)

If this lexical entry is a subentry, get its main entry.

Returns

A LexicalEntry if it exists, 'None' otherwise.

Definition at line 1094 of file lexical_entry.py.

6.13.3.33 def lmf.src.core.lexical_entry.LexicalEntry.get_morphologies (*self*)

Get all morphologies.

This attribute is owned by Paradigm, which is owned by Sense.

Returns

A Python list of Paradigm attributes 'morphology'.

Definition at line 923 of file lexical_entry.py.

6.13.3.34 def lmf.src.core.lexical_entry.LexicalEntry.get_paradigms (*self*)

Get all paradigms.

This attribute is owned by Sense.

Returns

Sense attribute 'paradigm'.

Definition at line 913 of file lexical_entry.py.

6.13.3.35 def lmf.src.core.lexical_entry.LexicalEntry.get_partOfSpeech (*self*)

Get grammatical category.

Returns

LexicalEntry attribute 'partOfSpeech'.

Definition at line 94 of file lexical_entry.py.

6.13.3.36 def lmf.src.core.lexical_entry.LexicalEntry.get_phonetic_forms (*self, script_name* = `None`)

Get all phonetic forms.

Attribute 'phoneticForm' is owned by FormRepresentation, which is owned by Lemma.

Parameters

script_name	If provided, get only phonetic forms that are written using this script.

Returns

A Python list of FormRepresentation attributes 'phoneticForm' if any.

Definition at line 343 of file lexical_entry.py.

6.13.3.37 def lmf.src.core.lexical_entry.LexicalEntry.get_related_forms (*self, semantic_relation* = `None`)

Get all related forms maintained by the lexical entry.

Parameters

semantic_↩ relation	The semantic relation to consider to retrieve the related forms.

Returns

A Python set of related forms.

Definition at line 242 of file lexical_entry.py.

6.13.3.38 def lmf.src.core.lexical_entry.LexicalEntry.get_scientific_name (*self*)

Get scientific name.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'scientificName'.

Definition at line 783 of file lexical_entry.py.

6.13.3.39 def lmf.src.core.lexical_entry.LexicalEntry.get_semantic_domains (*self, language* = None)

Get all semantic domains.

This attribute is owned by SubjectField, which is owned by Sense.

Parameters

language	If this argument is given, get only semantic domains that are described using this language.

Returns

A Python list of filtered SubjectField attributes 'semanticDomain'.

Definition at line 1015 of file lexical_entry.py.

6.13.3.40 def lmf.src.core.lexical_entry.LexicalEntry.get_senses (*self*)

Get all senses maintained by the lexical entry.

Returns

LexicalEntry attribute 'sense'.

Definition at line 482 of file lexical_entry.py.

6.13.3.41 def lmf.src.core.lexical_entry.LexicalEntry.get_speaker (*self*)

Get speaker.

Returns

LexicalEntry private attribute '__speaker'.

Definition at line 1131 of file lexical_entry.py.

6.13.3.42 def lmf.src.core.lexical_entry.LexicalEntry.get_spelling_variants (*self*)

Get all spelling variants.

Attribute 'spellingVariant' is owned by FormRepresentation, which is owned by Lemma.

Returns

A Python list of FormRepresentation attributes 'spellingVariant' if any.

Definition at line 384 of file lexical_entry.py.

6.13.3.43 def lmf.src.core.lexical_entry.LexicalEntry.get_status (*self*)

Get lexical entry status.

Returns

LexicalEntry attribute 'status'.

Definition at line 108 of file lexical_entry.py.

6.13.3.44 def lmf.src.core.lexical_entry.LexicalEntry.get_subentries (*self*)

Get subentries of this lexical entry.

Returns

A Python list of LexicalEntry.

Definition at line 1084 of file lexical_entry.py.

6.13.3.45 def lmf.src.core.lexical_entry.LexicalEntry.get_term_source_language (*self*)

Get language used for the etymology comment.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'termSourceLanguage'.

Definition at line 705 of file lexical_entry.py.

6.13.3.46 def lmf.src.core.lexical_entry.LexicalEntry.get_tones (*self*)

Get all tones.

Attribute 'tone' is owned by FormRepresentation, which is owned by Lemma.

Returns

A Python list of FormRepresentation attributes 'tone' if any.

Definition at line 310 of file lexical_entry.py.

6.13.3.47 def lmf.src.core.lexical_entry.LexicalEntry.get_transliterations (*self*)

Get all transliterations.

Attribute 'transliteration' is owned by FormRepresentation, which is owned by Lemma.

Returns

A Python list of FormRepresentation attributes 'transliteration' if any.

Definition at line 438 of file lexical_entry.py.

6.13.3.48 def lmf.src.core.lexical_entry.LexicalEntry.get_variant_forms (*self, type =* "unspecified")

Get all variant forms of specified type.

Attribute 'variantForm' is owned by FormRepresentation, which is owned by Lemma.

Returns

A Python list of FormRepresentation attributes 'variantForm' if type matches.

Definition at line 277 of file lexical_entry.py.

6.13.3.49 def lmf.src.core.lexical_entry.LexicalEntry.get_word_forms (*self*)

Get all word forms maintained by the lexical entry.

Returns

A Python list of word forms.

Definition at line 808 of file lexical_entry.py.

6.13.3.50 def lmf.src.core.lexical_entry.LexicalEntry.get_written_form (*self*)

Get loan word.

This attribute is owned by Statement, which is owned by Definition, itself owned by Sense.

Returns

Statement attribute 'writtenForm'.

Definition at line 644 of file lexical_entry.py.

6.13.3.51 def lmf.src.core.lexical_entry.LexicalEntry.has_subentries (*self*)

Check if this lexical entry has subentries.

Returns

'True' if it has subentries, 'False' otherwise.

Definition at line 1074 of file lexical_entry.py.

6.13.3.52 def lmf.src.core.lexical_entry.LexicalEntry.is_component (*self*)

Check if this lexical entry is a component.

Returns

'True' if it is a component, 'False' otherwise.

Definition at line 1121 of file lexical_entry.py.

6.13.3.53 def lmf.src.core.lexical_entry.LexicalEntry.is_subentry (*self*)

Check if this lexical entry is a subentry.

Returns

'True' if it is a subentry, 'False' otherwise.

Definition at line 1064 of file lexical_entry.py.

6.13.3.54 def lmf.src.core.lexical_entry.LexicalEntry.set_audio (*self*, *media_type* = "audio", *file_name* = None, *author* = None, *quality* = None, *start_position* = "T00:00:00", *duration* = None, *external_reference* = None, *audio_file_format* = None)

Set audio resource.

Attributes 'mediaType', 'fileName', 'author', 'quality', 'startPosition', 'durationOfEffectiveSpeech', 'external↩ Reference', 'audioFileFormat' are owned by Material/Audio, which is owned by FormRepresentation, itself owend by Lemma.

Parameters

media_type	The media type to set.
file_name	Name of the audio file.
author	Author of the recording.
quality	Quality of the recording, in range 'quality_range' defined in 'common/range.py'.
start_position	Start position of the form in the recording, in format 'Thh:mm:ss,msms', e.g. "T00:05:00".
duration	Duration of the effcetive speech, in format 'PThhHmmMssS', e.g. "PT00:05:00".
external_↩ reference	Reference of the audio file, if not directly provided.
audio_file_↩ format	Format of the audio file, e.g. "wav".

Returns

LexicalEntry instance.

Definition at line 1045 of file lexical_entry.py.

6.13.3.55 def lmf.src.core.lexical_entry.LexicalEntry.set_bibliography (*self*, *bibliography*)

Set lexical entry bibliography.

Parameters

bibliography	The bibliography to set.

Returns

LexicalEntry instance.

Definition at line 142 of file lexical_entry.py.

6.13.3.56 def lmf.src.core.lexical_entry.LexicalEntry.set_borrowed_word (*self*, *borrowed_word*)

Set source language (in English).

Attribute 'borrowedWord' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

borrowed_word	Source language.

Returns

LexicalEntry instance.

Definition at line 603 of file lexical_entry.py.

6.13.3.57 def lmf.src.core.lexical_entry.LexicalEntry.set_citation_form (*self, citation_form, script_name* = None)

Set citation form.

Attribute 'citationForm' is owned by FormRepresentation, which is owned by Lemma.

Parameters

citation_form	The citation form to set.
script_name	The name of the script used to write the citation form, e.g. devanagari.

Returns

LexicalEntry instance.

Definition at line 392 of file lexical_entry.py.

6.13.3.58 def lmf.src.core.lexical_entry.LexicalEntry.set_contextual_variation (*self, contextual_variation*)

Set contextual variation.

Attribute 'contextualVariation' is owned by FormRepresentation, which is owned by Lemma.

Parameters

contextual_↩ variation	The contextual variation to set.

Returns

LexicalEntry instance.

Definition at line 352 of file lexical_entry.py.

6.13.3.59 def lmf.src.core.lexical_entry.LexicalEntry.set_date (*self, date*)

Set lexical entry date.

Parameters

status	The date to set.

Returns

LexicalEntry instance.

Definition at line 114 of file lexical_entry.py.

6.13.3.60 def lmf.src.core.lexical_entry.LexicalEntry.set_definition (*self, definition, language* = None)

Set definition and language.

Attributes 'definition' and 'language' are owned by Definition, which is owned by Sense.

Parameters

definition	Definition.
language	Language of definition.

Returns

LexicalEntry instance.

Definition at line 495 of file lexical_entry.py.

6.13.3.61 def lmf.src.core.lexical_entry.LexicalEntry.set_dialect (*self, dialect*)

Set dialect.

Attribute 'dialect' is owned by FormRepresentation, which is owned by Lemma.

Parameters

dialect	The dialect to set.

Returns

LexicalEntry instance.

Definition at line 414 of file lexical_entry.py.

6.13.3.62 def lmf.src.core.lexical_entry.LexicalEntry.set_encyclopedic_information (*self, encyclopedic_information, language* = `None`)

Set encyclopedic information and language.

Attributes 'encyclopedicInformation' and 'language' are owned by Statement, which owned by Definition, itself owned by Sense.

Parameters

encyclopedic_information	Encyclopedic information to set.
language	Language of the encyclopedic information.

Returns

LexicalEntry instance.

Definition at line 571 of file lexical_entry.py.

6.13.3.63 def lmf.src.core.lexical_entry.LexicalEntry.set_etymology (*self, etymology*)

Set etymology.

Attribute 'etymology' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

etymology	Etymology.

Returns

LexicalEntry instance.

Definition at line 655 of file lexical_entry.py.

6.13.3.64 def lmf.src.core.lexical_entry.LexicalEntry.set_etymology_comment (*self, etymology_comment, term_source_language* = `None`)

Set etymology comment and language.

Attributes 'etymologyComment' and 'termSourceLanguage' are owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

etymology_↩ comment	Etymology comment.
term_source_↩ language	Language of the comment.

Returns

LexicalEntry instance.

Definition at line 679 of file lexical_entry.py.

6.13.3.65 def lmf.src.core.lexical_entry.LexicalEntry.set_etymology_gloss (*self, etymology_gloss*)

Set etymology gloss.

Attribute 'etymologyGloss' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

etymology_gloss	Etymology gloss.

Returns

LexicalEntry instance.

Definition at line 716 of file lexical_entry.py.

6.13.3.66 def lmf.src.core.lexical_entry.LexicalEntry.set_etymology_source (*self, etymology_source*)

Set etymology source.

Attribute 'etymologySource' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

etymology_↩ source	Etymology source.

Returns

LexicalEntry instance.

Definition at line 742 of file lexical_entry.py.

6.13.3.67 def lmf.src.core.lexical_entry.LexicalEntry.set_example_comment (*self, comment*)

Set comment of an existing example.

Attribute 'comment' is owned by TextRepresentation, which is owned by Context, itself owend by Sense.

Parameters

comment	The comment to set.

Returns

LexicalEntry instance.

Definition at line 984 of file lexical_entry.py.

6.13.3.68 def lmf.src.core.lexical_entry.LexicalEntry.set_geographical_variant (*self, geographical_variant*)

Set geographical variant.

Attribute 'geographicalVariant' is owned by FormRepresentation, which is owned by Lemma.

Parameters

geographical_↩ variant	The geographical variant to set.

Returns

LexicalEntry instance.

Definition at line 318 of file lexical_entry.py.

6.13.3.69 def lmf.src.core.lexical_entry.LexicalEntry.set_gloss (*self, gloss, language* = `None`)

Set gloss and language.

Attributes 'gloss' and 'language' are owned by Definition, which is owned by Sense.

Parameters

gloss	Gloss.
language	Language of gloss.

Returns

LexicalEntry instance.

Definition at line 511 of file lexical_entry.py.

6.13.3.70 def lmf.src.core.lexical_entry.LexicalEntry.set_homonymNumber (*self, homonym_number*)

Set lexical entry homonym number.

Parameters

homonym_↩ number	The homonym number to set.

Returns

LexicalEntry instance.

Definition at line 128 of file lexical_entry.py.

6.13.3.71 def lmf.src.core.lexical_entry.LexicalEntry.set_independentWord (*self, independent_word*)

Set lexical entry independent word indication.

Parameters

independent_↩ word	The independent word indication to set.

Returns

LexicalEntry instance.

Definition at line 156 of file lexical_entry.py.

6.13.3.72 def lmf.src.core.lexical_entry.LexicalEntry.set_lexeme (*self, lexeme*)

Set lexeme.

Attribute 'lexeme' is owned by Lemma.

Parameters

lexeme	The lexeme to set.

Returns

LexicalEntry instance.

Definition at line 181 of file lexical_entry.py.

6.13.3.73 def lmf.src.core.lexical_entry.LexicalEntry.set_morphology (*self, morphology*)

Set morphology.

Attribute 'morphology' is owned by Paradigm, which is owned by Sense.

Parameters

morphology	Morphology.

Returns

LexicalEntry instance.

Definition at line 898 of file lexical_entry.py.

6.13.3.74 def lmf.src.core.lexical_entry.LexicalEntry.set_note (*self, note, type* = None, *language* = None)

Set note, type and language.

Attributes 'note', 'noteType' and 'language' are owned by Statement, which owned by Definition, itself owned by Sense.

Parameters

note	Note to set.
type	Type of the note.
language	Language of the note.

Returns

LexicalEntry instance.

Definition at line 527 of file lexical_entry.py.

6.13.3.75 def lmf.src.core.lexical_entry.LexicalEntry.set_paradigm (*self, written_form, script_name* = None, *person* = None, *anymacy* = None, *grammatical_number* = None, *clusivity* = None)

Set paradigm.

Attributes 'writtenForm' and 'scriptName' are owned by FormRepresentation, wich is owned by WordForm. Attributes 'person', 'anymacy', 'grammaticalNumber' and 'clusivity' are owned by WordForm.

Parameters

written_form	The paradigm to set.
script_name	Script used for the written form.
person	Person, e.g. first person.
anymacy	Anymacy, e.g. animate or inanimate.
grammatical_number	Grammatical number, e.g. singular or plural.
clusivity	Clusivity, e.g. inclusive or exclusive.

Returns

LexicalEntry instance.

Definition at line 814 of file lexical_entry.py.

6.13.3.76 def lmf.src.core.lexical_entry.LexicalEntry.set_paradigm_form (*self, paradigm, language* = None)

Set paradigm form.

Attribute 'paradigm' is owned by Paradigm, which is owned by Sense.

Parameters

paradigm	Paradigm form.
language	Language of the paradigm form.

Returns

LexicalEntry instance.

Definition at line 882 of file lexical_entry.py.

6.13.3.77 def lmf.src.core.lexical_entry.LexicalEntry.set_paradigm_label (*self, paradigm_label*)

Set paradigm label.

Attribute 'paradigmLabel' is owned by Paradigm, which is owned by Sense.

Parameters

paradigm_label	Paradigm label.

Returns

LexicalEntry instance.

Definition at line 867 of file lexical_entry.py.

6.13.3.78 def lmf.src.core.lexical_entry.LexicalEntry.set_partOfSpeech (*self, part_of_speech, range* = `partOfSpeech_range`, *mapping* = `ps_partOfSpeech`)

Set grammatical category.

Parameters

part_of_speech	The grammatical category to set.
range	A Python set giving all possible values of part of speech LMF attribute.
mapping	A Python dictionary giving the mapping between MDF and LMF values.

Returns

LexicalEntry instance.

Definition at line 79 of file lexical_entry.py.

6.13.3.79 def lmf.src.core.lexical_entry.LexicalEntry.set_phonetic_form (*self, phonetic_form, script_name* = `None`)

Set phonetic form.

Attribute 'phoneticForm' is owned by FormRepresentation, which is owned by Lemma.

Parameters

phonetic_form	The phonetic form to set.
script_name	The name of the script used to write the phonetic form, e.g. pinyin.

Returns

LexicalEntry instance.

Definition at line 330 of file lexical_entry.py.

6.13.3.80 def lmf.src.core.lexical_entry.LexicalEntry.set_restriction (*self, restriction, language* = `None`)

Set restriction and language.

Attributes 'restriction' and 'language' are owned by Statement, which owned by Definition, itself owned by Sense.

Parameters

restriction	Restriction to set.
language	Language of the restriction.

Returns

LexicalEntry instance.

Definition at line 587 of file lexical_entry.py.

6.13.3.81 def lmf.src.core.lexical_entry.LexicalEntry.set_scientific_name (*self, scientific_name*)

Set scientific_name.

Attribute 'scientificName' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

scientific_name	Scientific name.

Returns

LexicalEntry instance.

Definition at line 768 of file lexical_entry.py.

6.13.3.82 def lmf.src.core.lexical_entry.LexicalEntry.set_script_name (*self, script_name*)

Set script name.

