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/MorphologyProcessor.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/morphology/english/DeterminerAgrHelper.html)
* [Next Class](http://docs.google.com/simplenlg/morphology/english/MorphologyRules.html)
* [Frames](http://docs.google.com/index.html?simplenlg/morphology/english/MorphologyProcessor.html)
* [No Frames](http://docs.google.com/MorphologyProcessor.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.2s8eyo1)

simplenlg.morphology.english

## Class MorphologyProcessor

* java.lang.Object
  + [simplenlg.framework.NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html)
    - simplenlg.morphology.english.MorphologyProcessor
* public class MorphologyProcessor  
  extends [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html)  
  This is the processor for handling morphology within the SimpleNLG. The processor inflects words form the base form depending on the features applied to the word. For example, *kiss* is inflected to *kissed* for past tense, *dog* is inflected to *dogs* for pluralisation.  
  As a matter of course, the processor will first use any user-defined inflection for the world. If no inflection is provided then the lexicon, if it exists, will be examined for the correct inflection. Failing this a set of very basic rules will be examined to inflect the word.  
  All processing modules perform realisation on a tree of NLGElements. The modules can alter the tree in whichever way they wish. For example, the syntax processor replaces phrase elements with list elements consisting of inflected words while the morphology processor replaces inflected words with string elements.  
  **N.B.** the use of *module*, *processing module* and *processor* is interchangeable. They all mean an instance of this class.Version: 4.0 Author: D. Westwater, University of Aberdeen.

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**MorphologyProcessor**](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html#MorphologyProcessor())() |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [**initialise**](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html#initialise())() Performs one-time initialisation of the module. |
| * + java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> | * + [**realise**](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html#realise(java.util.List))(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> elements) Realises a List of NLGElements usually by iteratively calling the realise(NLGElement) method on each element in the list and adding the result into a new a List |
| * + [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**realise**](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html#realise(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Realises the given element. |

### Methods inherited from class simplenlg.framework.[**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html)[getLexicon](http://docs.google.com/simplenlg/framework/NLGModule.html#getLexicon()), [setLexicon](http://docs.google.com/simplenlg/framework/NLGModule.html#setLexicon(simplenlg.lexicon.Lexicon))

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

### Constructor Detail

#### MorphologyProcessor public MorphologyProcessor()

### Method Detail

#### initialise public void initialise() **Description copied from class:**[**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html#initialise()) Performs one-time initialisation of the module.**Specified by:** [initialise](http://docs.google.com/simplenlg/framework/NLGModule.html#initialise()) in class [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html)

#### realise public [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) realise([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) **Description copied from class:**[**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(simplenlg.framework.NLGElement)) Realises the given element. This call is usually recursive as the call processes the child elements of the given element.**Specified by:** [realise](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(simplenlg.framework.NLGElement)) in class [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) Parameters:element - the NLGElement to be realised. Returns:the NLGElement representing the realised state. This may be the initial element in a changed form or be a completely new element.

#### realise public java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> realise(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> elements) **Description copied from class:**[**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(java.util.List)) Realises a List of NLGElements usually by iteratively calling the realise(NLGElement) method on each element in the list and adding the result into a new a List**Specified by:** [realise](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(java.util.List)) in class [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) Parameters:elements - the List of NLGElements to be realised. Returns:the List of realised NLGElements.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/MorphologyProcessor.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/morphology/english/DeterminerAgrHelper.html)
* [Next Class](http://docs.google.com/simplenlg/morphology/english/MorphologyRules.html)
* [Frames](http://docs.google.com/index.html?simplenlg/morphology/english/MorphologyProcessor.html)
* [No Frames](http://docs.google.com/MorphologyProcessor.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.2s8eyo1)