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/Realiser.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
* Next Class
* [Frames](http://docs.google.com/index.html?simplenlg/realiser/english/Realiser.html)
* [No Frames](http://docs.google.com/Realiser.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.17dp8vu)

simplenlg.realiser.english

## Class Realiser

* java.lang.Object
  + [simplenlg.framework.NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html)
    - simplenlg.realiser.english.Realiser
* public class Realiser  
  extends [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html)Author: D. Westwater, Data2Text Ltd

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**Realiser**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#Realiser())() create a realiser (no lexicon) |
| * + [**Realiser**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#Realiser(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lexicon) Create a realiser with a lexicon (should match lexicon used for NLGFactory) |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [**initialise**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#initialise())() Performs one-time initialisation of the module. |
| * + boolean | * + [**isCommaSepCuephrase**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#isCommaSepCuephrase())() Check whether this processor separates cue phrases from the matrix clause using a comma. |
| * + boolean | * + [**isCommaSepPremodifiers**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#isCommaSepPremodifiers())() Check whether this processor separates premodifiers using a comma. |
| * + java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> | * + [**realise**](http://docs.google.com/simplenlg/realiser/english/Realiser.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/realiser/english/Realiser.html#realise(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Realises the given element. |
| * + java.lang.String | * + [**realiseSentence**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#realiseSentence(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Convenience class to realise any NLGElement as a sentence |
| * + void | * + [**setCommaSepCuephrase**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#setCommaSepCuephrase(boolean))(boolean commaSepCuephrase) Set whether to separate cue phrases from the host phrase using a comma. |
| * + void | * + [**setCommaSepPremodifiers**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#setCommaSepPremodifiers(boolean))(boolean commaSepPremodifiers) Set whether to separate premodifiers using a comma. |
| * + void | * + [**setDebugMode**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#setDebugMode(boolean))(boolean debugOn) |
| * + void | * + [**setFormatter**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#setFormatter(simplenlg.framework.NLGModule))([NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) formatter) |
| * + void | * + [**setLexicon**](http://docs.google.com/simplenlg/realiser/english/Realiser.html#setLexicon(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Sets the lexicon to be used by this module. |

### 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())

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

### Constructor Detail

#### Realiser public Realiser() create a realiser (no lexicon)

#### Realiser public Realiser([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lexicon) Create a realiser with a lexicon (should match lexicon used for NLGFactory)Parameters:lexicon -

### Method Detail

* + - isCommaSepPremodifiers  
      public boolean isCommaSepPremodifiers()  
      Check whether this processor separates premodifiers using a comma.  
        
      **Implementation note:** this method checks whether the [OrthographyProcessor](http://docs.google.com/simplenlg/orthography/english/OrthographyProcessor.html) has the parameter set.Returns:true if premodifiers in the noun phrase are comma-separated.
    - setCommaSepPremodifiers  
      public void setCommaSepPremodifiers(boolean commaSepPremodifiers)  
      Set whether to separate premodifiers using a comma. If true, premodifiers will be comma-separated, as in *the long, dark road*. If false, they won't.  
        
      **Implementation note:**: this method sets the relevant parameter in the [OrthographyProcessor](http://docs.google.com/simplenlg/orthography/english/OrthographyProcessor.html).Parameters:commaSepPremodifiers - the commaSepPremodifiers to set
    - isCommaSepCuephrase  
      public boolean isCommaSepCuephrase()  
      Check whether this processor separates cue phrases from the matrix clause using a comma.  
        
      **Implementation note:** this method checks whether the [OrthographyProcessor](http://docs.google.com/simplenlg/orthography/english/OrthographyProcessor.html) has the parameter set.Returns:true if cue phrases have a comma before the remainder of the host phrase
    - setCommaSepCuephrase  
      public void setCommaSepCuephrase(boolean commaSepCuephrase)  
      Set whether to separate cue phrases from the host phrase using a comma. If true, a comma will be inserted, as in *however, Bill arrived late*. If false, they won't.  
        
      **Implementation note:**: this method sets the relevant parameter in the [OrthographyProcessor](http://docs.google.com/simplenlg/orthography/english/OrthographyProcessor.html).Parameters:commaSepcuephrase -

#### 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.

#### realiseSentence public java.lang.String realiseSentence([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Convenience class to realise any NLGElement as a sentenceParameters:element - Returns:String realisation of the NLGElement

#### 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.

#### setLexicon public void setLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) **Description copied from class:**[**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html#setLexicon(simplenlg.lexicon.Lexicon)) Sets the lexicon to be used by this module. Passing in null will remove the existing lexicon and no lexicon will be used.**Overrides:** [setLexicon](http://docs.google.com/simplenlg/framework/NLGModule.html#setLexicon(simplenlg.lexicon.Lexicon)) in class [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) Parameters:newLexicon - the new Lexicon to be used.

#### setFormatter public void setFormatter([NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) formatter)

#### setDebugMode public void setDebugMode(boolean debugOn)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Realiser.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
* Next Class
* [Frames](http://docs.google.com/index.html?simplenlg/realiser/english/Realiser.html)
* [No Frames](http://docs.google.com/Realiser.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.17dp8vu)