Attribute 'scriptName' is owned by FormRepresentation, which is owned by Lemma.

Parameters

script_name	The script name to set.

Returns

LexicalEntry instance.

Definition at line 446 of file lexical_entry.py.

6.13.3.83 def lmf.src.core.lexical_entry.LexicalEntry.set_semantic_domain (*self, semantic_domain, language* = `None`)

Set semantic domain and language.

Attributes 'semanticDomain' and 'language' are owned by SubjectField, which is owned by Sense.

Parameters

semantic_↩ domain	The semantic domain to set.
language	Language used to describe the semantic domain.

Returns

LexicalEntry instance.

Definition at line 999 of file lexical_entry.py.

6.13.3.84 def lmf.src.core.lexical_entry.LexicalEntry.set_spelling_variant (*self, spelling_variant*)

Set spelling variant.

Attribute 'spellingVariant' is owned by FormRepresentation, which is owned by Lemma.

Parameters

spelling_variant	The spelling variant to set.

Returns

LexicalEntry instance.

Definition at line 372 of file lexical_entry.py.

6.13.3.85 def lmf.src.core.lexical_entry.LexicalEntry.set_status (*self, status*)

Set lexical entry status.

Parameters

status	The status to set.

Returns

LexicalEntry instance.

Definition at line 100 of file lexical_entry.py.

6.13.3.86 def lmf.src.core.lexical_entry.LexicalEntry.set_tone (*self, tone*)

Set tone.

Attribute 'tone' is owned by FormRepresentation, which is owned by Lemma.

Parameters

tone	The tone to set.

Returns

LexicalEntry instance.

Definition at line 298 of file lexical_entry.py.

6.13.3.87 def lmf.src.core.lexical_entry.LexicalEntry.set_translation (*self, translation, language* = `None`)

Set translation and language.

Attributes 'translation' and 'language' are owned by Equivalent, which is owned by Sense.

Parameters

translation	The translation to set.
language	Language used for the translation.

Returns

LexicalEntry instance.

Definition at line 1029 of file lexical_entry.py.

6.13.3.88 def lmf.src.core.lexical_entry.LexicalEntry.set_transliteration (*self, transliteration*)

Set transliteration.

Attribute 'transliteration' is owned by FormRepresentation, which is owned by Lemma.

Parameters

transliteration	The transliteration to set.

Returns

LexicalEntry instance.

Definition at line 426 of file lexical_entry.py.

6.13.3.89 def lmf.src.core.lexical_entry.LexicalEntry.set_usage_note (*self, usage_note, language* = `None`)

Set usage note and language.

Attributes 'usageNote' and 'language' are owned by Statement, which owned by Definition, itself owned by Sense.

Parameters

usage_note	Usage note to set.
language	Language of the usage note.

Returns

LexicalEntry instance.

Definition at line 555 of file lexical_entry.py.

6.13.3.90 def lmf.src.core.lexical_entry.LexicalEntry.set_variant_comment (*self, comment, language* = `None`)

Set variant comment and language.

Attributes 'comment' and 'language' are owned by FormRepresentation, which is owned by Lemma.

Parameters

comment	Variant comment.
language	Language of comment.

Returns

LexicalEntry instance.

Definition at line 285 of file lexical_entry.py.

6.13.3.91 def lmf.src.core.lexical_entry.LexicalEntry.set_variant_form (*self, variant_form, type =* "unspecified")

Set variant form and type.

Attributes 'variantForm' and 'type' are owned by FormRepresentation, which is owned by Lemma.

Parameters

variant_form	Variant form.
type	Type of variant, in range 'type_variant_range' defined in 'common/range.py'.

Returns

LexicalEntry instance.

Definition at line 264 of file lexical_entry.py.

6.13.3.92 def lmf.src.core.lexical_entry.LexicalEntry.set_written_form (*self, written_form*)

Set loan word.

Attribute 'writtenForm' is owned by Statement, which is owned by Definition, itself owned by Sense.

Parameters

written_form	Loan word.

Returns

LexicalEntry instance.

Definition at line 629 of file lexical_entry.py.

6.13.4 Member Data Documentation

6.13.4.1 lmf.src.core.lexical_entry.LexicalEntry.bibliography

Definition at line 29 of file lexical_entry.py.

6.13.4.2 lmf.src.core.lexical_entry.LexicalEntry.date

Definition at line 26 of file lexical_entry.py.

6.13.4.3 lmf.src.core.lexical_entry.LexicalEntry.homonymNumber

Definition at line 24 of file lexical_entry.py.

6.13.4.4 lmf.src.core.lexical_entry.LexicalEntry.id

UID is managed at the Lexicon level.

Definition at line 31 of file lexical_entry.py.

6.13.4.5 lmf.src.core.lexical_entry.LexicalEntry.independentWord

Definition at line 28 of file lexical_entry.py.

6.13.4.6 lmf.src.core.lexical_entry.LexicalEntry.lemma

Lemma instance is owned by LexicalEntry There is one Lemma instance by LexicalEntry instance.

Definition at line 37 of file lexical_entry.py.

6.13.4.7 lmf.src.core.lexical_entry.LexicalEntry.list_of_components

ListOfComponents instance is owned by LexicalEntry There is zero or one ListOfComponents instance per LexicalEntry.

Definition at line 49 of file lexical_entry.py.

6.13.4.8 lmf.src.core.lexical_entry.LexicalEntry.partOfSpeech

Definition at line 27 of file lexical_entry.py.

6.13.4.9 lmf.src.core.lexical_entry.LexicalEntry.related_form

RelatedForm instances are owned by LexicalEntry There is zero to many RelatedForm instances per LexicalEntry.

Definition at line 40 of file lexical_entry.py.

6.13.4.10 lmf.src.core.lexical_entry.LexicalEntry.sense

Sense instances are owned by LexicalEntry There is zero to many Sense instances per LexicalEntry.

Definition at line 34 of file lexical_entry.py.

6.13.4.11 lmf.src.core.lexical_entry.LexicalEntry.status

Definition at line 25 of file lexical_entry.py.

6.13.4.12 lmf.src.core.lexical_entry.LexicalEntry.stem

Stem instances are owned by LexicalEntry There is zero to many Stem instances per LexicalEntry.

Definition at line 46 of file lexical_entry.py.

6.13.4.13 lmf.src.core.lexical_entry.LexicalEntry.targets

Definition at line 51 of file lexical_entry.py.

6.13.4.14 lmf.src.core.lexical_entry.LexicalEntry.word_form

WordForm instances are owned by LexicalEntry There is zero to many WordForm instances per LexicalEntry.

Definition at line 43 of file lexical_entry.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_entry.py

6.14 lmf.src.core.lexical_resource.LexicalResource Class Reference

"Lexical Resource is a class representing the entire resource and is a container for one or more lexicons. There is only one Lexical Resource instance." (LMF)

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def get_lexicons

  Get all lexicons maintained by the lexical resource.
- def add_lexicon

  Add a lexicon to the lexical resource.
- def remove_lexicon

  Remove a lexicon from the lexical resource.
- def get_lexicon
- def set_dtdVersion

  Set DTD version.
- def get_dtdVersion

  Get DTD version.
- def set_language_code

  Set language code.
- def get_language_code

  Get language code.
- def set_version

  Set version.
- def get_version

  Get version.
- def set_license

  Set license.
- def get_license

  Get license.
- def set_character_encoding

  Set character encoding.
- def get_character_encoding

  Get character encoding.
- def set_date_coding

  Set date coding.
- def get_date_coding

  Get date coding.
- def set_project_name

  Set project name.
- def get_project_name

  Get project name.
- def set_creation_date

  Set creation date.
- def get_creation_date

  Get creation date.
- def set_last_update

  Set last update.
- def get_last_update

  Get last update.
- def set_author

  Set author.

- def get_author

  Get author.
- def set_description

  Set description.
- def get_description

  Get description.
- def get_bibliographic_citation

  Get bibliographic citation.

Public Attributes

- dtdVersion
- global_information

  GlobalInformation instance is owned by LexicalResource There is one GlobalInformation for one LexicalResource.
- lexicon

  Lexicon instances are owned by LexicalResource There is one or more Lexicon instances for one unique LexicalResource.
- speaker

  Speaker instances are owned by LexicalResource There is zero to many Speaker instances for one unique LexicalResource.

6.14.1 Detailed Description

"Lexical Resource is a class representing the entire resource and is a container for one or more lexicons. There is only one Lexical Resource instance." (LMF)

Definition at line 8 of file lexical_resource.py.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 def lmf.src.core.lexical_resource.LexicalResource.__init__ (*self, dtd_version* = 16)

Constructor.

Returns

A LexicalResource instance.

Definition at line 11 of file lexical_resource.py.

6.14.2.2 def lmf.src.core.lexical_resource.LexicalResource.__del__ (*self*)

Destructor.

Release GlobalInformation, Lexicon, Speaker instances.

Definition at line 26 of file lexical_resource.py.

6.14.3 Member Function Documentation

6.14.3.1 def lmf.src.core.lexical_resource.LexicalResource.add_lexicon (*self, lexicon*)

Add a lexicon to the lexical resource.

Parameters

lexicon	A Lexicon instance to add to the Lexical Resource.

Returns

Lexical Resource instance.

Definition at line 45 of file lexical_resource.py.

6.14.3.2 def lmf.src.core.lexical_resource.LexicalResource.get_author (*self*)

Get author.

Attribute 'author' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'author'.

Definition at line 221 of file lexical_resource.py.

6.14.3.3 def lmf.src.core.lexical_resource.LexicalResource.get_bibliographic_citation (*self*)

Get bibliographic citation.

Attribute 'bibliographicCitation' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'bibliographicCitation'.

Definition at line 244 of file lexical_resource.py.

6.14.3.4 def lmf.src.core.lexical_resource.LexicalResource.get_character_encoding (*self*)

Get character encoding.

Attribute 'characterEncoding' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'characterEncoding'.

Definition at line 141 of file lexical_resource.py.

6.14.3.5 def lmf.src.core.lexical_resource.LexicalResource.get_creation_date (*self*)

Get creation date.

Attribute 'creationDate' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'creationdDate'.

Definition at line 189 of file lexical_resource.py.

6.14.3.6 def lmf.src.core.lexical_resource.LexicalResource.get_date_coding (*self*)

Get date coding.

Attribute 'dateCoding' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'dateCoding'.

Definition at line 157 of file lexical_resource.py.

6.14.3.7 def lmf.src.core.lexical_resource.LexicalResource.get_description (*self*)

Get description.

Attribute 'description' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'description'.

Definition at line 237 of file lexical_resource.py.

6.14.3.8 def lmf.src.core.lexical_resource.LexicalResource.get_dtdVersion (*self*)

Get DTD version.

Returns

LexicalResource attribute 'dtdVersion'.

Definition at line 78 of file lexical_resource.py.

6.14.3.9 def lmf.src.core.lexical_resource.LexicalResource.get_language_code (*self*)

Get language code.

Attribute 'languageCode' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'languageCode'.

Definition at line 93 of file lexical_resource.py.

6.14.3.10 def lmf.src.core.lexical_resource.LexicalResource.get_last_update (*self*)

Get last update.

Attribute 'lastUpdate' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'lastUpdate'.

Definition at line 205 of file lexical_resource.py.

6.14.3.11 def lmf.src.core.lexical_resource.LexicalResource.get_lexicon (*self, id*)

```
Retrieve a lexicon from its identifier.
@param id The identifier of the lexicon to retrieve.
@result A Lexicon instance, or None if not found.
```

Definition at line 61 of file lexical_resource.py.

6.14.3.12 def lmf.src.core.lexical_resource.LexicalResource.get_lexicons (*self*)

Get all lexicons maintained by the lexical resource.

Returns

A Python list of lexicons.

Definition at line 39 of file lexical_resource.py.

6.14.3.13 def lmf.src.core.lexical_resource.LexicalResource.get_license (*self*)

Get license.

Attribute 'license' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'license'.

Definition at line 125 of file lexical_resource.py.

6.14.3.14 def lmf.src.core.lexical_resource.LexicalResource.get_project_name (*self*)

Get project name.

Attribute 'projectName' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'projectName'.

Definition at line 173 of file lexical_resource.py.

6.14.3.15 def lmf.src.core.lexical_resource.LexicalResource.get_version (*self*)

Get version.

Attribute 'version' is owned by GlobalInformation.

Returns

GlobalInformation attribute 'version'.

Definition at line 109 of file lexical_resource.py.

6.14.3.16 def lmf.src.core.lexical_resource.LexicalResource.remove_lexicon (*self, lexicon*)

Remove a lexicon from the lexical resource.

Parameters

lexicon	The Lexicon instance to remove from the Lexical Resource.

Returns

Lexical Resource instance.

Definition at line 53 of file lexical_resource.py.

6.14.3.17 def lmf.src.core.lexical_resource.LexicalResource.set_author (*self, author*)

Set author.

Attribute 'author' is owned by GlobalInformation.

Parameters

author	The author's name to set.

Returns

LexicalResource instance.

Definition at line 212 of file lexical_resource.py.

6.14.3.18 def lmf.src.core.lexical_resource.LexicalResource.set_character_encoding (*self, character_encoding*)

Set character encoding.

Attribute 'characterEncoding' is owned by GlobalInformation.

Parameters

character_↩ encoding	The character encoding to use.

Returns

LexicalResource instance.

Definition at line 132 of file lexical_resource.py.

6.14.3.19 def lmf.src.core.lexical_resource.LexicalResource.set_creation_date (*self, date*)

Set creation date.

Attribute 'creationDate' is owned by GlobalInformation.

Parameters

date	The date to set, in format YYYY-MM-DD.

Returns

LexicalResource instance.

Definition at line 180 of file lexical_resource.py.

6.14.3.20 def lmf.src.core.lexical_resource.LexicalResource.set_date_coding (*self, date_coding*)

Set date coding.

Attribute 'dateCoding' is owned by GlobalInformation.

Parameters

date_coding	The date coding to use.

Returns

LexicalResource instance.

Definition at line 148 of file lexical_resource.py.

6.14.3.21 def lmf.src.core.lexical_resource.LexicalResource.set_description (*self, description*)

Set description.

Attribute 'description' is owned by GlobalInformation.

Parameters

description	The description to set.

Returns

LexicalResource instance.

Definition at line 228 of file lexical_resource.py.

6.14.3.22 def lmf.src.core.lexical_resource.LexicalResource.set_dtdVersion (*self, dtd_version*)

Set DTD version.

Parameters

dtd_version	The DTD version to use.

Returns

LexicalResource instance.

Definition at line 70 of file lexical_resource.py.

6.14.3.23 def lmf.src.core.lexical_resource.LexicalResource.set_language_code (*self, language_code*)

Set language code.

Attribute 'languageCode' is owned by GlobalInformation.

Parameters

language_code	The language code to use.

Returns

LexicalResource instance.

Definition at line 84 of file lexical_resource.py.

6.14.3.24 def lmf.src.core.lexical_resource.LexicalResource.set_last_update (*self, date*)

Set last update.

Attribute 'lastUpdate' is owned by GlobalInformation.

Parameters

date	The date to set, in format YYYY-MM-DD.

Returns

LexicalResource instance.

Definition at line 196 of file lexical_resource.py.

6.14.3.25 def lmf.src.core.lexical_resource.LexicalResource.set_license (*self, license*)

Set license.

Attribute 'license' is owned by GlobalInformation.

Parameters

license	The license to set.

Returns

LexicalResource instance.

Definition at line 116 of file lexical_resource.py.

6.14.3.26 def lmf.src.core.lexical_resource.LexicalResource.set_project_name (*self, project_name*)

Set project name.

Attribute 'projectName' is owned by GlobalInformation.

Parameters

project_name	The project's name to set.

Returns

LexicalResource instance.

Definition at line 164 of file lexical_resource.py.

6.14.3.27 def lmf.src.core.lexical_resource.LexicalResource.set_version (*self, version*)

Set version.

Attribute 'version' is owned by GlobalInformation.

Parameters

version	The version to set.

Returns

LexicalResource instance.

Definition at line 100 of file lexical_resource.py.

6.14.4 Member Data Documentation

6.14.4.1 lmf.src.core.lexical_resource.LexicalResource.dtdVersion

Definition at line 15 of file lexical_resource.py.

6.14.4.2 lmf.src.core.lexical_resource.LexicalResource.global_information

GlobalInformation instance is owned by LexicalResource There is one GlobalInformation for one LexicalResource.

Definition at line 18 of file lexical_resource.py.

6.14.4.3 lmf.src.core.lexical_resource.LexicalResource.lexicon

Lexicon instances are owned by LexicalResource There is one or more Lexicon instances for one unique LexicalResource.

Definition at line 21 of file lexical_resource.py.

6.14.4.4 lmf.src.core.lexical_resource.LexicalResource.speaker

Speaker instances are owned by LexicalResource There is zero to many Speaker instances for one unique LexicalResource.

Definition at line 24 of file lexical_resource.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_resource.py

6.15 lmf.src.core.lexicon.Lexicon Class Reference

"Lexicon is a class containing all the lexical entries of a given language within the entire resource." (LMF)

Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_id
  Set lexicon identifier.
- def get_id
  Get identifier.
- def set_entrySource
  Set lexicon entry source.
- def get_entrySource
  Get entry source.
- def set_language
  Set lexicon language.
- def get_language
  Get language.
- def set_languageScript

Set lexicon language script.

- def get_languageScript

  Get language script.

- def set_label

  Set lexicon label.

- def get_label

  Get label.

- def set_lexiconType

  Set lexicon type.

- def get_lexiconType

  Get lexicon type.

- def set_vowelHarmony
- def get_vowelHarmony
- def set_localPath

  Set lexicon local path.

