JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/LexicalFeature.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/features/InterrogativeType.html)
* [Next Class](http://docs.google.com/simplenlg/features/NumberAgreement.html)
* [Frames](http://docs.google.com/index.html?simplenlg/features/LexicalFeature.html)
* [No Frames](http://docs.google.com/LexicalFeature.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#id.3znysh7)|
* Constr |
* [Method](#id.2et92p0)
* Detail:
* [Field](#id.3dy6vkm)|
* Constr |
* [Method](#id.ihv636)

simplenlg.features

## Class LexicalFeature

* java.lang.Object
  + simplenlg.features.LexicalFeature
* public abstract class LexicalFeature  
  extends java.lang.Object
* This class defines a list of constant values used by SimpleNLG lexicons. Note that there are three feature classes in SimpleNLG.
  + Feature: features typically set up developers invoking SimpleNLG
  + LexicalFeature: features typically set up the SimpleNLG lexicon
  + InternalFeature: features typically used internally by SimpleNLG
* Elements in the system can, in theory, take any kind of feature. Some features will only be expected by certain processors, however. Developers can define their own features but should choose names that do not conflict with those presented here.  
    
  The details for each feature are supplied in a table. The entries are:

|  |  |
| --- | --- |
| * **Feature name** | * This is the name that will appear in the element's feature list as produced by the toString() method or by calling getAllFeatureNames |
| * **Expected type** | * As features are represented as a Map connecting a String and an Object then, in theory, a feature can take any object as a value. This table entry defines the type that the SimpleNLG system expects. |
| * **Created by** | * Defines where the feature is created. In addition, all features can be added specifically by users |
| * **Used by** | * Defines which processors use the feature. |
| * **Applies to** | * Defines which structural, syntactical or lexical elements this feature is applied to. |
| * **Default** | * Any default values attributed to the feature are given here. |
|  |

Version: 4.0 Author: E. Reiter and D. Westwater, University of Aberdeen.

