JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* Package
* Class
* [Use](http://docs.google.com/package-use.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 Package](http://docs.google.com/simplenlg/format/english/package-summary.html)
* [Next Package](http://docs.google.com/simplenlg/lexicon/package-summary.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/package-summary.html)
* [No Frames](http://docs.google.com/package-summary.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)

# 

Package simplenlg.framework

The framework package contains the base components required for running the SimpleNLG package.

See: [Description](#id.3znysh7)

* Interface Summary

|  |  |
| --- | --- |
| * Interface | * Description |
| * [ElementCategory](http://docs.google.com/simplenlg/framework/ElementCategory.html) | * This is the base interface for defining categories for the sub classes of NLGElement. |

* Class Summary

|  |  |
| --- | --- |
| * Class | * Description |
| * [CoordinatedPhraseElement](http://docs.google.com/simplenlg/framework/CoordinatedPhraseElement.html) | * This class defines coordination between two or more phrases. |
| * [DocumentElement](http://docs.google.com/simplenlg/framework/DocumentElement.html) | * DocumentElement is a convenient extension of the base NLGElement class. |
| * [InflectedWordElement](http://docs.google.com/simplenlg/framework/InflectedWordElement.html) | * This class defines the NLGElement that is used to represent an word that requires inflection by the morphology. |
| * [ListElement](http://docs.google.com/simplenlg/framework/ListElement.html) | * ListElement is used to define elements that can be grouped together and treated in a similar manner. |
| * [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * NLGElement is the base class that all elements extend from. |
| * [NLGFactory](http://docs.google.com/simplenlg/framework/NLGFactory.html) | * This class contains methods for creating syntactic phrases. |
| * [NLGModule](http://docs.google.com/simplenlg/framework/NLGModule.html) | * NLGModule is the base class that all processing modules extend from. |
| * [PhraseElement](http://docs.google.com/simplenlg/framework/PhraseElement.html) | * This class defines a phrase. |
| * [StringElement](http://docs.google.com/simplenlg/framework/StringElement.html) | * This class defines an element for representing canned text within the SimpleNLG library. |
| * [WordElement](http://docs.google.com/simplenlg/framework/WordElement.html) | * This is the class for a lexical entry (ie, a word). |

* Enum Summary

|  |  |
| --- | --- |
| * Enum | * Description |
| * [DocumentCategory](http://docs.google.com/simplenlg/framework/DocumentCategory.html) | * This enumerated type defines the different *types* of components found in the structure of text. |
| * [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) | * This enumeration defines the different lexical components. |
| * [PhraseCategory](http://docs.google.com/simplenlg/framework/PhraseCategory.html) | * This enumeration defines the different syntactical phrases. |

## Package simplenlg.framework Description

The framework package contains the base components required for running the SimpleNLG package. The components can be split into four different categories. The first category contains only the NLGModule abstract class. This module is used to define processors that will run in the SimpleNLG package. Some example basic processors are included with the application and they all extend this class. Note that modules and processors are interchangeable in this documentation. They both refer to a class that extends the NLGModule base class.

The second category of components are formed from the elements. Elements in SimpleNLG form the document structure, phrases, clauses, canned text and the words themselves. There are numerous Element classes all of which extend from the top-level abstract class NLGElement. Generic modules can be written to run with the generic NLGElement class. The other element classes are:

* CoordinatedPhraseElement - used for coordinating phrases together through the use of conjunctions such as *and* and *but*.
* DocumentElement - defines document structure of a text to the level of document, section, paragraph and sentences.
* InflectedWordElement - represents a word that needs to be inflected.
* ListElement - a single element entity representing a list of NLGElements and should not be confused with List<NLGElement>
* PhraseElement - defines the phrases of the text such as: clauses, noun phrases and verb phrases.
* StringElement - a canned text representation that is not meant to be changed by any of the processors.
* WordElement - a representation of a base word as read from the lexicon definition.

The third set of components are the enumerated types representing the different categories for the elements. All the enumerations implement the ElementCategory interface. This allows for generic modules to access a generic category type. The other enumerations are:

* DocumentCategory - defines which part of the document structure the DocumentElement represents. For example, sentence and paragraph.
* LexicalCategory - represents the types of grammatical categories for the individual words. For example, noun, verb and adjective.
* PhraseCategory - details the type of phrase a PhraseElement represents. For example, noun phrase, clause, prepositional phrase.

The final category covers the two factories. The factories should be used to construct other elements. They are: DocumentFactory, which constructs document elements pertaining to the text structure; and PhraseFactory, which constructs various types of phrases.

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is "Simplenlg".

The Initial Developer of the Original Code is Ehud Reiter, Albert Gatt and Dave Westwater. Portions created by Ehud Reiter, Albert Gatt and Dave Westwater are Copyright (C) 2010-11 The University of Aberdeen. All Rights Reserved.

Contributor(s): Ehud Reiter, Albert Gatt, Dave Wewstwater, Roman Kutlak, Margaret Mitchell.

* [Overview](http://docs.google.com/overview-summary.html)
* Package
* Class
* [Use](http://docs.google.com/package-use.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 Package](http://docs.google.com/simplenlg/format/english/package-summary.html)
* [Next Package](http://docs.google.com/simplenlg/lexicon/package-summary.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/package-summary.html)
* [No Frames](http://docs.google.com/package-summary.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)