- def get_localPath

  Get lexicon local path.

- def get_lexical_entries

  Get all lexical entries maintained by the lexicon.

- def add_lexical_entry

  Add a lexical entry to the lexicon.

- def remove_lexical_entry

  Remove a lexical entry from the lexicon.

- def count_lexical_entries

  Count number of lexical entries of the lexicon.

- def sort_homonym_numbers

  Sort similar given items of lexical entries contained in the lexicon according to their homonym number.

- def sort_lexical_entries

  Sort given items of lexical entries contained in the lexicon according to a certain order.

- def find_lexical_entries

  Find all lexical entries which characteristics meet the given condition.

- def check_cross_references

  Check all cross-references in the lexicon.

- def reset_check

  Reset boolean to be able to check all cross-references in the lexicon again.

- def convert_to_latex

Public Attributes

- language
- languageScript
- label
- lexiconType
- entrySource
- vowelHarmony
- localPath
- lexical_entry

  All LexicalEntry instances are maintained by Lexicon There is one or more LexicalEntry instances per Lexicon.

- id

6.15.1 Detailed Description

"Lexicon is a class containing all the lexical entries of a given language within the entire resource." (LMF)

Definition at line 9 of file lexicon.py.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 def lmf.src.core.lexicon.Lexicon.__init__ (*self, id* = `None`)

Constructor.

Lexicon instances are owned by LexicalResource.

Returns

A Lexicon instance.

Definition at line 12 of file lexicon.py.

6.15.2.2 def lmf.src.core.lexicon.Lexicon.__del__ (*self*)

Destructor.

Release LexicalEntry instances.

Definition at line 31 of file lexicon.py.

6.15.3 Member Function Documentation

6.15.3.1 def lmf.src.core.lexicon.Lexicon.add_lexical_entry (*self, lexical_entry*)

Add a lexical entry to the lexicon.

Parameters

lexical_entry	A LexicalEntry instance to add to the Lexicon.

Returns

Lexicon instance.

Definition at line 149 of file lexicon.py.

6.15.3.2 def lmf.src.core.lexicon.Lexicon.check_cross_references (*self*)

Check all cross-references in the lexicon.

Fill the private attribute '__lexicalEntry' of each RelatedForm instance for all lexical entries.

Returns

Lexicon instance.

Definition at line 283 of file lexicon.py.

6.15.3.3 def lmf.src.core.lexicon.Lexicon.convert_to_latex (*self*)

```
This method converts the lexicon into LaTeX format.
```

Definition at line 349 of file lexicon.py.

6.15.3.4 def lmf.src.core.lexicon.Lexicon.count_lexical_entries (*self*)

Count number of lexical entries of the lexicon.

Returns

The number of lexical entries without duplicates maintained by the lexicon.

Definition at line 165 of file lexicon.py.

6.15.3.5 def lmf.src.core.lexicon.Lexicon.find_lexical_entries (*self, filter*)

Find all lexical entries which characteristics meet the given condition.

Parameters

filter	Function or lambda function taking a lexical entry as argument, and returning True or False; for instance 'lambda lexical_entry: lexical_entry.get_lexeme() == "Hello"'.

Returns

A Python list of LexicalEntry instances.

Definition at line 272 of file lexicon.py.

6.15.3.6 def lmf.src.core.lexicon.Lexicon.get_entrySource (*self*)

Get entry source.

Returns

Lexicon attribute 'entrySource'.

Definition at line 61 of file lexicon.py.

6.15.3.7 def lmf.src.core.lexicon.Lexicon.get_id (*self*)

Get identifier.

Returns

Lexicon attribute 'id'.

Definition at line 47 of file lexicon.py.

6.15.3.8 def lmf.src.core.lexicon.Lexicon.get_label (*self*)

Get label.

Returns

Lexicon attribute 'label'.

Definition at line 103 of file lexicon.py.

6.15.3.9 def lmf.src.core.lexicon.Lexicon.get_language (*self*)

Get language.

Returns

Lexicon attribute 'language'.

Definition at line 75 of file lexicon.py.

6.15.3.10 def lmf.src.core.lexicon.Lexicon.get_languageScript (*self*)

Get language script.

Returns

Lexicon attribute 'languageScript'.

Definition at line 89 of file lexicon.py.

6.15.3.11 def lmf.src.core.lexicon.Lexicon.get_lexical_entries (*self*)

Get all lexical entries maintained by the lexicon.

Returns

A Python set of lexical entries.

Definition at line 143 of file lexicon.py.

6.15.3.12 def lmf.src.core.lexicon.Lexicon.get_lexiconType (*self*)

Get lexicon type.

Returns

Lexicon attribute 'lexiconType'.

Definition at line 117 of file lexicon.py.

6.15.3.13 def lmf.src.core.lexicon.Lexicon.get_localPath (*self*)

Get lexicon local path.

Returns

Lexicon attribute 'localPath'.

Definition at line 137 of file lexicon.py.

6.15.3.14 def lmf.src.core.lexicon.Lexicon.get_vowelHarmony (*self*)

Definition at line 126 of file lexicon.py.

6.15.3.15 def lmf.src.core.lexicon.Lexicon.remove_lexical_entry (*self, lexical_entry*)

Remove a lexical entry from the lexicon.

Parameters

lexical_entry	The LexicalEntry instance to remove from the Lexicon.

Returns

Lexicon instance.

Definition at line 157 of file lexicon.py.

6.15.3.16 def lmf.src.core.lexicon.Lexicon.reset_check (*self*)

Reset boolean to be able to check all cross-references in the lexicon again.

Reset the private attribute '__checked'.

Returns

Lexicon instance.

Definition at line 341 of file lexicon.py.

6.15.3.17 def lmf.src.core.lexicon.Lexicon.set_entrySource (*self, entry_source*)

Set lexicon entry source.

Parameters

entry_source	The entry source to set.

Returns

Lexicon instance.

Definition at line 53 of file lexicon.py.

6.15.3.18 def lmf.src.core.lexicon.Lexicon.set_id (*self, id*)

Set lexicon identifier.

Parameters

id	The identifier to set.

Returns

Lexicon instance.

Definition at line 39 of file lexicon.py.

6.15.3.19 def lmf.src.core.lexicon.Lexicon.set_label (*self, label*)

Set lexicon label.

Parameters

label	The label to set.

Returns

Lexicon instance.

Definition at line 95 of file lexicon.py.

6.15.3.20 def lmf.src.core.lexicon.Lexicon.set_language (*self, language*)

Set lexicon language.

Parameters

language	The language to set.

Returns

Lexicon instance.

Definition at line 67 of file lexicon.py.

6.15.3.21 def lmf.src.core.lexicon.Lexicon.set_languageScript (*self, language_script*)

Set lexicon language script.

Parameters

language_script	The language script to set.

Returns

Lexicon instance.

Definition at line 81 of file lexicon.py.

6.15.3.22 def lmf.src.core.lexicon.Lexicon.set_lexiconType (*self, lexicon_type*)

Set lexicon type.

Parameters

lexicon_type	The lexicon type to set.

Returns

Lexicon instance.

Definition at line 109 of file lexicon.py.

6.15.3.23 def lmf.src.core.lexicon.Lexicon.set_localPath (*self, local_path*)

Set lexicon local path.

Parameters

local_path	The absolute path to audio files to set.

Returns

Lexicon instance.

Definition at line 129 of file lexicon.py.

6.15.3.24 def lmf.src.core.lexicon.Lexicon.set_vowelHarmony (*self*, *vowel_harmony*)

Definition at line 123 of file lexicon.py.

6.15.3.25 def lmf.src.core.lexicon.Lexicon.sort_homonym_numbers (*self*, *items* = `lambda lexical_entry: lexical_entry.get_lexeme()`, *condition* = `lambda lexical_entry: True`)

Sort similar given items of lexical entries contained in the lexicon according to their homonym number.

Parameters

items	Lambda function giving the item to sort. Default value is 'lambda lexical_entry: lexical_entry.get_lexeme()', which means that the items to sort are lexemes.
condition	Lambda function giving a condition to apply classification.

Returns

The sorted Python list of lexical entries.

Definition at line 171 of file lexicon.py.

6.15.3.26 def lmf.src.core.lexicon.Lexicon.sort_lexical_entries (*self*, *items* = `lambda lexical_entry: lexical_entry.get_lexeme()`, *sort_order* = `None`, *comparison* = `None`)

Sort given items of lexical entries contained in the lexicon according to a certain order.

Parameters

items	Lambda function giving the item to sort. Default value is 'lambda lexical_entry: lexical_entry.get_lexeme()', which means that the items to sort are lexemes.
sort_order	Default value is 'None', which means that the lexicographical ordering uses the ASCII ordering.
comparison	Function to compare items. If 'None', a default function to compare character by character is provided.

Returns

The sorted Python list of lexical entries.

Definition at line 201 of file lexicon.py.

6.15.4 Member Data Documentation

6.15.4.1 lmf.src.core.lexicon.Lexicon.entrySource

Definition at line 22 of file lexicon.py.

6.15.4.2 lmf.src.core.lexicon.Lexicon.id

Definition at line 44 of file lexicon.py.

6.15.4.3 lmf.src.core.lexicon.Lexicon.label

Definition at line 20 of file lexicon.py.

6.15.4.4 lmf.src.core.lexicon.Lexicon.language

Definition at line 18 of file lexicon.py.

6.15.4.5 lmf.src.core.lexicon.Lexicon.languageScript

Definition at line 19 of file lexicon.py.

6.15.4.6 lmf.src.core.lexicon.Lexicon.lexical_entry

All LexicalEntry instances are maintained by Lexicon There is one or more LexicalEntry instances per Lexicon.

Definition at line 27 of file lexicon.py.

6.15.4.7 lmf.src.core.lexicon.Lexicon.lexiconType

Definition at line 21 of file lexicon.py.

6.15.4.8 lmf.src.core.lexicon.Lexicon.localPath

Definition at line 24 of file lexicon.py.

6.15.4.9 lmf.src.core.lexicon.Lexicon.vowelHarmony

Definition at line 23 of file lexicon.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexicon.py

6.16 lmf.src.morphology.list_of_components.ListOfComponents Class Reference

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def create_and_add_component

  Create and add a component to the list.
- def get_components

  Get the list of components.

Public Attributes

- component

  Component instances are owned by ListOfComponents There are two or more Component instances per ListOf↩ Components.

6.16.1 Detailed Description

```
"List of Components is a class representing the aggregative aspect of a multiword expression (MWE). The mechan
```

Definition at line 8 of file list_of_components.py.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 def lmf.src.morphology.list_of_components.ListOfComponents.__init__ (*self*)

Constructor.

ListOfComponents instance is owned by LexicalEntry.

Returns

A ListOfComponents instance.

Definition at line 11 of file list_of_components.py.

6.16.2.2 def lmf.src.morphology.list_of_components.ListOfComponents.__del__ (*self*)

Destructor.

Release Component instances.

Definition at line 20 of file list_of_components.py.

6.16.3 Member Function Documentation

6.16.3.1 def lmf.src.morphology.list_of_components.ListOfComponents.create_and_add_component (*self, position, lexeme*)

Create and add a component to the list.

Parameters

position	The position of the component in the multiword expression.
lexeme	Related lexeme.

Returns

ListOfComponents instance.

Definition at line 28 of file list_of_components.py.

6.16.3.2 def lmf.src.morphology.list_of_components.ListOfComponents.get_components (*self*)

Get the list of components.

Returns

ListOfComponents attribute 'component'.

Definition at line 37 of file list_of_components.py.

6.16.4 Member Data Documentation

6.16.4.1 lmf.src.morphology.list_of_components.ListOfComponents.component

Component instances are owned by ListOfComponents There are two or more Component instances per ListOf↩ Components.

Definition at line 18 of file list_of_components.py.

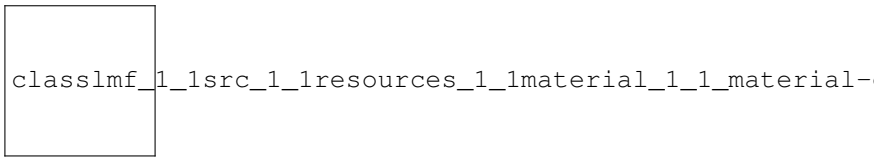
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/list_of_components.py

6.17 lmf.src.resources.material.Material Class Reference

Material is a Resource subclass.

Inheritance diagram for lmf.src.resources.material.Material:



Public Member Functions

- def __init__

  As Material is an abstract class, constructor raises an error.
- def __del__

  As Material is an abstract class, desctructor raises an error.
- def __new__

  Private initialization called from Material subclasses.

Public Attributes

- mediaType
- fileName
- author

6.17.1 Detailed Description

Material is a Resource subclass.

Material is an abstract class representing an audiovisual resource: an audio recording, a picture or a video. The Material class allows subclasses.

Definition at line 8 of file material.py.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 def lmf.src.resources.material.Material.__init__ (*self*)

As Material is an abstract class, constructor raises an error.

Definition at line 11 of file material.py.

6.17.2.2 def lmf.src.resources.material.Material.__del__ (*self*)

As Material is an abstract class, desctructor raises an error.

Definition at line 16 of file material.py.

6.17.3 Member Function Documentation

6.17.3.1 def lmf.src.resources.material.Material.__new__ (*self*)

Private initialization called from Material subclasses.

Material subinstances are owned by FormRepresentation.

Definition at line 21 of file material.py.

6.17.4 Member Data Documentation

6.17.4.1 lmf.src.resources.material.Material.author

Definition at line 27 of file material.py.

6.17.4.2 lmf.src.resources.material.Material.fileName

Definition at line 26 of file material.py.

6.17.4.3 lmf.src.resources.material.Material.mediaType

Definition at line 25 of file material.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/material.py

6.18 lmf.src.utils.error_handling.OutputError Class Reference

Exception raised for errors in the output.

Inheritance diagram for lmf.src.utils.error_handling.OutputError:

Public Member Functions

- def __init__

  Constructor.
- def handle

  Define behavior to follow in case this error is caught: display error and exit program.

Public Attributes

- msg
- expr
- frame_info

6.18.1 Detailed Description

Exception raised for errors in the output.

Definition at line 69 of file error_handling.py.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 def lmf.src.utils.error_handling.OutputError.__init__ (*self, msg, expr* = None)

Constructor.

Parameters

msg	Explanation of the error.
expr	Output expression in which the error occurred.

Returns

An OutputError instance.

Definition at line 72 of file error_handling.py.

6.18.3 Member Function Documentation

6.18.3.1 def lmf.src.utils.error_handling.OutputError.handle (*self*)

Define behavior to follow in case this error is caught: display error and exit program.

Definition at line 84 of file error_handling.py.

6.18.4 Member Data Documentation

6.18.4.1 lmf.src.utils.error_handling.OutputError.expr

Definition at line 79 of file error_handling.py.

6.18.4.2 lmf.src.utils.error_handling.OutputError.frame_info

Definition at line 82 of file error_handling.py.

6.18.4.3 lmf.src.utils.error_handling.OutputError.msg

Definition at line 78 of file error_handling.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py

6.19 lmf.src.morphosyntax.paradigm.Paradigm Class Reference

Paradigm is a class representing a morphological paradigm.

Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.
- def set_paradigmLabel

  Set paradigm label.
- def get_paradigmLabel

  Get paradigm label.
- def set_paradigm

  Set paradigm.
- def get_paradigm

  Get paradigm.
- def set_language

  Set language of the paradigm.
- def get_language

  Get paradigm language.
- def set_morphology

  Set morphology.
- def get_morphology

  Get morphology.
- def get_lexical_entry

  Get pointed lexical entry.

Public Attributes

- paradigmLabel
- paradigm
- language
- morphology
- targets

6.19.1 Detailed Description

Paradigm is a class representing a morphological paradigm.

Definition at line 10 of file paradigm.py.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 def lmf.src.morphosyntax.paradigm.Paradigm.__init__ (*self*)

Constructor.

Paradigm instances are owned by Sense.

Returns

A Paradigm instance.

Definition at line 13 of file paradigm.py.

6.19.2.2 def lmf.src.morphosyntax.paradigm.Paradigm.__del__ (*self*)

Destructor.

Definition at line 28 of file paradigm.py.

6.19.3 Member Function Documentation

6.19.3.1 def lmf.src.morphosyntax.paradigm.Paradigm.get_language (*self*)

Get paradigm language.

Returns

Paradigm attribute 'language'.

Definition at line 82 of file paradigm.py.

6.19.3.2 def lmf.src.morphosyntax.paradigm.Paradigm.get_lexical_entry (*self*)

Get pointed lexical entry.

Returns

Paradigm private attribute '__lexical_entry'.

Definition at line 102 of file paradigm.py.

6.19.3.3 def lmf.src.morphosyntax.paradigm.Paradigm.get_morphology (*self*)

Get morphology.

Returns

Paradigm attribute 'morphology'.

Definition at line 96 of file paradigm.py.

6.19.3.4 def lmf.src.morphosyntax.paradigm.Paradigm.get_paradigm (*self, language* = `None`)

Get paradigm.

Parameters

language	Language filter.

Returns

Paradigm attribute 'paradigm'.

Definition at line 64 of file paradigm.py.

6.19.3.5 def lmf.src.morphosyntax.paradigm.Paradigm.get_paradigmLabel (*self*)

Get paradigm label.

Returns

Paradigm attribute 'paradigmLabel'.

Definition at line 50 of file paradigm.py.

6.19.3.6 def lmf.src.morphosyntax.paradigm.Paradigm.set_language (*self, language*)

Set language of the paradigm.

Parameters

language	The paradigm language to set.

Returns

Paradigm instance.

Definition at line 74 of file paradigm.py.