### Field SummaryFields

|  |  |
| --- | --- |
| * + Modifier and Type | * + Field and Description |
| * + static java.lang.String | * + [**ACRONYM\_OF**](http://docs.google.com/simplenlg/features/LexicalFeature.html#ACRONYM_OF) This feature is used to map an acronym element to the full forms of the acronym. |
| * + static java.lang.String | * + [**ACRONYMS**](http://docs.google.com/simplenlg/features/LexicalFeature.html#ACRONYMS) This feature is used to map a word to its acronyms. |
| * + static java.lang.String | * + [**BASE\_FORM**](http://docs.google.com/simplenlg/features/LexicalFeature.html#BASE_FORM) This feature is used to define the base form for phrases and words. |
| * + static java.lang.String | * + [**CLASSIFYING**](http://docs.google.com/simplenlg/features/LexicalFeature.html#CLASSIFYING) This feature is used for determining the position of adjectives. |
| * + static java.lang.String | * + [**COLOUR**](http://docs.google.com/simplenlg/features/LexicalFeature.html#COLOUR) This feature is used for determining the position of adjectives. |
| * + static java.lang.String | * + [**COMPARATIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#COMPARATIVE) This feature gives the comparative form for adjectives and adverbs. |
| * + static java.lang.String | * + [**DEFAULT\_INFL**](http://docs.google.com/simplenlg/features/LexicalFeature.html#DEFAULT_INFL) This feature is used to specify, for a given word, what its default inflectional variant is, if more than one is possible. |
| * + static java.lang.String | * + [**DEFAULT\_SPELL**](http://docs.google.com/simplenlg/features/LexicalFeature.html#DEFAULT_SPELL) This feature is used to specify the default spelling variant of a word, if it has more than one. |
| * + static java.lang.String | * + [**DITRANSITIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#DITRANSITIVE) This feature determines if a verb is ditransitive, meaning that it can have a subject, direct object and indirect object. |
| * + static java.lang.String | * + [**EXPLETIVE\_SUBJECT**](http://docs.google.com/simplenlg/features/LexicalFeature.html#EXPLETIVE_SUBJECT) This feature determines if the pronoun is an expletive or not. |
| * + static java.lang.String | * + [**GENDER**](http://docs.google.com/simplenlg/features/LexicalFeature.html#GENDER) This feature determines whether a noun is masculine, feminine or neuter in nature. |
| * + static java.lang.String | * + [**INTENSIFIER**](http://docs.google.com/simplenlg/features/LexicalFeature.html#INTENSIFIER) This flag determines if an adverb is an intensifier, such as *very*. |
| * + static java.lang.String | * + [**INTRANSITIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#INTRANSITIVE) This flag highlights a verb that can only take a subject and no objects. |
| * + static java.lang.String | * + [**PAST**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PAST) This feature gives the past tense form of a verb. |
| * + static java.lang.String | * + [**PAST\_PARTICIPLE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PAST_PARTICIPLE) This feature gives the past participle tense form of a verb. |
| * + static java.lang.String | * + [**PLURAL**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PLURAL) This feature gives the plural form of a noun. |
| * + static java.lang.String | * + [**PREDICATIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PREDICATIVE) This flag is set on adjectives that can also be used as a predicate. |
| * + static java.lang.String | * + [**PRESENT\_PARTICIPLE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PRESENT_PARTICIPLE) This feature gives the present participle form of a verb. |
| * + static java.lang.String | * + [**PRESENT3S**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PRESENT3S) This feature gives the present third person singular form of a verb. |
| * + static java.lang.String | * + [**PROPER**](http://docs.google.com/simplenlg/features/LexicalFeature.html#PROPER) This flag is used to determine whether a noun is a proper noun, such as a person's name. |
| * + static java.lang.String | * + [**QUALITATIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#QUALITATIVE) This feature is used for determining the position of adjectives. |
| * + static java.lang.String | * + [**REFLEXIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#REFLEXIVE) This flag is set if a pronoun is written in the reflexive form. |
| * + static java.lang.String | * + [**SENTENCE\_MODIFIER**](http://docs.google.com/simplenlg/features/LexicalFeature.html#SENTENCE_MODIFIER) This feature is used to define whether an adverb can be used as a clause modifier, which are normally applied at the beginning of clauses. |
| * + static java.lang.String | * + [**SPELL\_VARS**](http://docs.google.com/simplenlg/features/LexicalFeature.html#SPELL_VARS) This feature is used to specify the spelling variants of a word. |
| * + static java.lang.String | * + [**SUPERLATIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#SUPERLATIVE) This feature gives the superlative form for adjectives and adverbs. |
| * + static java.lang.String | * + [**TRANSITIVE**](http://docs.google.com/simplenlg/features/LexicalFeature.html#TRANSITIVE) This flag highlights a verb that can only take a subject and an object. |
| * + static java.lang.String | * + [**VERB\_MODIFIER**](http://docs.google.com/simplenlg/features/LexicalFeature.html#VERB_MODIFIER) This feature is used to define whether an adverb can be used as a verb modifier, which are normally added in a phrase before the verb itself. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + static java.lang.String[] | * + [**getInflectionalFeatures**](http://docs.google.com/simplenlg/features/LexicalFeature.html#getInflectionalFeatures(simplenlg.framework.ElementCategory))([ElementCategory](http://docs.google.com/simplenlg/framework/ElementCategory.html) cat) Return those features related to a word's inflection, depending on its category, that is, the constants for PAST, PAST\_PARTICIPLE, PLURAl, PRESENT\_PARTICIPLE, PRESENT3S, COMPARATIVE or SUPERLATIVE. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### ACRONYM\_OF public static final java.lang.String ACRONYM\_OF This feature is used to map an acronym element to the full forms of the acronym.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *acronymOf* |
| * + - **Expected type** | * + - List |
| * + - **Created by** | * + - Lexicons that support acronyms should set this feature. |
| * + - **Used by** | * + - No processors currently use this feature. |
| * + - **Applies to** | * + - Any lexical item. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.ACRONYM_OF)

#### ACRONYMS public static final java.lang.String ACRONYMS This feature is used to map a word to its acronyms.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *acronyms* |
| * + - **Expected type** | * + - List |
| * + - **Created by** | * + - Lexicons that support acronyms should set this feature. |
| * + - **Used by** | * + - No processors currently use this feature. |
| * + - **Applies to** | * + - Any lexical item. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.ACRONYMS)

#### DEFAULT\_INFL public static final java.lang.String DEFAULT\_INFL This feature is used to specify, for a given word, what its default inflectional variant is, if more than one is possible.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *default\_infl* |
| * + - **Expected type** | * + - Inflection |
| * + - **Created by** | * + - Lexicons that support multiple inflectional variants should set this feature. |
| * + - **Used by** | * + - [MorphologyProcessor](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html). |
| * + - **Applies to** | * + - Nouns and verbs. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.DEFAULT_INFL)

#### SPELL\_VARS public static final java.lang.String SPELL\_VARS This feature is used to specify the spelling variants of a word.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *spell\_vars* |
| * + - **Expected type** | * + - List |
| * + - **Created by** | * + - Lexicons that support multiple spelling variants should set this feature. |
| * + - **Used by** | * + - No processors currently use this feature. |
| * + - **Applies to** | * + - Any lexical item. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.SPELL_VARS)

#### DEFAULT\_SPELL public static final java.lang.String DEFAULT\_SPELL This feature is used to specify the default spelling variant of a word, if it has more than one.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *default\_spell* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - Lexicons that support multiple spelling variants should set this feature. |
| * + - **Used by** | * + - [MorphologyProcessor](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html) |
| * + - **Applies to** | * + - Any lexical item. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.DEFAULT_SPELL)

#### BASE\_FORM public static final java.lang.String BASE\_FORM This feature is used to define the base form for phrases and words.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *baseForm* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - The lexicon accessor also creates the feature when looking up words in the lexicon. Sometimes the phrase factory sets this feature as well, as an approximate realisation for debuggin purposes |
| * + - **Used by** | * + - The morphology processor uses the base form in its simple rules for determining word inflection. The morphology processor and syntax processor also use the base form for lexicon look ups if the base word has not been set. Base forms on phrases are purely to aid debugging. |
| * + - **Applies to** | * + - Phrases and words. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.BASE_FORM)

#### CLASSIFYING public static final java.lang.String CLASSIFYING This feature is used for determining the position of adjectives. Setting this value to true means that the adjective can occupy the *classifying* position.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *classifying* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon that supports adjective positioning. |
| * + - **Used by** | * + - The syntax processor to determine the ordering of adjectives. |
| * + - **Applies to** | * + - Adjectives within noun phrases. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.CLASSIFYING)

#### COLOUR public static final java.lang.String COLOUR This feature is used for determining the position of adjectives. Setting this value to true means that the adjective can occupy the *colour* position.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *colour* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon that supports adjective positioning. |
| * + - **Used by** | * + - The syntax processor to determine the ordering of adjectives. |
| * + - **Applies to** | * + - Adjectives within noun phrases. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.COLOUR)

#### COMPARATIVE public static final java.lang.String COMPARATIVE This feature gives the comparative form for adjectives and adverbs. For example, *dizzier* is the comparative form of *dizzy*, *fatter* is the comparative form of *fat* and *earlier* is the comparative form of *early*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *comparative* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - Can be created automatically by the lexicon or added manually by users. |
| * + - **Used by** | * + - The morphology processor uses this information to correctly inflect words. |
| * + - **Applies to** | * + - Adjectives and adverbs only. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.COMPARATIVE)

#### DITRANSITIVE public static final java.lang.String DITRANSITIVE This feature determines if a verb is ditransitive, meaning that it can have a subject, direct object and indirect object. For example in the phrase *he gave Mary ten pounds*, the verb *give* has three components: the subject is the person doing the giving (*he*), the direct object is the object being passed (*ten pounds*) and the indirect object is the recipient (*Mary*).

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *ditransitive* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - The feature is set by the lexicon if it supports the recording of the transitive nature of verbs. |
| * + - **Used by** | * + - The ditransitive value is currently not used. |
| * + - **Applies to** | * + - Verbs only. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.DITRANSITIVE)

#### GENDER public static final java.lang.String GENDER This feature determines whether a noun is masculine, feminine or neuter in nature.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *gender* |
| * + - **Expected type** | * + - Gender |
| * + - **Created by** | * + - The phrase factory creates the gender of pronouns when creating phrases and on all nouns within a noun phrase. |
| * + - **Used by** | * + - The syntax processor ensures that the head noun in a noun phrase has a gender matching that applied to the phrase as a whole. The morphology processor uses gender to determine the appropriate form for pronouns and for setting the form of some verbs. |
| * + - **Applies to** | * + - Specifically it applies to nouns and pronouns but the feature is also written to noun phrases and verbs. |
| * + - **Default** | * + - Gender.NEUTER |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.GENDER)

#### INTENSIFIER public static final java.lang.String INTENSIFIER This flag determines if an adverb is an intensifier, such as *very*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *intensifier* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - The information is read from Lexicons that support this feature. |
| * + - **Used by** | * + - Currently not used. |
| * + - **Applies to** | * + - Adverbs only. |
| * + - **Default** | * + - Boolean.FALSE. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.INTENSIFIER)