6.19.3.7 def lmf.src.morphosyntax.paradigm.Paradigm.set_morphology (*self, morphology*)

Set morphology.

Parameters

morphology	The morphology to set.

Returns

Paradigm instance.

Definition at line 88 of file paradigm.py.

6.19.3.8 def lmf.src.morphosyntax.paradigm.Paradigm.set_paradigm (*self, paradigm*)

Set paradigm.

Parameters

paradigm	The paradigm to set.

Returns

Paradigm instance.

Definition at line 56 of file paradigm.py.

6.19.3.9 def lmf.src.morphosyntax.paradigm.Paradigm.set_paradigmLabel (*self, paradigm_label*)

Set paradigm label.

Parameters

paradigm_label	The paradigm label to set.

Returns

Paradigm instance.

Definition at line 34 of file paradigm.py.

6.19.4 Member Data Documentation

6.19.4.1 lmf.src.morphosyntax.paradigm.Paradigm.language

Definition at line 20 of file paradigm.py.

6.19.4.2 lmf.src.morphosyntax.paradigm.Paradigm.morphology

Definition at line 21 of file paradigm.py.

6.19.4.3 lmf.src.morphosyntax.paradigm.Paradigm.paradigm

Definition at line 19 of file paradigm.py.

6.19.4.4 lmf.src.morphosyntax.paradigm.Paradigm.paradigmLabel

Definition at line 18 of file paradigm.py.

6.19.4.5 lmf.src.morphosyntax.paradigm.Paradigm.targets

Definition at line 23 of file paradigm.py.

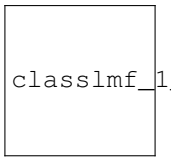
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphosyntax/paradigm.py

6.20 lmf.src.resources.picture.Picture Class Reference

Picture is a Material subclass representing a picture.

Inheritance diagram for lmf.src.resources.picture.Picture:



Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.

Public Attributes

- filename
- reference
- width
- height
- format
- statement

  Statement instances are owned by Picture There is zero to many Statement instances per Picture.

6.20.1 Detailed Description

Picture is a Material subclass representing a picture.

Definition at line 8 of file picture.py.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 def lmf.src.resources.picture.Picture.__init__ (*self*)

Constructor.

Picture instances are owned by ?.

Returns

A Picture instance.

Definition at line 11 of file picture.py.

6.20.2.2 def lmf.src.resources.picture.Picture.__del__ (*self*)

Destructor.

Release Statement instances.

Definition at line 27 of file picture.py.

6.20.3 Member Data Documentation

6.20.3.1 lmf.src.resources.picture.Picture.filename

Definition at line 18 of file picture.py.

6.20.3.2 lmf.src.resources.picture.Picture.format

Definition at line 22 of file picture.py.

6.20.3.3 lmf.src.resources.picture.Picture.height

Definition at line 21 of file picture.py.

6.20.3.4 lmf.src.resources.picture.Picture.reference

Definition at line 19 of file picture.py.

6.20.3.5 lmf.src.resources.picture.Picture.statement

Statement instances are owned by Picture There is zero to many Statement instances per Picture.

Definition at line 25 of file picture.py.

6.20.3.6 lmf.src.resources.picture.Picture.width

Definition at line 20 of file picture.py.

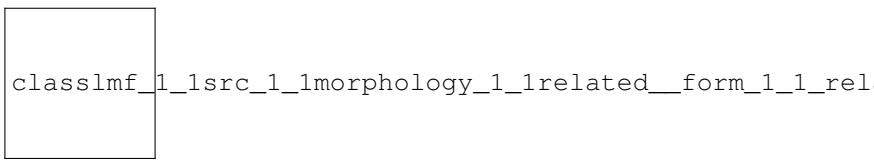
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/picture.py

6.21 lmf.src.morphology.related_form.RelatedForm Class Reference

"Related Form is a Form subclass representing a word form or a morph that can be related to the Lexical Entry. There is no asumption that the Related Form is associated with the Sense class in the Lexical Entry." (LMF)

Inheritance diagram for lmf.src.morphology.related_form.RelatedForm:



Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_semanticRelation

Set semantic relation.

- def get_semanticRelation

  Get semantic relation.

- def get_lexeme

  Get related LexicalEntry lexeme.

- def set_lexical_entry

  Set pointer to the related lexical entry instance.

- def get_lexical_entry

  Get related LexicalEntry.

Public Attributes

- semanticRelation
- targets

6.21.1 Detailed Description

"Related Form is a Form subclass representing a word form or a morph that can be related to the Lexical Entry. There is no asumption that the Related Form is associated with the Sense class in the Lexical Entry." (LMF)

Definition at line 10 of file related_form.py.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 def lmf.src.morphology.related_form.RelatedForm.__init__ (*self, lexeme* = `None`)

Constructor.

RelatedForm instances are owned by LexicalEntry.

Parameters

lexeme	Related lexeme. If not provided, default value is None.

Returns

A RelatedForm instance.

Definition at line 13 of file related_form.py.

6.21.2.2 def lmf.src.morphology.related_form.RelatedForm.__del__ (*self*)

Destructor.

Definition at line 28 of file related_form.py.

6.21.3 Member Function Documentation

6.21.3.1 def lmf.src.morphology.related_form.RelatedForm.get_lexeme (*self*)

Get related LexicalEntry lexeme.

Returns

RelatedForm attribute 'targets'.

Definition at line 53 of file related_form.py.

6.21.3.2 def lmf.src.morphology.related_form.RelatedForm.get_lexical_entry (*self*)

Get related LexicalEntry.

Returns

RelatedForm private attribute '__lexical_entry'.

Definition at line 67 of file related_form.py.

6.21.3.3 def lmf.src.morphology.related_form.RelatedForm.get_semanticRelation (*self*)

Get semantic relation.

Returns

RelatedForm attribute 'semanticRelation'.

Definition at line 47 of file related_form.py.

6.21.3.4 def lmf.src.morphology.related_form.RelatedForm.set_lexical_entry (*self, lexical_entry*)

Set pointer to the related lexical entry instance.

This function can only be called once the full dictionary has been parsed.

Parameters

lexical_entry	The related LexicalEntry.

Returns

RelatedForm instance.

Definition at line 59 of file related_form.py.

6.21.3.5 def lmf.src.morphology.related_form.RelatedForm.set_semanticRelation (*self, semantic_relation*)

Set semantic relation.

Parameters

semantic_↩ relation	The semantic relation to set.

Returns

RelatedForm instance.

Definition at line 34 of file related_form.py.

6.21.4 Member Data Documentation

6.21.4.1 lmf.src.morphology.related_form.RelatedForm.semanticRelation

Definition at line 21 of file related_form.py.

6.21.4.2 lmf.src.morphology.related_form.RelatedForm.targets

Definition at line 23 of file related_form.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/related_form.py

6.22 lmf.src.core.representation.Representation Class Reference

"Representation class is an abstract class representing a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Public Member Functions

- def __init__

  As Representation is an abstract class, constructor raises an error.
- def __del__

  As Representation is an abstract class, desctructor raises an error.
- def __new__

  Private initialization called from Representation subclasses.

Public Attributes

- comment
- writtenForm
- language
- scriptName

6.22.1 Detailed Description

"Representation class is an abstract class representing a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Definition at line 6 of file representation.py.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 def lmf.src.core.representation.Representation.__init__ (*self*)

As Representation is an abstract class, constructor raises an error.

Definition at line 9 of file representation.py.

6.22.2.2 def lmf.src.core.representation.Representation.__del__ (*self*)

As Representation is an abstract class, desctructor raises an error.

Definition at line 14 of file representation.py.

6.22.3 Member Function Documentation

6.22.3.1 def lmf.src.core.representation.Representation.__new__ (*self*)

Private initialization called from Representation subclasses.

Definition at line 19 of file representation.py.

6.22.4 Member Data Documentation

6.22.4.1 lmf.src.core.representation.Representation.comment

Definition at line 22 of file representation.py.

6.22.4.2 lmf.src.core.representation.Representation.language

Definition at line 24 of file representation.py.

6.22.4.3 lmf.src.core.representation.Representation.scriptName

Definition at line 25 of file representation.py.

6.22.4.4 lmf.src.core.representation.Representation.writtenForm

Definition at line 23 of file representation.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/representation.py

6.23 lmf.src.resources.resource.Resource Class Reference

Resource is an abstract class representing a material or a human resource.

Public Member Functions

- def __init__

  As Resource is an abstract class, constructor raises an error.
- def __del__

  As Resource is an abstract class, desctructor raises an error.
- def __call__

  Private initialization called from Resource subclasses.

6.23.1 Detailed Description

Resource is an abstract class representing a material or a human resource.

The Resource class allows subclasses.

Definition at line 6 of file resource.py.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 def lmf.src.resources.resource.Resource.__init__ (*self*)

As Resource is an abstract class, constructor raises an error.

Definition at line 9 of file resource.py.

6.23.2.2 def lmf.src.resources.resource.Resource.__del__ (*self*)

As Resource is an abstract class, desctructor raises an error.

Definition at line 14 of file resource.py.

6.23.3 Member Function Documentation

6.23.3.1 def lmf.src.resources.resource.Resource.__call__ (*self*)

Private initialization called from Resource subclasses.

Resource subinstances are owned by LexicalResource.

Definition at line 19 of file resource.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/resource.py

6.24 lmf.src.core.sense.Sense Class Reference

"Sense is a class representing one meaning of a lexical entry. The Sense class allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry." (LMF)

Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def get_id
  IDentifier.
- def set_senseNumber
  Set sense number.
- def get_senseNumber
  Get sense number.
- def create_definition
  Create a definition.
- def add_definition
  Add a definition to the sense.
- def get_definitions
  Get all definitions maintained by the sense.
- def get_last_definition
  Get the previously registered Definition instance.
- def find_definitions

Find definitions.

- def set_definition

  Set definition and language.
- def find_glosses

  Find glosses.
- def set_gloss

  Set gloss and language.
- def set_note

  Set note, note type and language.
- def find_notes

  Find notes.
- def set_usage_note

  Set usage note and language.
- def find_usage_notes

  Find usage notes.
- def set_encyclopedic_information

  Set encyclopedic information and language.
- def find_encyclopedic_informations

  Find encyclopedic informations.
- def set_restriction

  Set restriction and language.
- def find_restrictions

  Find restrictions.
- def set_borrowed_word

  Set source language (in English).
- def get_borrowed_word

  Get source language (in English).
- def set_written_form

  Set loan word.
- def get_written_form

  Get loan word.
- def set_etymology

  Set etymology.
- def get_etymology

  Get etymology.
- def set_etymology_comment

  Set etymology comment and language.
- def get_etymology_comment

  Get etymology comment.
- def get_term_source_language

  Get language used for the etymology comment.
- def set_etymology_gloss

  Set etymology gloss.
- def get_etymology_gloss

  Get etymology gloss.
- def set_etymology_source

  Set etymology source.
- def get_etymology_source

  Get etymology source.
- def set_scientific_name

  Set scientific name.

- def get_scientific_name

  Get scientific name.
- def create_paradigm

  Create a paradigm.
- def add_paradigm

  Add a paradigm to the sense.
- def get_paradigms

  Get all paradigms maintained by the sense.
- def get_last_paradigm

  Get the previously registered Paradigm instance.
- def set_paradigm_label

  Set paradigm label.
- def set_paradigm_form

  Set paradigm form and language.
- def set_morphology

  Set morphology.
- def create_and_add_context

  Create a context and add it to the list.
- def get_contexts

  Get all contexts maintained by the sense.
- def get_last_context

  Get the previously registered Context instance.
- def create_example

  Create a Context instance and set its reference.
- def create_and_add_example

  Set written form, language and script of a new Context instance.
- def add_example

  Set written form, language and script of an existing Context instance.
- def set_example_comment

  Set comment of an existing Context instance.
- def create_and_add_subject_field

  Create a subject field and add it to the list.
- def get_subject_fields

  Get all subject fields maintained by the sense.
- def set_semantic_domain

  Create a SubjectField instance and set its semantic domain and language.
- def create_and_add_equivalent

  Create an equivalent and add it to the list.
- def get_equivalents

  Get all equivalents maintained by the sense.
- def set_translation

  Create an Equivalent instance and set its translation and language.
- def get_translations

  Get all translations.

Public Attributes

- senseNumber
- id
- definition

  Definition instances are owned by Sense There is zero to many Definition instances per Sense.
- sense

  Sense instances are owned by Sense There is zero to many Sense instances per Sense.
- equivalent

  Equivalent instances are owned by Sense There is zero to many Equivalent instances per Sense.
- context

  Context instances are owned by Sense There is zero to many Context instances per Sense.
- subject_field

  SubjectField instances are owned by Sense There is zero to many SubjectField instances per Sense.
- paradigm

  Paradigm instances are owned by Sense There is zero to many Paradigm instances per Sense.

6.24.1 Detailed Description

"Sense is a class representing one meaning of a lexical entry. The Sense class allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry." (LMF)

Definition at line 12 of file sense.py.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 def lmf.src.core.sense.Sense.__init__ (*self, id* = 0)

Constructor.

Sense instances are owned by LexicalEntry.

Parameters

id	IDentifier. If not provided, default value is 0.

Returns

A Sense instance.

Definition at line 15 of file sense.py.

6.24.2.2 def lmf.src.core.sense.Sense.__del__ (*self*)

Destructor.

Release Definition, Sense, Equivalent, Context, SubjectField, Paradigm instances.

Definition at line 43 of file sense.py.

6.24.3 Member Function Documentation

6.24.3.1 def lmf.src.core.sense.Sense.add_definition (*self, definition*)

Add a definition to the sense.

Parameters

definition	The Definition instance to add to the sense.

Returns

Sense instance.

Definition at line 97 of file sense.py.

6.24.3.2 def lmf.src.core.sense.Sense.add_example (*self, written_form, language* = None, *script_name* = None)

Set written form, language and script of an existing Context instance.

Attributes 'writtenForm', 'language' and 'scriptName' are owned by TextRepresentation, which is owned by Context.

Parameters

written_form	The written form to set.
language	Language used for the written form.
script_name	The name of the script used to write the example, e.g. devanagari.

Returns

Sense instance.

Definition at line 630 of file sense.py.

6.24.3.3 def lmf.src.core.sense.Sense.add_paradigm (*self, paradigm*)

Add a paradigm to the sense.

Parameters

paradigm	The Paradigm instance to add to the sense.

Returns

Sense instance.

Definition at line 504 of file sense.py.

6.24.3.4 def lmf.src.core.sense.Sense.create_and_add_context (*self, reference* = None)

Create a context and add it to the list.

Parameters

reference	The context reference to set. If not provided, default value is None.

Returns

Context instance.

Definition at line 583 of file sense.py.

6.24.3.5 def lmf.src.core.sense.Sense.create_and_add_equivalent (*self*)

Create an equivalent and add it to the list.

Returns

Equivalent instance.

Definition at line 684 of file sense.py.

6.24.3.6 def lmf.src.core.sense.Sense.create_and_add_example (*self, written_form, language* = `None`, *script_name* = `None`)

Set written form, language and script of a new Context instance.

Attributes 'writtenForm', 'language' and 'scriptName' are owned by TextRepresentation, which is owned by Context.

Parameters

written_form	The written form to set.
language	Language used for the written form.
script_name	The name of the script used to write the example, e.g. devanagari.

Returns

Sense instance.

Definition at line 614 of file sense.py.

6.24.3.7 def lmf.src.core.sense.Sense.create_and_add_subject_field (*self*)

Create a subject field and add it to the list.

Returns

SubjectField instance.

Definition at line 660 of file sense.py.

6.24.3.8 def lmf.src.core.sense.Sense.create_definition (*self*)

Create a definition.

Returns

Definition instance.

Definition at line 91 of file sense.py.

6.24.3.9 def lmf.src.core.sense.Sense.create_example (*self, reference* = `None`)

Create a Context instance and set its reference.

Attribute 'targets' is owned by Context.

Parameters

reference	The example reference to set. If not provided, default value is None.

Returns

Sense instance.

Definition at line 605 of file sense.py.

6.24.3.10 def lmf.src.core.sense.Sense.create_paradigm (*self*)

Create a paradigm.

Returns

Paradigm instance.

Definition at line 498 of file sense.py.

6.24.3.11 def lmf.src.core.sense.Sense.find_definitions (*self, language*)

Find definitions.

This attribute is owned by Definition.

Parameters

language	The language to consider to retrieve the definition.

Returns

A Python list of found Definition attributes 'definition'.

Definition at line 118 of file sense.py.

6.24.3.12 def lmf.src.core.sense.Sense.find_encyclopedic_informations (*self, language*)

Find encyclopedic informations.

This attribute is owned by Statement, which owned by Definition.

Parameters

language	Language to consider to retrieve the encyclopedic informations.

Returns

A Python list of found Statement attributes 'encyclopedicInformation'.

Definition at line 269 of file sense.py.

6.24.3.13 def lmf.src.core.sense.Sense.find_glosses (*self, language*)

Find glosses.

This attribute is owned by Definition.

Parameters

language	The language to consider to retrieve the gloss.

Returns

A Python list of found Definition attributes 'gloss'.

Definition at line 158 of file sense.py.

6.24.3.14 def lmf.src.core.sense.Sense.find_notes (*self, type*)

Find notes.

This attribute is owned by Statement, which owned by Definition.

Parameters

type	Type of the note to consider to retrieve the note.

Returns

A Python list of found Statement attributes 'notes'.

Definition at line 215 of file sense.py.

6.24.3.15 def lmf.src.core.sense.Sense.find_restrictions (*self, language*)

Find restrictions.

This attribute is owned by Statement, which owned by Definition.

Parameters

language	Language to consider to retrieve the restriction.

Returns

A Python list of found Statement attributes 'restriction'.

Definition at line 296 of file sense.py.

6.24.3.16 def lmf.src.core.sense.Sense.find_usage_notes (*self, language*)

Find usage notes.

This attribute is owned by Statement, which owned by Definition.

Parameters

language	Language to consider to retrieve the usage note.

Returns

A Python list of found Statement attributes 'usageNote'.

Definition at line 242 of file sense.py.

6.24.3.17 def lmf.src.core.sense.Sense.get_borrowed_word (*self*)

Get source language (in English).

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'borrowedWord'.

Definition at line 322 of file sense.py.

6.24.3.18 def lmf.src.core.sense.Sense.get_contexts (*self*)

Get all contexts maintained by the sense.

Returns

A Python list of contexts.

Definition at line 592 of file sense.py.

6.24.3.19 def lmf.src.core.sense.Sense.get_definitions (*self*)

Get all definitions maintained by the sense.

Returns

A Python list of definitions.

Definition at line 105 of file sense.py.

6.24.3.20 def lmf.src.core.sense.Sense.get_equivalents (*self*)

Get all equivalents maintained by the sense.

Returns

A Python list of equivalents.

Definition at line 692 of file sense.py.

6.24.3.21 def lmf.src.core.sense.Sense.get_etymology (*self*)

Get etymology.

This attribute is owned by Statement, which is owned by Definition.

Returns

The first found Statement attribute 'etymology'.

Definition at line 374 of file sense.py.

6.24.3.22 def lmf.src.core.sense.Sense.get_etymology_comment (*self, term_source_language* = `None`)

Get etymology comment.

This attribute is owned by Statement, which is owned by Definition.

Parameters

term_source_↩ language	The language of the etymology comment to retrieve.

Returns

The first found Statement attribute 'etymologyComment'.

Definition at line 399 of file sense.py.

6.24.3.23 def lmf.src.core.sense.Sense.get_etymology_gloss (*self*)

Get etymology gloss.

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'etymologyGloss'.

Definition at line 435 of file sense.py.

6.24.3.24 def lmf.src.core.sense.Sense.get_etymology_source (*self*)

Get etymology source.

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'etymologySource'.

Definition at line 461 of file sense.py.

6.24.3.25 def lmf.src.core.sense.Sense.get_id (*self*)

IDentifier.

Returns

Sense attribute 'id'.

Definition at line 66 of file sense.py.

6.24.3.26 def lmf.src.core.sense.Sense.get_last_context (*self*)

Get the previously registered Context instance.

Returns

The last element of Sense attribute 'context'.

Definition at line 598 of file sense.py.

6.24.3.27 def lmf.src.core.sense.Sense.get_last_definition (*self*)

Get the previously registered Definition instance.

Returns

The last element of Sense attribute 'definition'.

Definition at line 111 of file sense.py.

6.24.3.28 def lmf.src.core.sense.Sense.get_last_paradigm (*self*)

Get the previously registered Paradigm instance.

Returns

The last element of Sense attribute 'paradigm'.

Definition at line 518 of file sense.py.

6.24.3.29 def lmf.src.core.sense.Sense.get_paradigms (*self*)

Get all paradigms maintained by the sense.

Returns

A Python list of paradigms.

Definition at line 512 of file sense.py.

6.24.3.30 def lmf.src.core.sense.Sense.get_scientific_name (*self*)

Get scientific name.

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'scientificName'.

Definition at line 487 of file sense.py.

6.24.3.31 def lmf.src.core.sense.Sense.get_senseNumber (*self, integer* = `False`)

Get sense number.

If True, return a numerical value.

Returns

Sense attribute 'senseNumber'.

Definition at line 80 of file sense.py.

6.24.3.32 def lmf.src.core.sense.Sense.get_subject_fields (*self*)

Get all subject fields maintained by the sense.

Returns

A Python list of subject fields.

Definition at line 668 of file sense.py.

6.24.3.33 def lmf.src.core.sense.Sense.get_term_source_language (*self*)

Get language used for the etymology comment.

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'termSourceLanguage'.

Definition at line 409 of file sense.py.

6.24.3.34 def lmf.src.core.sense.Sense.get_translations (*self, language* = `None`)

Get all translations.

This attribute is owned by Equivalent.

Parameters

language	If this argument is given, get only translations that are described using this language.

Returns

A Python list of filtered Equivalent attributes 'translation'.

Definition at line 708 of file sense.py.

6.24.3.35 def lmf.src.core.sense.Sense.get_written_form (*self*)

Get loan word.

This attribute is owned by Statement, which is owned by Definition.

Returns

Statement attribute 'writtenForm'.

Definition at line 348 of file sense.py.

6.24.3.36 def lmf.src.core.sense.Sense.set_borrowed_word (*self, borrowed_word*)

Set source language (in English).

Attribute 'borrowedWord' is owned by Statement, which is owned by Definition.

Parameters

borrowed_word	Source language.

Returns

Sense instance.

Definition at line 307 of file sense.py.

6.24.3.37 def lmf.src.core.sense.Sense.set_definition (*self, definition, language* = None)

Set definition and language.

These attributes are owned by Definition.

Parameters

definition	Definition.
language	Language of definition.

Returns

Sense instance.

Definition at line 130 of file sense.py.

6.24.3.38 def lmf.src.core.sense.Sense.set_encyclopedic_information (*self, encyclopedic_information, language* = None)

Set encyclopedic information and language.

These attributes are owned by Statement, which is owned by Definition.

Parameters

encyclopedic_↩ information	Encyclopedic information to set.
language	Language used for the encyclopedic information.

Returns

Sense instance.

Definition at line 253 of file sense.py.

6.24.3.39 def lmf.src.core.sense.Sense.set_etymology (*self, etymology*)

Set etymology.

Attribute 'etymology' is owned by Statement, which is owned by Definition.

Parameters

etymology	Etymology.

Returns

Sense instance.

Definition at line 359 of file sense.py.

6.24.3.40 def lmf.src.core.sense.Sense.set_etymology_comment (*self, etymology_comment, term_source_language =* None)

Set etymology comment and language.

Attributes 'etymologyComment' and 'termSourceLanguage' are owned by Statement, which is owned by Definition.

Parameters

etymology_↩ comment	Etymology comment.
term_source_↩ language	Language of the comment.

Returns

Sense instance.

Definition at line 383 of file sense.py.

6.24.3.41 def lmf.src.core.sense.Sense.set_etymology_gloss (*self, etymology_gloss*)

Set etymology gloss.

Attribute 'etymologyGloss' is owned by Statement, which is owned by Definition.

Parameters

etymology_gloss	Etymology gloss.

Returns

Sense instance.

Definition at line 420 of file sense.py.

6.24.3.42 def lmf.src.core.sense.Sense.set_etymology_source (*self, etymology_source*)

Set etymology source.

Attribute 'etymologySource' is owned by Statement, which is owned by Definition.

Parameters

etymology_↩ source	Etymology source.

Returns

Sense instance.

Definition at line 446 of file sense.py.

6.24.3.43 def lmf.src.core.sense.Sense.set_example_comment (*self, comment*)

Set comment of an existing Context instance.

Attribute 'comment' is owned by TextRepresentation, which is owned by Context.

Parameters

comment	The comment to set.

Returns

Sense instance.

Definition at line 646 of file sense.py.

6.24.3.44 def lmf.src.core.sense.Sense.set_gloss (*self, gloss, language* = None)

Set gloss and language.

These attributes are owned by Definition.

Parameters

gloss	Gloss.
language	Language of gloss.

Returns

Sense instance.

Definition at line 170 of file sense.py.

6.24.3.45 def lmf.src.core.sense.Sense.set_morphology (*self, morphology*)

Set morphology.

Attribute 'morphology' is owned by Paradigm.

Parameters

morphology	Morphology.

Returns

Sense instance.

Definition at line 564 of file sense.py.

6.24.3.46 def lmf.src.core.sense.Sense.set_note (*self, note, type* = None, *language* = None)

Set note, note type and language.

These attributes are owned by Statement, which is owned by Definition.

Parameters

note	Note to set.
type	Type of the note.
language	Language used for the note.

Returns

Sense instance.

Definition at line 198 of file sense.py.

6.24.3.47 def lmf.src.core.sense.Sense.set_paradigm_form (*self, paradigm_form, language* = None)

Set paradigm form and language.

Attributes 'paradigm' and 'language' are owned by Paradigm.

Parameters

paradigm_form	Paradigm form.
language	Language used for the paradigm form.

Returns

Sense instance.

Definition at line 535 of file sense.py.

6.24.3.48 def lmf.src.core.sense.Sense.set_paradigm_label (*self, paradigm_label*)

Set paradigm label.

Attribute 'paradigmLabel' is owned by Paradigm.

Parameters

paradigm_label	Paradigm label.

Returns

Sense instance.

Definition at line 525 of file sense.py.

6.24.3.49 def lmf.src.core.sense.Sense.set_restriction (*self, restriction, language* = None)

Set restriction and language.

These attributes are owned by Statement, which is owned by Definition.

Parameters

restriction	Restriction to set.
language	Language used for the restriction.

Returns

Sense instance.

Definition at line 280 of file sense.py.

6.24.3.50 def lmf.src.core.sense.Sense.set_scientific_name (*self, scientific_name*)

Set scientific name.

Attribute 'scientificName' is owned by Statement, which is owned by Definition.

Parameters

scientific_name	Scientific name.

Returns

Sense instance.

Definition at line 472 of file sense.py.

6.24.3.51 def lmf.src.core.sense.Sense.set_semantic_domain (*self, semantic_domain, language* = None)

Create a SubjectField instance and set its semantic domain and language.

Attributes 'semanticDomain' and 'language' are owned by SubjectField.

Parameters

semantic_domain	The semantic domain to set.
language	Language used to describe the semantic domain.

Returns

Sense instance.

Definition at line 674 of file sense.py.

6.24.3.52 def lmf.src.core.sense.Sense.set_senseNumber (*self, sense_number*)

Set sense number.

Parameters

sense_number	The sense number to set.

Returns

Sense instance.

Definition at line 72 of file sense.py.

6.24.3.53 def lmf.src.core.sense.Sense.set_translation (*self, translation, language* = None)

Create an Equivalent instance and set its translation and language.

Attributes 'translation' and 'language' are owned by Equivalent.

Parameters

translation	The translation to set.
language	Language used for the translation.

Returns

Sense instance.

Definition at line 698 of file sense.py.

6.24.3.54 def lmf.src.core.sense.Sense.set_usage_note (*self, usage_note, language* = None)

Set usage note and language.

These attributes are owned by Statement, which is owned by Definition.

Parameters

usage_note	Usage note to set.
language	Language used for the usage note.

Returns

Sense instance.

Definition at line 226 of file sense.py.

6.24.3.55 def lmf.src.core.sense.Sense.set_written_form (*self, written_form*)

Set loan word.

Attribute 'writtenForm' is owned by Statement, which is owned by Definition.

Parameters

written_form	Loan word.

Returns

Sense instance.

Definition at line 333 of file sense.py.

6.24.4 Member Data Documentation

6.24.4.1 lmf.src.core.sense.Sense.context

Context instances are owned by Sense There is zero to many Context instances per Sense.

Definition at line 35 of file sense.py.

6.24.4.2 lmf.src.core.sense.Sense.definition

Definition instances are owned by Sense There is zero to many Definition instances per Sense.

Definition at line 26 of file sense.py.

6.24.4.3 lmf.src.core.sense.Sense.equivalent

Equivalent instances are owned by Sense There is zero to many Equivalent instances per Sense.

Definition at line 32 of file sense.py.

6.24.4.4 lmf.src.core.sense.Sense.id

Definition at line 23 of file sense.py.

6.24.4.5 lmf.src.core.sense.Sense.paradigm

Paradigm instances are owned by Sense There is zero to many Paradigm instances per Sense.

Definition at line 41 of file sense.py.

6.24.4.6 lmf.src.core.sense.Sense.sense

Sense instances are owned by Sense There is zero to many Sense instances per Sense.

Definition at line 29 of file sense.py.

6.24.4.7 lmf.src.core.sense.Sense.senseNumber

Definition at line 21 of file sense.py.

6.24.4.8 lmf.src.core.sense.Sense.subject_field

SubjectField instances are owned by Sense There is zero to many SubjectField instances per Sense.

Definition at line 38 of file sense.py.

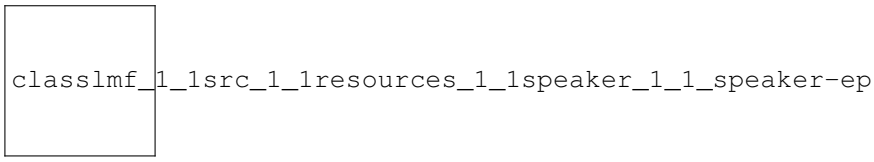
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/sense.py

6.25 lmf.src.resources.speaker.Speaker Class Reference

Speaker is a HumanResource subclass.

Inheritance diagram for lmf.src.resources.speaker.Speaker:



Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.

Public Attributes

- speakerID

6.25.1 Detailed Description

Speaker is a HumanResource subclass.

The Speaker is a class representing a speaker.

Definition at line 8 of file speaker.py.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 def lmf.src.resources.speaker.Speaker.__init__ (*self*)

Constructor.

Speaker instances are owned by LexicalResource.

Returns

A Speaker instance.

Definition at line 11 of file speaker.py.

6.25.2.2 def lmf.src.resources.speaker.Speaker.__del__ (*self*)

Destructor.

Definition at line 20 of file speaker.py.

6.25.3 Member Data Documentation

6.25.3.1 lmf.src.resources.speaker.Speaker.speakerID

Definition at line 18 of file speaker.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/speaker.py

6.26 lmf.src.core.statement.Statement Class Reference

"Statement is a class representating a narrative description that refines or complements Definition." (LMF)

Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_note
  Set note.
- def get_note
  Get note.
- def set_language
  Set language used for the note.
- def get_language
  Get language used for the note.
- def set_noteType
  Set type of the note.
- def get_noteType
  Get type of the note.
- def set_usageNote

Set usage note.

- def get_usageNote

  Get usage note.

- def set_encyclopedicInformation

  Set encyclopedic information.

- def get_encyclopedicInformation

  Get encyclopedic information.

- def set_restriction

  Set restriction.

- def get_restriction

  Get restriction.

- def set_borrowedWord
- def get_borrowedWord
- def set_writtenForm
- def get_writtenForm
- def set_etymology
- def get_etymology
- def set_etymologyComment
- def get_etymologyComment
- def set_termSourceLanguage
- def get_termSourceLanguage
- def set_etymologyGloss
- def get_etymologyGloss
- def set_etymologySource
- def get_etymologySource
- def set_scientificName
- def get_scientificName

Public Attributes

- noteType
- note
- language
- encyclopedicInformation
- usageNote
- restriction
- derivation
- borrowedWord
- writtenForm
- sense
- etymology
- etymologyComment
- etymologyGloss
- etymologySource
- termSourceLanguage
- targetLexicalEntry
- scientificName
- text_representation

  TextRepresentation instances are owned by Statement There is zero to many TextRepresentation instances per Statement.

6.26.1 Detailed Description

"Statement is a class representating a narrative description that refines or complements Definition." (LMF)

Definition at line 9 of file statement.py.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 def lmf.src.core.statement.Statement.__init__ (*self*)

Constructor.

Statement instances are owned by Definition.

Returns

A Statement instance.

Definition at line 12 of file statement.py.

6.26.2.2 def lmf.src.core.statement.Statement.__del__ (*self*)

Destructor.

Release TextRepresentation instances.

Definition at line 38 of file statement.py.

6.26.3 Member Function Documentation

6.26.3.1 def lmf.src.core.statement.Statement.get_borrowedWord (*self*)

```
Get source language (in English).
@return Statement attribute 'borrowedWord'.
```

Definition at line 179 of file statement.py.

6.26.3.2 def lmf.src.core.statement.Statement.get_encyclopedicInformation (*self*, *language* = None)

Get encyclopedic information.

Parameters

language	If this argument is given, get encyclopedic information only if written in this language.

Returns

The filtered Statement attribute 'encyclopedicInformation'.

Definition at line 138 of file statement.py.

6.26.3.3 def lmf.src.core.statement.Statement.get_etymology (*self*)

```
Get etymology.
@return Statement attribute 'etymology'.
```

Definition at line 211 of file statement.py.

6.26.3.4 def lmf.src.core.statement.Statement.get_etymologyComment (*self, term_source_language* = None)

```
Get etymology comment (in English).
@param term_source_language The language of the etymology comment to retrieve.
@return Statement attribute 'etymologyComment'.
```

Definition at line 230 of file statement.py.

6.26.3.5 def lmf.src.core.statement.Statement.get_etymologyGloss (*self*)

```
Get etymology gloss.
@return Statement attribute 'etymologyGloss'.
```

Definition at line 264 of file statement.py.

6.26.3.6 def lmf.src.core.statement.Statement.get_etymologySource (*self*)

```
Get etymology source.
@return Statement attribute 'etymologySource'.
```

Definition at line 280 of file statement.py.

6.26.3.7 def lmf.src.core.statement.Statement.get_language (*self*)

Get language used for the note.

Returns

Statement attribute 'language'.

Definition at line 83 of file statement.py.

6.26.3.8 def lmf.src.core.statement.Statement.get_note (*self, type* = None, *language* = None)

Get note.

Parameters

type	If this argument is given, get note only if its type corresponds.
language	If this argument is given, get note only if written in this language.

Returns

The filtered Statement attribute 'note'.

Definition at line 60 of file statement.py.

6.26.3.9 def lmf.src.core.statement.Statement.get_noteType (*self*)

Get type of the note.

Returns

Statement attribute 'noteType'.