#### INTRANSITIVE public static final java.lang.String INTRANSITIVE This flag highlights a verb that can only take a subject and no objects.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *intransitive* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - The information is read from Lexicons that support this feature. |
| * + - **Used by** | * + - Currently not used. |
| * + - **Applies to** | * + - Verbs only. |
| * + - **Default** | * + - Boolean.FALSE. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.INTRANSITIVE)

#### PAST public static final java.lang.String PAST This feature gives the past tense form of a verb. For example, the past tense of *eat* is *ate*, the past tense of *walk* is *walked*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *past* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - All supporting lexicons but can be set by the user for irregular cases. |
| * + - **Used by** | * + - The morphology processor uses this feature to correctly inflect verbs. This feature will be looked at first before any reference to lexicons or morphology rules. |
| * + - **Applies to** | * + - Verbs and verb phrases only. |
| * + - **Default** | * + - null. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PAST)

#### PAST\_PARTICIPLE public static final java.lang.String PAST\_PARTICIPLE This feature gives the past participle tense form of a verb. For many verbs the past participle is exactly the same as the past tense, for example, the verbs *talk*, *walk* and *say* have past tense and past participles of *talked*, *walked* and *said*. Contrast this with the verbs *do*, *eat* and *sing*. The past tense of these verbs is *did*, *ate* and *sang* respectively. while the respective past participles are *done*, *eaten* and *sung*

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *pastParticiple* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - All supporting lexicons but can be set by the user for irregular cases. |
| * + - **Used by** | * + - The morphology processor uses this feature to correctly inflect verbs. This feature will be looked at first before any reference to lexicons or morphology rules. |
| * + - **Applies to** | * + - Verbs and verb phrases only. |
| * + - **Default** | * + - null. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PAST_PARTICIPLE)

#### PLURAL public static final java.lang.String PLURAL This feature gives the plural form of a noun. For example, the plural of *dog* is *dogs* and the plural of *sheep* is *sheep*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *plural* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - All supporting lexicons but can be set by the user for irregular cases. |
| * + - **Used by** | * + - The morphology processor uses this feature to correctly inflect plural nouns. This feature will be looked at first before any reference to lexicons or morphology rules. |
| * + - **Applies to** | * + - Nouns only. |
| * + - **Default** | * + - null. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PLURAL)

#### PREDICATIVE public static final java.lang.String PREDICATIVE This flag is set on adjectives that can also be used as a predicate. For example *happy*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *predicative* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any supporting lexicon. |
| * + - **Used by** | * + - Currently not used. |
| * + - **Applies to** | * + - Adjectives only. |
| * + - **Default** | * + - Boolean.FALSE. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PREDICATIVE)

#### PRESENT\_PARTICIPLE public static final java.lang.String PRESENT\_PARTICIPLE This feature gives the present participle form of a verb. For example, the present participle form of *eat* is *eating* and the present participle form of *walk* is *walking*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *presentParticiple* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - All supporting lexicons but can be set by the user for irregular cases. |
| * + - **Used by** | * + - The morphology processor uses this feature to correctly inflect verbs. This feature will be looked at first before any reference to lexicons or morphology rules. |
| * + - **Applies to** | * + - Verbs only. |
| * + - **Default** | * + - null. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PRESENT_PARTICIPLE)

#### PRESENT3S public static final java.lang.String PRESENT3S This feature gives the present third person singular form of a verb. For example, the present participle form of *eat* is *eats* as in *the dog eats*. Another example is *ran* being the present third person singular form of *run* as in *John ran home*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *present3s* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - All supporting lexicons but can be set by the user for irregular cases. |
| * + - **Used by** | * + - The morphology processor uses this feature to correctly inflect verbs. This feature will be looked at first before any reference to lexicons or morphology rules. |
| * + - **Applies to** | * + - Verbs only. |
| * + - **Default** | * + - null. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PRESENT3S)

#### PROPER public static final java.lang.String PROPER This flag is used to determine whether a noun is a proper noun, such as a person's name.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *proper* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Can be set by supporting lexicons or by the user. |
| * + - **Used by** | * + - The morphology processor will not pluralise proper nouns. |
| * + - **Applies to** | * + - Nouns only. |
| * + - **Default** | * + - Boolean.FALSE. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.PROPER)