Definition at line 100 of file statement.py.

6.26.3.10 def lmf.src.core.statement.Statement.get_restriction (*self, language* = None)

Get restriction.

Parameters

language	If this argument is given, get restriction only if written in this language.

Returns

The filtered Statement attribute 'restriction'.

Definition at line 159 of file statement.py.

6.26.3.11 def lmf.src.core.statement.Statement.get_scientificName (*self*)

```
Get scientific name.
@return Statement attribute 'scientificName'.
```

Definition at line 296 of file statement.py.

6.26.3.12 def lmf.src.core.statement.Statement.get_termSourceLanguage (*self*)

```
Get language used for the etymology comment.
@return Statement attribute 'termSourceLanguage'.
```

Definition at line 248 of file statement.py.

6.26.3.13 def lmf.src.core.statement.Statement.get_usageNote (*self, language* = None)

Get usage note.

Parameters

language	If this argument is given, get usage note only if written in this language.

Returns

The filtered Statement attribute 'usageNote'.

Definition at line 117 of file statement.py.

6.26.3.14 def lmf.src.core.statement.Statement.get_writtenForm (*self*)

```
Get loan word.
@return Statement attribute 'writtenForm'.
```

Definition at line 195 of file statement.py.

6.26.3.15 def lmf.src.core.statement.Statement.set_borrowedWord (*self, borrowed_word*)

```
Set source language (in English), e.g. "Chinese".
@param borrowed_word The source language to set.
@return Statement instance.
```

Definition at line 169 of file statement.py.

6.26.3.16 def lmf.src.core.statement.Statement.set_encyclopedicInformation (*self, encyclopedic_information, language* = None)

Set encyclopedic information.

Parameters

encyclopedic_↩ information	Encyclopedic information to set.
language	Language used for the encyclopedic information.

Returns

Statement instance.

Definition at line 127 of file statement.py.

6.26.3.17 def lmf.src.core.statement.Statement.set_etymology (*self, etymology*)

```
Set etymology.
@param etymolgy The etymology to set.
@return Statement instance.
```

Definition at line 201 of file statement.py.

6.26.3.18 def lmf.src.core.statement.Statement.set_etymologyComment (*self, etymology_comment, term_source_language* = None)

```
Set etymology comment (in English).
@param etymolgy_comment The etymology comment to set.
@param term_source_language The language used for the comment.
@return Statement instance.
```

Definition at line 217 of file statement.py.

6.26.3.19 def lmf.src.core.statement.Statement.set_etymologyGloss (*self, etymology_gloss*)

```
Set etymology gloss.
@param etymolgy_gloss The etymology gloss to set.
@return Statement instance.
```

Definition at line 254 of file statement.py.

6.26.3.20 def lmf.src.core.statement.Statement.set_etymologySource (*self, etymology_source*)

```
Set etymology source.
@param etymolgy_source The etymology source to set.
@return Statement instance.
```

Definition at line 270 of file statement.py.

6.26.3.21 def lmf.src.core.statement.Statement.set_language (*self, language*)

Set language used for the note.

Parameters

language	Language used for the note.

Returns

Statement instance.

Definition at line 73 of file statement.py.

6.26.3.22 def lmf.src.core.statement.Statement.set_note (*self, note, type* = None, *language* = None)

Set note.

Parameters

note	Note to set.
type	Type of the note.
language	Language used for the note.

Returns

Statement instance.

Definition at line 46 of file statement.py.

6.26.3.23 def lmf.src.core.statement.Statement.set_noteType (*self, note_type*)

Set type of the note.

Parameters

note_type	Type of the note.

Returns

Statement instance.

Definition at line 89 of file statement.py.

6.26.3.24 def lmf.src.core.statement.Statement.set_restriction (*self, restriction, language* = None)

Set restriction.

Parameters

restriction	Restriction to set.
language	Language used for the restriction.

Returns

Statement instance.

Definition at line 148 of file statement.py.

6.26.3.25 def lmf.src.core.statement.Statement.set_scientificName (*self, scientific_name*)

```
Set scientific name.
@param scientific_name The scientific name to set.
@return Statement instance.
```

Definition at line 286 of file statement.py.

6.26.3.26 def lmf.src.core.statement.Statement.set_termSourceLanguage (*self, term_source_language*)

```
Set language used for the etymology comment.
@param term_source_language The etymology comment language to set.
@return Statement instance.
```

Definition at line 238 of file statement.py.

6.26.3.27 def lmf.src.core.statement.Statement.set_usageNote (*self, usage_note, language* = None)

Set usage note.

Parameters

usage_note	Usage note to set.
language	Language used for the usage note.

Returns

Statement instance.

Definition at line 106 of file statement.py.

6.26.3.28 def lmf.src.core.statement.Statement.set_writtenForm (*self, written_form*)

```
Set loan word.
@param written_form The loan word to set.
@return Statement instance.
```

Definition at line 185 of file statement.py.

6.26.4 Member Data Documentation

6.26.4.1 lmf.src.core.statement.Statement.borrowedWord

Definition at line 24 of file statement.py.

6.26.4.2 lmf.src.core.statement.Statement.derivation

Definition at line 23 of file statement.py.

6.26.4.3 lmf.src.core.statement.Statement.encyclopedicInformation

Definition at line 20 of file statement.py.

6.26.4.4 lmf.src.core.statement.Statement.etymology

Definition at line 27 of file statement.py.

6.26.4.5 lmf.src.core.statement.Statement.etymologyComment

Definition at line 28 of file statement.py.

6.26.4.6 lmf.src.core.statement.Statement.etymologyGloss

Definition at line 29 of file statement.py.

6.26.4.7 lmf.src.core.statement.Statement.etymologySource

Definition at line 30 of file statement.py.

6.26.4.8 lmf.src.core.statement.Statement.language

Definition at line 19 of file statement.py.

6.26.4.9 lmf.src.core.statement.Statement.note

Definition at line 18 of file statement.py.

6.26.4.10 lmf.src.core.statement.Statement.noteType

Definition at line 17 of file statement.py.

6.26.4.11 lmf.src.core.statement.Statement.restriction

Definition at line 22 of file statement.py.

6.26.4.12 lmf.src.core.statement.Statement.scientificName

Definition at line 33 of file statement.py.

6.26.4.13 lmf.src.core.statement.Statement.sense

Definition at line 26 of file statement.py.

6.26.4.14 lmf.src.core.statement.Statement.targetLexicalEntry

Definition at line 32 of file statement.py.

6.26.4.15 lmf.src.core.statement.Statement.termSourceLanguage

Definition at line 31 of file statement.py.

6.26.4.16 lmf.src.core.statement.Statement.text_representation

TextRepresentation instances are owned by Statement There is zero to many TextRepresentation instances per Statement.

Definition at line 36 of file statement.py.

6.26.4.17 lmf.src.core.statement.Statement.usageNote

Definition at line 21 of file statement.py.

6.26.4.18 lmf.src.core.statement.Statement.writtenForm

Definition at line 25 of file statement.py.

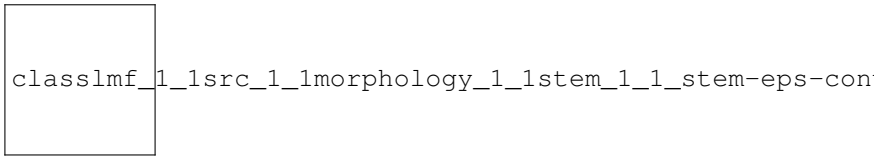
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/statement.py

6.27 lmf.src.morphology.stem.Stem Class Reference

"Stem is a Form subclass representing a morph, thus manages the sublexme parts" (LMF)

Inheritance diagram for lmf.src.morphology.stem.Stem:



Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.

6.27.1 Detailed Description

"Stem is a Form subclass representing a morph, thus manages the sublexme parts" (LMF)

Definition at line 8 of file stem.py.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 def lmf.src.morphology.stem.Stem.__init__ (*self*)

Constructor.

Stem instances are owned by LexicalEntry.

Returns

A Stem instance.

Definition at line 11 of file stem.py.

6.27.2.2 def lmf.src.morphology.stem.Stem.__del__ (*self*)

Destructor.

Definition at line 19 of file stem.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/stem.py

6.28 lmf.src.mrd.subject_field.SubjectField Class Reference

"Subject Field is a class representing a text string that provides domain or status information." (LMF)

Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_semanticDomain
  Set semantic domain and language.
- def get_semanticDomain
  Get semantic domain.
- def set_language
  Set language used for semantic domain.
- def get_language
  Get language used for semantic domain.
- def create_and_add_subject_field
  Create and add a subject field.
- def get_subject_fields
  Get all subject fields maintained by this subject field.
- def set_sub_domain
  Set a sub-domain and language.
- def get_sub_domains
  Get all sub-domains.

Public Attributes

- language
- semanticDomain
- subject_field
  SubjectField instances are owned by SubjectField There is zero to many SubjectField instances per SubjectField.

6.28.1 Detailed Description

"Subject Field is a class representing a text string that provides domain or status information." (LMF)

Definition at line 8 of file subject_field.py.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 def lmf.src.mrd.subject_field.SubjectField.__init__ (*self*)

Constructor.

SubjectField instances are owned by Sense.

Returns

A SubjectField instance.

Definition at line 11 of file subject_field.py.

6.28.2.2 def lmf.src.mrd.subject_field.SubjectField.__del__ (*self*)

Destructor.

Release SubjectField instances.

Definition at line 22 of file subject_field.py.

6.28.3 Member Function Documentation

6.28.3.1 def lmf.src.mrd.subject_field.SubjectField.create_and_add_subject_field (*self*)

Create and add a subject field.

Returns

The created SubjectField instance.

Definition at line 67 of file subject_field.py.

6.28.3.2 def lmf.src.mrd.subject_field.SubjectField.get_language (*self*)

Get language used for semantic domain.

Returns

SubjectField attribute 'language'.

Definition at line 61 of file subject_field.py.

6.28.3.3 def lmf.src.mrd.subject_field.SubjectField.get_semanticDomain (*self*, *language* = `None`)

Get semantic domain.

Parameters

language	If this argument is given, get semantic domain only if written in this language.

Returns

The filtered SubjectField attribute 'semanticDomain'.

Definition at line 43 of file subject_field.py.

6.28.3.4 def lmf.src.mrd.subject_field.SubjectField.get_sub_domains (*self*, *language* = `None`)

Get all sub-domains.

Attribute 'semanticDomain' is owned by SubjectField, which is owned by SubjectField, etc.

Parameters

language	If this argument is given, get only semantic domains that are described using this language.

Returns

A Python list of all SubjectField attributes 'semanticDomain'.

Definition at line 90 of file subject_field.py.

6.28.3.5 def lmf.src.mrd.subject_field.SubjectField.get_subject_fields (*self*)

Get all subject fields maintained by this subject field.

Returns

A Python list of subject fields.

Definition at line 75 of file subject_field.py.

6.28.3.6 def lmf.src.mrd.subject_field.SubjectField.set_language (*self, language*)

Set language used for semantic domain.

Parameters

language	Language used to describe the semantic domain.

Returns

SubjectField instance.

Definition at line 51 of file subject_field.py.

6.28.3.7 def lmf.src.mrd.subject_field.SubjectField.set_semanticDomain (*self, semantic_domain, language* = `None`)

Set semantic domain and language.

Parameters

semantic_↩ domain	The semantic domain to set.
language	Language used to describe the semantic domain.

Returns

SubjectField instance.

Definition at line 30 of file subject_field.py.

6.28.3.8 def lmf.src.mrd.subject_field.SubjectField.set_sub_domain (*self, semantic_domain, language* = `None`)

Set a sub-domain and language.

Parameters

semantic_↩ domain	The sub-domain to set.
language	Language used to describe the sub-domain.

Returns

SubjectField instance.

Definition at line 81 of file subject_field.py.

6.28.4 Member Data Documentation

6.28.4.1 lmf.src.mrd.subject_field.SubjectField.language

Definition at line 16 of file subject_field.py.

6.28.4.2 lmf.src.mrd.subject_field.SubjectField.semanticDomain

Definition at line 17 of file subject_field.py.

6.28.4.3 lmf.src.mrd.subject_field.SubjectField.subject_field

SubjectField instances are owned by SubjectField There is zero to many SubjectField instances per SubjectField.

Definition at line 20 of file subject_field.py.

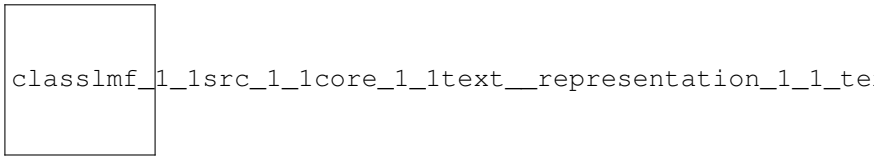
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/subject_field.py

6.29 lmf.src.core.text_representation.TextRepresentation Class Reference

"Text Representation is a class representing the textual content of definition or statement. When there is more than one variant orthography, the Text Representation instance contains a Unicode string representing the textual content as well as unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Inheritance diagram for lmf.src.core.text_representation.TextRepresentation:



Public Member Functions

- def __init__
  Constructor.
- def __del__
  Destructor.
- def set_comment
  Set written form comment.
- def get_comment
  Get written form comment.
- def set_writtenForm
  Set written form and language.
- def get_writtenForm
  Get written form.
- def set_language
  Set language used for written form.
- def get_language
  Get language used for written form.

- def set_scriptName

  Set script name.
- def get_scriptName

  Get script name.

Public Attributes

- font
- comment
- writtenForm
- language
- scriptName

6.29.1 Detailed Description

"Text Representation is a class representing the textual content of definition or statement. When there is more than one variant orthography, the Text Representation instance contains a Unicode string representing the textual content as well as unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Definition at line 9 of file text_representation.py.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 def lmf.src.core.text_representation.TextRepresentation.__init__ (*self*)

Constructor.

TextRepresentation instances are owned by Definition and Statement.

Returns

A TextRepresentation instance.

Definition at line 12 of file text_representation.py.

6.29.2.2 def lmf.src.core.text_representation.TextRepresentation.__del__ (*self*)

Destructor.

Definition at line 21 of file text_representation.py.

6.29.3 Member Function Documentation

6.29.3.1 def lmf.src.core.text_representation.TextRepresentation.get_comment (*self*)

Get written form comment.

Returns

Representation attribute 'comment'.

Definition at line 36 of file text_representation.py.

6.29.3.2 def lmf.src.core.text_representation.TextRepresentation.get_language (*self*)

Get language used for written form.

Returns

Representation attribute 'language'.

Definition at line 73 of file text_representation.py.

6.29.3.3 def lmf.src.core.text_representation.TextRepresentation.get_scriptName (*self*)

Get script name.

Returns

Representation attribute 'scriptName'.

Definition at line 89 of file text_representation.py.

6.29.3.4 def lmf.src.core.text_representation.TextRepresentation.get_writtenForm (*self, language* = `None`)

Get written form.

Parameters

language	If this argument is given, get written form only if written in this language.

Returns

The filtered Representation attribute 'writtenForm'.

Definition at line 55 of file text_representation.py.

6.29.3.5 def lmf.src.core.text_representation.TextRepresentation.set_comment (*self, comment*)

Set written form comment.

Parameters

comment	Comment about the written form.

Returns

TextRepresentation instance.

Definition at line 26 of file text_representation.py.

6.29.3.6 def lmf.src.core.text_representation.TextRepresentation.set_language (*self, language*)

Set language used for written form.

Parameters

language	Language used for the written form.

Returns

TextRepresentation instance.

Definition at line 63 of file text_representation.py.

6.29.3.7 def lmf.src.core.text_representation.TextRepresentation.set_scriptName (*self, script_name*)

Set script name.

Parameters

script_name	The script name to set.

Returns

TextRepresentation instance.

Definition at line 79 of file text_representation.py.

6.29.3.8 def lmf.src.core.text_representation.TextRepresentation.set_writtenForm (*self, written_form, language* = `None`)

Set written form and language.

Parameters

written_form	The written form to set.
language	Language used for the written form.

Returns

TextRepresentation instance.

Definition at line 42 of file text_representation.py.

6.29.4 Member Data Documentation

6.29.4.1 lmf.src.core.text_representation.TextRepresentation.comment

Definition at line 33 of file text_representation.py.

6.29.4.2 lmf.src.core.text_representation.TextRepresentation.font

Definition at line 19 of file text_representation.py.

6.29.4.3 lmf.src.core.text_representation.TextRepresentation.language

Definition at line 70 of file text_representation.py.

6.29.4.4 lmf.src.core.text_representation.TextRepresentation.scriptName

Definition at line 86 of file text_representation.py.

6.29.4.5 lmf.src.core.text_representation.TextRepresentation.writtenForm

Definition at line 50 of file text_representation.py.

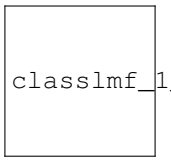
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/text_representation.py

6.30 lmf.src.resources.video.Video Class Reference

Video is a Material subclass representing a video.

Inheritance diagram for lmf.src.resources.video.Video:



Public Member Functions

- def __init__

  Constructor.
- def __del__

  Destructor.

Public Attributes

- description

6.30.1 Detailed Description

Video is a Material subclass representing a video.

Definition at line 8 of file video.py.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 def lmf.src.resources.video.Video.__init__ (*self*)

Constructor.

Video instances are owned by ?.

Returns

A Video instance.

Definition at line 11 of file video.py.

6.30.2.2 def lmf.src.resources.video.Video.__del__ (*self*)

Destructor.

Definition at line 20 of file video.py.

6.30.3 Member Data Documentation

6.30.3.1 lmf.src.resources.video.Video.description

Definition at line 18 of file video.py.

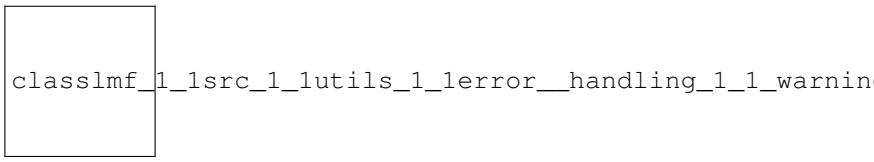
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/video.py

6.31 lmf.src.utils.error_handling.Warning Class Reference

Base class for warnings in this library.

Inheritance diagram for lmf.src.utils.error_handling.Warning:



Public Member Functions

- def __init__
  Constructor.
- def __str__
  Build the string to be displayed.

Public Attributes

- msg
- frame_info

6.31.1 Detailed Description

Base class for warnings in this library.

Definition at line 97 of file error_handling.py.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 def lmf.src.utils.error_handling.Warning.__init__ (*self, msg*)

Constructor.

Parameters

msg	String to be reported to user.

Returns

A Warning instance.

Definition at line 100 of file error_handling.py.

6.31.3 Member Function Documentation

6.31.3.1 def lmf.src.utils.error_handling.Warning.__str__ (*self*)

Build the string to be displayed.

Returns

A Python string.

Definition at line 110 of file error_handling.py.

6.31.4 Member Data Documentation

6.31.4.1 lmf.src.utils.error_handling.Warning.frame_info

Definition at line 108 of file error_handling.py.

6.31.4.2 lmf.src.utils.error_handling.Warning.msg

Definition at line 105 of file error_handling.py.

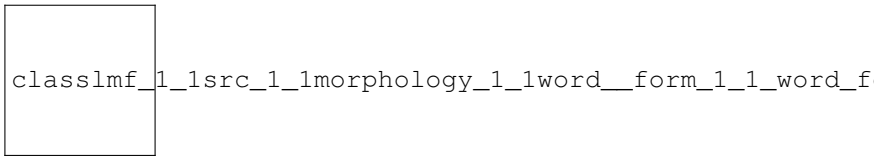
The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py

6.32 lmf.src.morphology.word_form.WordForm Class Reference

"Word Form is a Form subclass representing a form that a lexeme can take when used in a sentence or a phrase." (LMF)

Inheritance diagram for lmf.src.morphology.word_form.WordForm:



Public Member Functions

- def __init__

  Constructor.

- def __del__

  Destructor.
- def create_form_representation

  Create a form representation.
- def add_form_representation

  Add a form representation to the word form.
- def get_form_representations

  Get all form representations maintained by the word form.
- def set_written_form

  Set written form.
- def get_written_forms

  Get all written forms.
- def set_variant_form

  Set variant form.
- def get_variant_forms

  Get all variant forms.
- def set_person

  Set grammatical person.
- def get_person

  Get grammatical person.
- def set_anymacy

  Set grammatical anymacy.
- def get_anymacy

  Get anymacy.
- def set_grammaticalNumber

  Set grammatical number.
- def get_grammaticalNumber

  Get grammatical number.
- def set_clusivity

  Set grammatical clusivity.
- def get_clusivity

  Get grammatical clusivity.

Public Attributes

- grammaticalNumber
- grammaticalGender
- person
- anymacy
- clusivity
- tense
- case
- degree
- voice
- verbFormMood

6.32.1 Detailed Description

"Word Form is a Form subclass representing a form that a lexeme can take when used in a sentence or a phrase." (LMF)

Definition at line 12 of file word_form.py.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 def lmf.src.morphology.word_form.WordForm.__init__ (*self*)

Constructor.

WordForm instances are owned by LexicalEntry.

Returns

A WordForm instance.

Definition at line 15 of file word_form.py.

6.32.2.2 def lmf.src.morphology.word_form.WordForm.__del__ (*self*)

Destructor.

Definition at line 33 of file word_form.py.

6.32.3 Member Function Documentation

6.32.3.1 def lmf.src.morphology.word_form.WordForm.add_form_representation (*self, form_representation*)

Add a form representation to the word form.

Parameters

form_↩ representation	The FormRepresentation instance to add to the word form.

Returns

WordForm instance.

Definition at line 44 of file word_form.py.

6.32.3.2 def lmf.src.morphology.word_form.WordForm.create_form_representation (*self*)

Create a form representation.

Returns

FormRepresentation instance.

Definition at line 38 of file word_form.py.

6.32.3.3 def lmf.src.morphology.word_form.WordForm.get_anymacy (*self*)

Get anymacy.

Returns

WordForm attribute 'anymacy'.

Definition at line 148 of file word_form.py.

6.32.3.4 def lmf.src.morphology.word_form.WordForm.get_clusivity (*self*)

Get grammatical clusivity.

Returns

WordForm attribute 'clusivity'.

Definition at line 182 of file word_form.py.

6.32.3.5 def lmf.src.morphology.word_form.WordForm.get_form_representations (*self*)

Get all form representations maintained by the word form.

Returns

A Python list of form representations.

Definition at line 52 of file word_form.py.

6.32.3.6 def lmf.src.morphology.word_form.WordForm.get_grammaticalNumber (*self*)

Get grammatical number.

Returns

WordForm attribute 'grammaticalNumber'.

Definition at line 165 of file word_form.py.

6.32.3.7 def lmf.src.morphology.word_form.WordForm.get_person (*self*)

Get grammatical person.

Returns

WordForm attribute 'person'.

Definition at line 131 of file word_form.py.

6.32.3.8 def lmf.src.morphology.word_form.WordForm.get_variant_forms (*self*)

Get all variant forms.

This attribute is owned by FormRepresentation.

Returns

A Python list of FormRepresentation attributes 'variantForm'.

Definition at line 109 of file word_form.py.

6.32.3.9 def lmf.src.morphology.word_form.WordForm.get_written_forms (*self*, *script_name* = `None`)

Get all written forms.

This attribute is owned by Representation.

Parameters

script_name	If this argument is given, get written form only if written using this script.

Returns

A Python list of FormRepresentation attributes 'writtenForm'.

Definition at line 78 of file word_form.py.

6.32.3.10 def lmf.src.morphology.word_form.WordForm.set_anymacy (*self, anymacy*)

Set grammatical anymacy.

Parameters

anymacy	The grammatical anymacy to set.

Returns

WordForm instance.

Definition at line 137 of file word_form.py.

6.32.3.11 def lmf.src.morphology.word_form.WordForm.set_clusivity (*self, clusivity*)

Set grammatical clusivity.

Parameters

clusivity	The grammatical clusivity to set.

Returns

WordForm instance.

Definition at line 171 of file word_form.py.

6.32.3.12 def lmf.src.morphology.word_form.WordForm.set_grammaticalNumber (*self, grammatical_number*)

Set grammatical number.

Parameters

grammatical_↩ number	The grammatical number to set.

Returns

WordForm instance.

Definition at line 154 of file word_form.py.

6.32.3.13 def lmf.src.morphology.word_form.WordForm.set_person (*self, person*)

Set grammatical person.

Parameters

person	The grammatical person to set.

Returns

WordForm instance.

Definition at line 120 of file word_form.py.

6.32.3.14 def lmf.src.morphology.word_form.WordForm.set_variant_form (*self, variant_form*)

Set variant form.

This attribute is owned by FormRepresentation.

Parameters

variant_form	Variant form.

Returns

WordForm instance.

Definition at line 90 of file word_form.py.

6.32.3.15 def lmf.src.morphology.word_form.WordForm.set_written_form (*self, written_form, script_name* = `None`)

Set written form.

This attribute is owned by Representation.

Parameters

written_form	Written form.
script_name	Script used for the written form.

Returns

WordForm instance.

Definition at line 58 of file word_form.py.

6.32.4 Member Data Documentation

6.32.4.1 lmf.src.morphology.word_form.WordForm.anymacy

Definition at line 25 of file word_form.py.

6.32.4.2 lmf.src.morphology.word_form.WordForm.case

Definition at line 28 of file word_form.py.

6.32.4.3 lmf.src.morphology.word_form.WordForm.clusivity

Definition at line 26 of file word_form.py.

6.32.4.4 lmf.src.morphology.word_form.WordForm.degree

Definition at line 29 of file word_form.py.

6.32.4.5 lmf.src.morphology.word_form.WordForm.grammaticalGender

Definition at line 23 of file word_form.py.

6.32.4.6 lmf.src.morphology.word_form.WordForm.grammaticalNumber

Definition at line 22 of file word_form.py.

6.32.4.7 lmf.src.morphology.word_form.WordForm.person

Definition at line 24 of file word_form.py.

6.32.4.8 lmf.src.morphology.word_form.WordForm.tense

Definition at line 27 of file word_form.py.

6.32.4.9 lmf.src.morphology.word_form.WordForm.verbFormMood

Definition at line 31 of file word_form.py.

6.32.4.10 lmf.src.morphology.word_form.WordForm.voice

Definition at line 30 of file word_form.py.

The documentation for this class was generated from the following file:

- /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/word_form.py

Chapter 7

File Documentation

7.1 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/__init__.py File Reference

Namespaces

- lmf.src

7.2 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/__init__.py File Reference

Namespaces

- lmf.src.common

7.3 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/__init__.py File Reference

Namespaces

- lmf.src.config

7.4 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/__init__.py File Reference

Namespaces

- lmf.src.core

7.5 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/__init__.py File Reference

Namespaces

- lmf.src.input

7.6 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/__init__.py File Reference

Namespaces

- lmf.src.morphology

7.7 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphosyntax/__init__.py File Reference

Namespaces

- lmf.src.morphosyntax

7.8 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/__init__.py File Reference

Namespaces

- lmf.src.mrd

7.9 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/__init__.py File Reference

Namespaces

- lmf.src.output

7.10 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/__init__.py File Reference

Namespaces

- lmf.src.resources

7.11 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/__init__.py File Reference

Namespaces

- lmf.src.utils

7.12 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/ipa2sampa/__init__.py File Reference

Namespaces

- lmf.src.utils.ipa2sampa

7.13 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/defs.py File Reference

Namespaces

- lmf.src.common.defs

Variables

- string lmf.src.common.defs.VERNACULAR = "vernacular"

  Define languages.
- string lmf.src.common.defs.ENGLISH = "English"
- string lmf.src.common.defs.NATIONAL = "national"
- string lmf.src.common.defs.REGIONAL = "regional"

7.14 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/common/range.py File Reference

Namespaces

- lmf.src.common.range

Variables

- tuple lmf.src.common.range.partOfSpeech_range

  Possible values allowed for LMF part of speech LexicalEntry attribute.
- tuple lmf.src.common.range.type_variant_range

  Possible values allowed for LMF variant type FormRepresentation attribute.
- tuple lmf.src.common.range.noteType_range

  Possible values allowed for LMF note type Statement attribute.
- tuple lmf.src.common.range.grammaticalNumber_range

  Possible values allowed for LMF grammatical number WordForm attribute.
- tuple lmf.src.common.range.grammaticalGender_range

  Possible values allowed for LMF grammatical gender WordForm attribute.
- tuple lmf.src.common.range.person_range

  Possible values allowed for LMF grammatical person WordForm attribute.
- tuple lmf.src.common.range.anymacy_range

  Possible values allowed for LMF anymacy WordForm attribute.
- tuple lmf.src.common.range.clusivity_range

  Possible values allowed for LMF clusivity WordForm attribute.
- tuple lmf.src.common.range.tense_range

Possible values allowed for LMF grammatical tense WordForm attribute.

- tuple lmf.src.common.range.voice_range

  Possible values allowed for LMF voice WordForm attribute.
- tuple lmf.src.common.range.verbFormMood_range

  Possible values allowed for LMF verb form mood WordForm attribute.
- tuple lmf.src.common.range.degree_range

  Possible values allowed for LMF degree WordForm attribute.
- tuple lmf.src.common.range.semanticRelation_range

  Possible values allowed for semantic relation RelatedForm attribute.
- tuple lmf.src.common.range.paradigmLabel_range

  Possible values allowed for paradigm label Paradigm attribute.
- tuple lmf.src.common.range.type_example_range

  Possible values allowed for example type Context attribute.
- tuple lmf.src.common.range.mediaType_range

  Possible values allowed for media type Material attribute.
- tuple lmf.src.common.range.quality_range

  Possible values allowed for quality Audio attribute.

7.15 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/mdf.py File Reference

Namespaces

- lmf.src.config.mdf

Functions

- def lmf.src.config.mdf.set_bw

  Functions to process some MDF fields (input)
- def lmf.src.config.mdf.get_bw

  Functions to process some MDF fields (output)

Variables

- tuple lmf.src.config.mdf.mdf_lmf

  Mapping between MDF markers and LMF representation (input)
- list lmf.src.config.mdf.mdf_order

  Order in which MDF markers must be written (output) This is the standard order defined in Appendix B of "Making Dictionaries. A guide to lexicography and the Multi-Dictionary Formatter", Software version 1.0, David F.
- tuple lmf.src.config.mdf.lmf_mdf

  Mapping between LMF representation and MDF markers (output)
- tuple lmf.src.config.mdf.ps_range

  Possible values allowed for 'ps' MDF marker.
- tuple lmf.src.config.mdf.ps_partOfSpeech

  Mapping between 'ps' MDF marker value and LMF part of speech LexicalEntry attribute value (input) Source: http://www.isocat.org/rest/dcs/119.
- tuple lmf.src.config.mdf.mdf_semanticRelation

  Mapping between MDF markers and LMF semantic relation RelatedForm attribute value (input)
- tuple lmf.src.config.mdf.pd_person

Mapping between 'pd' MDF markers and LMF person WordForm attribute value (input)

- tuple lmf.src.config.mdf.pd_anymacy

  Mapping between 'pd' MDF markers and LMF anymacy WordForm attribute value (input)
- tuple lmf.src.config.mdf.pd_grammaticalNumber

  Mapping between 'pd' MDF markers and LMF grammatical number WordForm attribute value (input)
- tuple lmf.src.config.mdf.pd_clusivity

  Mapping between 'pd' MDF markers and LMF clusivity WordForm attribute value (input)
- tuple lmf.src.config.mdf.pdl_paradigmLabel

  Mapping between 'pdl' MDF marker value and LMF paradigm label Paradigm attribute value (input)
- tuple lmf.src.config.mdf.sd_range

  Possible values allowed for 'sd' MDF marker.
- tuple lmf.src.config.mdf.lf_range

  Possible values allowed for 'lf' MDF marker.

7.16 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/mdf.py File Reference

Namespaces

- lmf.src.input.mdf

Functions

- def lmf.src.input.mdf.mdf_read

  Read an MDF file.

7.17 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/mdf.py File Reference

Namespaces

- lmf.src.output.mdf

Functions

- def lmf.src.output.mdf.mdf_write

  Write an MDF file.
- def lmf.src.output.mdf.parse_list

  Parse a group of markers and write them into an MDF file.
- def lmf.src.output.mdf.write_line

  Write a line into an MDF file.

7.18 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/tex.py File Reference

Namespaces

- lmf.src.config.tex

Functions

- def lmf.src.config.tex.lmf_to_tex

  Function giving order in which information must be written in LaTeX and mapping between LMF representation and LaTeX (output)

Variables

- tuple lmf.src.config.tex.partOfSpeech_tex

  Mapping between LMF part of speech LexicalEntry attribute value and LaTeX layout (output)
- tuple lmf.src.config.tex.paradigmLabel_tex

  Mapping between LMF paradigmLabel Paradigm attribute value and LaTeX layout (output)

7.19 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/tex.py File Reference

Namespaces

- lmf.src.output.tex

Functions

- def lmf.src.output.tex.file_read

  Read file contents.
- def lmf.src.output.tex.insert_references

  Insert references to paradigms.
- def lmf.src.output.tex.tex_write

  Write a LaTeX file.
- def lmf.src.output.tex.handle_font

  Functions to process LaTeX layout.
- def lmf.src.output.tex.handle_reserved
- def lmf.src.output.tex.handle_fi
- def lmf.src.output.tex.handle_fv
- def lmf.src.output.tex.handle_fn
- def lmf.src.output.tex.handle_pinyin
- def lmf.src.output.tex.handle_caps
- def lmf.src.output.tex.handle_quotes
- def lmf.src.output.tex.format_uid

  Functions to process LaTeX fields (output)
- def lmf.src.output.tex.format_link

  Display hyperlink to a lexical entry in LaTeX format.
- def lmf.src.output.tex.format_lexeme

  'lx', 'hm' and 'lc' fields are flipped if 'lc' field has data.
- def lmf.src.output.tex.format_audio

  Embed sound file into PDF.
- def lmf.src.output.tex.format_part_of_speech

  Display part of speech in LaTeX format.
- def lmf.src.output.tex.format_definitions

  Glosses are supplanted by definitions.
- def lmf.src.output.tex.format_lt

Display 'lt' in LaTeX format.