#### QUALITATIVE public static final java.lang.String QUALITATIVE This feature is used for determining the position of adjectives. Setting this value to true means that the adjective can occupy the *qualitative* position.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *qualitative* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon that supports adjective positioning. |
| * + - **Used by** | * + - The syntax processor to determine the ordering of adjectives. |
| * + - **Applies to** | * + - Adjectives within noun phrases. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.QUALITATIVE)

#### REFLEXIVE public static final java.lang.String REFLEXIVE This flag is set if a pronoun is written in the reflexive form. For example, *myself*, *yourself*, *ourselves*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *isReflexive* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - The phrase factory will recognise personal pronouns in reflexive form. |
| * + - **Used by** | * + - The morphology processor will correctly inflect reflexive pronouns. |
| * + - **Applies to** | * + - Pronouns only. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.REFLEXIVE)

#### SENTENCE\_MODIFIER public static final java.lang.String SENTENCE\_MODIFIER This feature is used to define whether an adverb can be used as a clause modifier, which are normally applied at the beginning of clauses. For example, *unfortunately*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *sentenceModifier* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon that supports this feature. |
| * + - **Used by** | * + - generic addModifier methods, to decide where to put an adverb |
| * + - **Applies to** | * + - Adverbs only. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.SENTENCE_MODIFIER)

#### SUPERLATIVE public static final java.lang.String SUPERLATIVE This feature gives the superlative form for adjectives and adverbs. For example, *fattest* is the superlative form of *fat* and *earliest* is the superlative form of *early*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *superlative* |
| * + - **Expected type** | * + - String |
| * + - **Created by** | * + - Can be created automatically by the lexicon or added manually by users. |
| * + - **Used by** | * + - The morphology processor uses this information to correctly inflect words. |
| * + - **Applies to** | * + - Adjectives and adverbs only. |
| * + - **Default** | * + - null |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.SUPERLATIVE)

#### TRANSITIVE public static final java.lang.String TRANSITIVE This flag highlights a verb that can only take a subject and an object.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *transitive* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon supporting this feature. |
| * + - **Used by** | * + - Currently not used. |
| * + - **Applies to** | * + - Verbs only. |
| * + - **Default** | * + - Boolean.FALSE. |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.TRANSITIVE)

#### VERB\_MODIFIER public static final java.lang.String VERB\_MODIFIER This feature is used to define whether an adverb can be used as a verb modifier, which are normally added in a phrase before the verb itself. For example, *quickly*.

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *verbModifier* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - Any lexicon that supports this feature. |
| * + - **Used by** | * + - generic addModifier methods, to decide where to put an adverb. |
| * + - **Applies to** | * + - Adverbs only. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.VERB_MODIFIER)
    - EXPLETIVE\_SUBJECT  
      public static final java.lang.String EXPLETIVE\_SUBJECT  
      This feature determines if the pronoun is an expletive or not. Expletive pronouns are usually *it* or *there* in sentences such as:  
      ***It*** *is raining now.*  
      ***There*** *are ten desks in the room.*

|  |  |
| --- | --- |
| * + - **Feature name** | * + - *isExpletive* |
| * + - **Expected type** | * + - Boolean |
| * + - **Created by** | * + - The feature needs to be set by the user. |
| * + - **Used by** | * + - The syntax processor uses the expletive on verb phrases for determining the correct number agreement. |
| * + - **Applies to** | * + - Certain pronouns when used as subjects of verb phrases. |
| * + - **Default** | * + - Boolean.FALSE |

* + - See Also:[Constant Field Values](http://docs.google.com/constant-values.html#simplenlg.features.LexicalFeature.EXPLETIVE_SUBJECT)

### Method Detail

#### getInflectionalFeatures public static java.lang.String[] getInflectionalFeatures([ElementCategory](http://docs.google.com/simplenlg/framework/ElementCategory.html) cat) Return those features related to a word's inflection, depending on its category, that is, the constants for PAST, PAST\_PARTICIPLE, PLURAl, PRESENT\_PARTICIPLE, PRESENT3S, COMPARATIVE or SUPERLATIVE.Parameters:cat - the category Returns:the inflectional feature names

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/LexicalFeature.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/features/InterrogativeType.html)
* [Next Class](http://docs.google.com/simplenlg/features/NumberAgreement.html)
* [Frames](http://docs.google.com/index.html?simplenlg/features/LexicalFeature.html)
* [No Frames](http://docs.google.com/LexicalFeature.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#id.3znysh7)|
* Constr |
* [Method](#id.2et92p0)
* Detail:
* [Field](#id.3dy6vkm)|
* Constr |
* [Method](#id.ihv636)