- def lmf.src.output.tex.format_sc

  Display 'sc' in LaTeX format.
- def lmf.src.output.tex.format_rf

  Display 'rf' in LaTeX format.
- def lmf.src.output.tex.format_examples

  Display examples in LaTeX format.
- def lmf.src.output.tex.format_usage_notes

  Display usage notes in LaTeX format.
- def lmf.src.output.tex.format_encyclopedic_informations

  Display encyclopedic informations in LaTeX format.
- def lmf.src.output.tex.format_restrictions

  Display restrictions in LaTeX format.
- def lmf.src.output.tex.format_lexical_functions

  Display lexical functions in LaTeX format.
- def lmf.src.output.tex.format_related_forms

  Display related forms in LaTeX format.
- def lmf.src.output.tex.format_variant_forms

  Display variant forms in LaTeX format.
- def lmf.src.output.tex.format_borrowed_word

  Display borrowed word in LaTeX format.
- def lmf.src.output.tex.format_etymology

  Display etymology in LaTeX format.
- def lmf.src.output.tex.format_paradigms

  Display all paradigms in LaTeX format.
- def lmf.src.output.tex.format_table

  Display a table in LaTeX format.
- def lmf.src.output.tex.format_semantic_domains

  Display semantic domains in LaTeX format.
- def lmf.src.output.tex.format_bibliography

  Display bibliography in LaTeX format.
- def lmf.src.output.tex.format_picture

  Display a picture in LaTeX format.
- def lmf.src.output.tex.format_notes

  Display all notes in LaTeX format.
- def lmf.src.output.tex.format_source

  Display source in LaTeX format.
- def lmf.src.output.tex.format_status

  Display status in LaTeX format.
- def lmf.src.output.tex.format_date

  Do not display date in LaTeX format.

7.20 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/config/xml.py File Reference

Namespaces

- lmf.src.config.xml

Functions

- def lmf.src.config.xml.sort_order_read

  Read an XML file giving sort order.
- def lmf.src.config.xml.config_read

  Read an XML file giving the user configuration.

7.21 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/definition.py File Reference

Classes

- class lmf.src.core.definition.Definition

  "Definition is a class representing a narrative description of a sense. It is provided to help human users understand the meaning of a lexical entry. A Sense instance can have zero to many definitions. Each Definition instance may be associated with zero to many Text Representation instances in order to manage the text defintion in more than one language or script. In addition, the narrative description can be expressed in a different language or script than the one in the Lexical Entry instance." (LMF)

Namespaces

- lmf.src.core.definition

7.22 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form.py File Reference

Classes

- class lmf.src.core.form.Form

  "Form is an abstract class representing a lexeme, a morphological variant of a lexeme or a morph. The Form class allows subclasses." (LMF)

Namespaces

- lmf.src.core.form

7.23 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/form_representation.py File Reference

Classes

- class lmf.src.core.form_representation.FormRepresentation

  "Form Representation is a class representing one variant orthography of a Form." (LMF)

Namespaces

- lmf.src.core.form_representation

7.24 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/global_information.py File Reference

Classes

- class lmf.src.core.global_information.GlobalInformation

  "Global Information is a class for administrative information and other general attributes, such as /language coding/ or /script coding/, which are valid for the entire lexical resource." (LMF)

Namespaces

- lmf.src.core.global_information

7.25 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_entry.py File Reference

Classes

- class lmf.src.core.lexical_entry.LexicalEntry

  "Lexical Entry is a class representing a lexeme in a given language and is a container for managing the Form and Sense classes. A Lexical Entry instance can contain one to many different forms and can have from zero to many different senses." (LMF)

Namespaces

- lmf.src.core.lexical_entry

7.26 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexical_resource.py File Reference

Classes

- class lmf.src.core.lexical_resource.LexicalResource

  "Lexical Resource is a class representing the entire resource and is a container for one or more lexicons. There is only one Lexical Resource instance." (LMF)

Namespaces

- lmf.src.core.lexical_resource

7.27 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/lexicon.py File Reference

Classes

- class lmf.src.core.lexicon.Lexicon

  "Lexicon is a class containing all the lexical entries of a given language within the entire resource." (LMF)

Namespaces

- lmf.src.core.lexicon

7.28 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/representation.py File Reference

Classes

- class lmf.src.core.representation.Representation

  "Representation class is an abstract class representing a Unicode string as well as, if needed, the unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Namespaces

- lmf.src.core.representation

7.29 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/sense.py File Reference

Classes

- class lmf.src.core.sense.Sense

  "Sense is a class representing one meaning of a lexical entry. The Sense class allows for hierarchical senses in that a sense may be more specific than another sense of the same lexical entry." (LMF)

Namespaces

- lmf.src.core.sense

7.30 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/statement.py File Reference

Classes

- class lmf.src.core.statement.Statement

  "Statement is a class representating a narrative description that refines or complements Definition." (LMF)

Namespaces

- lmf.src.core.statement

7.31 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/core/text_representation.py File Reference

Classes

- class lmf.src.core.text_representation.TextRepresentation

  "Text Representation is a class representing the textual content of definition or statement. When there is more than one variant orthography, the Text Representation instance contains a Unicode string representing the textual content as well as unique attribute-value pairs that describe the specific language, script and orthography." (LMF)

Namespaces

- lmf.src.core.text_representation

7.32 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/elan.py File Reference

Namespaces

- lmf.src.input.elan

7.33 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/ite.py File Reference

Namespaces

- lmf.src.input.ite

7.34 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/toolbox_settings.py File Reference

Namespaces

- lmf.src.input.toolbox_settings

7.35 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/txt.py File Reference

Namespaces

- lmf.src.input.txt

7.36 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/txt.py File Reference

Namespaces

- lmf.src.output.txt

7.37 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/xls.py File Reference

Namespaces

- lmf.src.input.xls

7.38 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xls.py File Reference

Namespaces

- lmf.src.output.xls

7.39 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/input/xml_lmf.py File Reference

Namespaces

- lmf.src.input.xml_lmf

Functions

- def lmf.src.input.xml_lmf.compute_name

  Compute attribute/module name from object name as follows: 'ObjectName' attribute/module name is 'object_name'.
- def lmf.src.input.xml_lmf.factory

  This function is an object factory.
- def lmf.src.input.xml_lmf.xml_lmf_read

  Read an XML LMF file.
- def lmf.src.input.xml_lmf.get_sub_elements

  This function recursively parses the given XML element and creates corresponding LMF instances with their attributes.

7.40 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lmf.py File Reference

Namespaces

- lmf.src.output.xml_lmf

Functions

- def lmf.src.output.xml_lmf.xml_lmf_write

  Write an XML LMF file.
- def lmf.src.output.xml_lmf.build_sub_elements

  Create XML sub-elements to an existing XML element by parsing an LMF object instance.
- def lmf.src.output.xml_lmf.add_link

  Functions to process XML/XHTML layout.

- def lmf.src.output.xml_lmf.handle_reserved
- def lmf.src.output.xml_lmf.handle_fv
- def lmf.src.output.xml_lmf.handle_fn
- def lmf.src.output.xml_lmf.handle_font
- def lmf.src.output.xml_lmf.handle_pinyin
- def lmf.src.output.xml_lmf.handle_caps
- def lmf.src.output.xml_lmf.handle_tones

7.41 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/component.py File Reference

Classes

- class lmf.src.morphology.component.Component

Namespaces

- lmf.src.morphology.component

7.42 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/lemma.py File Reference

Classes

- class lmf.src.morphology.lemma.Lemma

  "Lemma is a Form subclass representing a form chosen by convention to designate the Lexical Entry. The lemma is usually equivalent to one of the inflected forms, the root, stem or compound phrase." (LMF).

Namespaces

- lmf.src.morphology.lemma

7.43 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/list_of_components.py File Reference

Classes

- class lmf.src.morphology.list_of_components.ListOfComponents

Namespaces

- lmf.src.morphology.list_of_components

7.44 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/related_form.py File Reference

Classes

- class lmf.src.morphology.related_form.RelatedForm

  "Related Form is a Form subclass representing a word form or a morph that can be related to the Lexical Entry. There is no asumption that the Related Form is associated with the Sense class in the Lexical Entry." (LMF)

Namespaces

- lmf.src.morphology.related_form

7.45 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/stem.py File Reference

Classes

- class lmf.src.morphology.stem.Stem

  "Stem is a Form subclass representing a morph, thus manages the sublexme parts" (LMF)

Namespaces

- lmf.src.morphology.stem

7.46 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphology/word_form.py File Reference

Classes

- class lmf.src.morphology.word_form.WordForm

  "Word Form is a Form subclass representing a form that a lexeme can take when used in a sentence or a phrase." (LMF)

Namespaces

- lmf.src.morphology.word_form

7.47 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/morphosyntax/paradigm.py File Reference

Classes

- class lmf.src.morphosyntax.paradigm.Paradigm

  Paradigm is a class representing a morphological paradigm.

Namespaces

- lmf.src.morphosyntax.paradigm

7.48 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/context.py File Reference

Classes

- class lmf.src.mrd.context.Context

  "Context is a class representing a text string that provides authentic context for the use of the word form managed by the Lemma. This class is to be distinguished from Sense Example." (LMF)

Namespaces

- lmf.src.mrd.context

7.49 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/equivalent.py File Reference

Classes

- class lmf.src.mrd.equivalent.Equivalent

  "Equivalent is a class representing the translation equivalent of the word form managed by the Lemma class." (LMF)

Namespaces

- lmf.src.mrd.equivalent

7.50 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/mrd/subject_field.py File Reference

Classes

- class lmf.src.mrd.subject_field.SubjectField

  "Subject Field is a class representing a text string that provides domain or status information." (LMF)

Namespaces

- lmf.src.mrd.subject_field

7.51 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/doc.py File Reference

Namespaces

- lmf.src.output.doc

Functions

- def lmf.src.output.doc.doc_write

  Write a document file.

7.52 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/html.py File Reference

Namespaces

- lmf.src.output.html

7.53 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/odt.py File Reference

Namespaces

- lmf.src.output.odt

Variables

- tuple lmf.src.output.odt.textdoc = OpenDocumentText()
- lmf.src.output.odt.s = textdoc.styles
- tuple lmf.src.output.odt.h1style = Style(name="Heading 1", family="paragraph")
- tuple lmf.src.output.odt.boldstyle = Style(name="Bold", family="text")
- tuple lmf.src.output.odt.boldprop = TextProperties(fontweight="bold", fontname="Arial", fontsize="8pt")
- tuple lmf.src.output.odt.h = H(outlinelevel=1, stylename=h1style, text="My first text")
- tuple lmf.src.output.odt.p = P(text="Hello world. ")
- tuple lmf.src.output.odt.boldpart = Span(stylename=boldstyle, text="This part is bold. ")

7.54 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_ite.py File Reference

Namespaces

- lmf.src.output.xml_ite

7.55 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lift.py File Reference

Namespaces

- lmf.src.output.xml_lift

7.56 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_lp.py File Reference

Namespaces

- lmf.src.output.xml_lp

7.57 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_olif.py File Reference

Namespaces

- lmf.src.output.xml_olif

7.58 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_tb.py File Reference

Namespaces

- lmf.src.output.xml_tb

7.59 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/output/xml_tei.py File Reference

Namespaces

- lmf.src.output.xml_tei

7.60 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/audio.py File Reference

Classes

- class lmf.src.resources.audio.Audio

  Audio is a Material subclass representing an audio recording.

Namespaces

- lmf.src.resources.audio

7.61 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/human_resource.py File Reference

Classes

- class lmf.src.resources.human_resource.HumanResource

HumanResource *is a Resource subclass.*

Namespaces

- lmf.src.resources.human_resource

7.62 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/material.py File Reference

Classes

- class lmf.src.resources.material.Material

  Material *is a Resource subclass.*

Namespaces

- lmf.src.resources.material

7.63 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/picture.py File Reference

Classes

- class lmf.src.resources.picture.Picture

  Picture *is a Material subclass representing a picture.*

Namespaces

- lmf.src.resources.picture

7.64 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/resource.py File Reference

Classes

- class lmf.src.resources.resource.Resource

  Resource *is an abstract class representing a material or a human resource.*

Namespaces

- lmf.src.resources.resource

7.65 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/speaker.py File Reference

Classes

- class lmf.src.resources.speaker.Speaker

  Speaker is a HumanResource subclass.

Namespaces

- lmf.src.resources.speaker

7.66 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/resources/video.py File Reference

Classes

- class lmf.src.resources.video.Video

  Video is a Material subclass representing a video.

Namespaces

- lmf.src.resources.video

7.67 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/attr.py File Reference

Namespaces

- lmf.src.utils.attr

Functions

- def lmf.src.utils.attr.check_attr_type

  Check that attribute value is of specified type.
- def lmf.src.utils.attr.check_attr_range

  Check that attribute value is in specified range.
- def lmf.src.utils.attr.check_date_format

  Verify that date format is composed as follows: YYYY-MM-DD (ISO 8601).
- def lmf.src.utils.attr.check_time_format

  Verify that time format is composed as follows: THH:MM:SS,MSMS (ISO 8601: 'T' for Time).
- def lmf.src.utils.attr.check_duration_format

  Verify that duration format is composed as follows: PTxxHxxMxxS (ISO 8601: 'P' for Period).

7.68 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/eol/eol.py File Reference

Namespaces

- eol

Variables

- tuple eol.parser = OptionParser()
- tuple eol.options = parser.parse_args()
- tuple eol.in_file = open(options.input, "r", encoding='utf-8')
- tuple eol.out_file = open(options.output, "w", encoding='utf-8')
- string eol.EOL = '\n'
- list eol.lines = []
- tuple eol.previous_line = lines.pop()
- tuple eol.line = previous_line.replace(EOL, " ")

7.69 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/error_handling.py File Reference

Classes

- class lmf.src.utils.error_handling.Error

  Base class for exceptions in this library.
- class lmf.src.utils.error_handling.InputError

  Exception raised for errors in the input.
- class lmf.src.utils.error_handling.OutputError

  Exception raised for errors in the output.
- class lmf.src.utils.error_handling.Warning

  Base class for warnings in this library.

Namespaces

- lmf.src.utils.error_handling

7.70 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/io.py File Reference

Namespaces

- lmf.src.utils.io

Functions

- def lmf.src.utils.io.open_file

  Open file in specified mode (automatically decode file in unicode).
- def lmf.src.utils.io.open_read

Open file in read mode (automatically decode file in unicode).

- def lmf.src.utils.io.open_write

  Open file in write mode (automatically decode file in unicode).

Variables

- string lmf.src.utils.io.EOL = '\n'
- string lmf.src.utils.io.ENCODING = 'utf-8'

7.71 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/ipa2sampa/ipa2sampa.py File Reference

Namespaces

- lmf.src.utils.ipa2sampa.ipa2sampa

Functions

- def lmf.src.utils.ipa2sampa.ipa2sampa.uni2sampa

Variables

- tuple lmf.src.utils.ipa2sampa.ipa2sampa.data = codecs.open('./src/utils/ipa2sampa/sampa.csv', 'r', 'utf-8')
- list lmf.src.utils.ipa2sampa.ipa2sampa.sota = []
- tuple lmf.src.utils.ipa2sampa.ipa2sampa.ta = eval('"""'+ta+'"""')
- tuple lmf.src.utils.ipa2sampa.ipa2sampa.seq = line.strip()

7.72 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/log.py File Reference

Namespaces

- lmf.src.utils.log

Functions

- def lmf.src.utils.log.log

  Write message into log file if any, or to standard output if verbose mode is on.

7.73 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/tables/tables.py File Reference

Namespaces

- tables

Variables

- tuple tables.parser = OptionParser()
- tuple tables.options = parser.parse_args()
- tuple tables.in_file = open(options.input, "r", encoding='utf-8')
- tuple tables.out_eng = open(options.output_eng, "w", encoding='utf-8')
- tuple tables.out_fra = open(options.output_fra, "w", encoding='utf-8')
- string tables.EOL = '\n'
- string tables.title_eng = """Words for which no close equivalent could be found"""
- string tables.introduction_eng = """The list that follows groups words for which no close equivalents could be found. These negative pieces of information contain hints about the consultants' Na vocabulary and its 'soft shoulders'."""
- string tables.title_fra = """Mots dont aucun équivalent n'a été trouvé"""
- string tables.introduction_fra = """Cette liste regroupe les mots dont aucun équivalent n'a été trouvé. Même s'il ne s'agit que d'informations négatives, elles éclairent les limites du vocabulaire na des consultants."""
- string tables.pattern = r"^\\(\w{2,3}) ?(.*)$"
- string tables.lx = ""
- string tables.ge = ""
- string tables.gn = ""
- string tables.gf = ""
- tuple tables.result = re.search(pattern, line)

7.74 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/uid/uid.py File Reference

Namespaces

- uid

Variables

- tuple uid.parser = OptionParser()
- tuple uid.options = parser.parse_args()
- tuple uid.in_file = open(options.input, "r", encoding='utf-8')
- tuple uid.out_file = open(options.output, "w", encoding='utf-8')
- string uid.EOL = '\n'
- string uid.pattern = r"^\\(\w{2,3}) ?(.*)$"
- string uid.lx = ""
- string uid.mkr = "lx"
- list uid.sf = []
- string uid.hm = ""
- tuple uid.result = re.search(pattern, line)
- tuple uid.uid = uni2sampa(lx)

7.75 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/utils/xml_format.py File Reference

Namespaces

- lmf.src.utils.xml_format

Functions

- def lmf.src.utils.xml_format.prettify

  Return a pretty-printed XML string for the given XML element.
- def lmf.src.utils.xml_format.write_result

  Write an XML element into a pretty XML output file.
- def lmf.src.utils.xml_format.parse_xml

  Parse an XML file.

7.76 /Users/celine/Work/CNRS/workspace/HimalCo/dev/lib/lmf/src/wrapper.py File Reference

Namespaces

- lmf.src.wrapper

Functions

- def lmf.src.wrapper.wrapper_rw

  Wrapper function that calls another function, restoring normal behavior on error.
- def lmf.src.wrapper.read_mdf
- def lmf.src.wrapper.read_xml_lmf
- def lmf.src.wrapper.read_sort_order
- def lmf.src.wrapper.read_config
- def lmf.src.wrapper.write_mdf
- def lmf.src.wrapper.write_xml_lmf
- def lmf.src.wrapper.write_tex
- def lmf.src.wrapper.write_doc

Variables

- lmf.src.wrapper.lexical_resource = None

  Module variable.