

# Rage BML Realizer

1.0

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	AssetPackage Namespace Reference . . . . .	7
4.2	BMLRealizer Namespace Reference . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	BMLRealizer.BMLBehavior Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	10
5.1.2	Constructor & Destructor Documentation . . . . .	10
5.1.2.1	BMLBehavior() . . . . .	10
5.1.3	Member Function Documentation . . . . .	10
5.1.3.1	Parse() . . . . .	10
5.2	BMLRealizer.BMLBlock Class Reference . . . . .	11
5.2.1	Detailed Description . . . . .	12
5.2.2	Constructor & Destructor Documentation . . . . .	12
5.2.2.1	BMLBlock() . . . . .	12
5.2.3	Member Function Documentation . . . . .	12

5.2.3.1	Parse()	12
5.2.3.2	TryParseAttribute< T >()	12
5.2.3.3	TryParseSyncPoint()	13
5.2.4	Member Data Documentation	14
5.2.4.1	id	14
5.2.4.2	parentBml	14
5.2.4.3	syncPoints	14
5.3	BMLRealizer.BMLBml Class Reference	15
5.3.1	Detailed Description	15
5.3.2	Member Enumeration Documentation	15
5.3.2.1	Composition	15
5.3.3	Constructor & Destructor Documentation	16
5.3.3.1	BMLBml()	16
5.3.4	Member Function Documentation	16
5.3.4.1	Parse()	16
5.3.5	Member Data Documentation	16
5.3.5.1	characterId	16
5.3.5.2	composition	16
5.4	BMLRealizer.BMLFace Class Reference	17
5.4.1	Detailed Description	17
5.4.2	Constructor & Destructor Documentation	17
5.4.2.1	BMLFace()	17
5.4.3	Member Function Documentation	17
5.4.3.1	Parse()	17
5.4.4	Member Data Documentation	18
5.4.4.1	amount	18
5.4.4.2	overshoot	18
5.5	BMLRealizer.BMLFaceFacs Class Reference	18
5.5.1	Detailed Description	19
5.5.2	Member Enumeration Documentation	19

5.5.2.1	Side	19
5.5.3	Constructor & Destructor Documentation	19
5.5.3.1	BMLFaceFacs()	19
5.5.4	Member Data Documentation	20
5.5.4.1	au	20
5.5.4.2	side	20
5.6	BMLRealizer.BMLFaceLexeme Class Reference	20
5.6.1	Detailed Description	21
5.6.2	Member Enumeration Documentation	21
5.6.2.1	Lexeme	21
5.6.3	Constructor & Destructor Documentation	21
5.6.3.1	BMLFaceLexeme()	21
5.6.4	Member Function Documentation	21
5.6.4.1	Parse()	21
5.6.5	Member Data Documentation	22
5.6.5.1	lexeme	22
5.7	BMLRealizer.BMLFaceShift Class Reference	22
5.7.1	Detailed Description	22
5.8	BMLRealizer.BMLFeedback Class Reference	23
5.8.1	Detailed Description	23
5.9	BMLRealizer.BMLGaze Class Reference	23
5.9.1	Detailed Description	24
5.9.2	Member Enumeration Documentation	24
5.9.2.1	Direction	24
5.9.2.2	Influence	24
5.9.3	Constructor & Destructor Documentation	24
5.9.3.1	BMLGaze()	24
5.9.4	Member Function Documentation	24
5.9.4.1	Parse()	24
5.9.5	Member Data Documentation	25

5.9.5.1	influence	25
5.9.5.2	offsetAngle	25
5.9.5.3	offsetDirection	25
5.9.5.4	target	25
5.10	BMLRealizer.BMLGazeShift Class Reference	26
5.10.1	Detailed Description	26
5.10.2	Constructor & Destructor Documentation	26
5.10.2.1	BMLGazeShift()	26
5.10.3	Member Function Documentation	26
5.10.3.1	Parse()	26
5.11	BMLRealizer.BMLGesture Class Reference	27
5.11.1	Detailed Description	28
5.11.2	Member Enumeration Documentation	28
5.11.2.1	Lexeme	28
5.11.2.2	Mode	28
5.11.3	Constructor & Destructor Documentation	28
5.11.3.1	BMLGesture()	28
5.11.4	Member Function Documentation	28
5.11.4.1	Parse()	28
5.11.5	Member Data Documentation	29
5.11.5.1	lexeme	29
5.11.5.2	mode	29
5.12	BMLRealizer.BMLHead Class Reference	29
5.12.1	Detailed Description	30
5.12.2	Member Enumeration Documentation	30
5.12.2.1	Lexeme	30
5.12.3	Constructor & Destructor Documentation	31
5.12.3.1	BMLHead()	31
5.12.4	Member Function Documentation	31
5.12.4.1	Parse()	31

5.12.5	Member Data Documentation . . . . .	31
5.12.5.1	amount . . . . .	31
5.12.5.2	lexeme . . . . .	31
5.12.5.3	repetition . . . . .	32
5.13	BMLRealizer.BMLHeadDirectionShift Class Reference . . . . .	32
5.13.1	Detailed Description . . . . .	32
5.13.2	Constructor & Destructor Documentation . . . . .	33
5.13.2.1	BMLHeadDirectionShift() . . . . .	33
5.13.3	Member Function Documentation . . . . .	33
5.13.3.1	Parse() . . . . .	33
5.13.4	Member Data Documentation . . . . .	33
5.13.4.1	target . . . . .	33
5.14	BMLRealizer.BMLLocomotion Class Reference . . . . .	34
5.14.1	Detailed Description . . . . .	34
5.14.2	Constructor & Destructor Documentation . . . . .	34
5.14.2.1	BMLLocomotion() . . . . .	34
5.14.3	Member Function Documentation . . . . .	34
5.14.3.1	Parse() . . . . .	34
5.14.4	Member Data Documentation . . . . .	35
5.14.4.1	manner . . . . .	35
5.14.4.2	target . . . . .	35
5.15	BMLRealizer.BMLPointing Class Reference . . . . .	35
5.15.1	Detailed Description . . . . .	36
5.15.2	Member Enumeration Documentation . . . . .	36
5.15.2.1	Mode . . . . .	36
5.15.3	Constructor & Destructor Documentation . . . . .	36
5.15.3.1	BMLPointing() . . . . .	36
5.15.4	Member Function Documentation . . . . .	36
5.15.4.1	Parse() . . . . .	36
5.15.5	Member Data Documentation . . . . .	37

5.15.5.1	mode	37
5.15.5.2	target	37
5.16	BMLRealizer.BMLPose Class Reference	37
5.16.1	Detailed Description	38
5.16.2	Member Enumeration Documentation	38
5.16.2.1	Lexeme	38
5.16.2.2	Part	39
5.16.3	Constructor & Destructor Documentation	39
5.16.3.1	BMLPose()	39
5.16.4	Member Function Documentation	39
5.16.4.1	Parse()	39
5.16.5	Member Data Documentation	39
5.16.5.1	lexeme	39
5.16.5.2	part	40
5.17	BMLRealizer.BMLPosture Class Reference	40
5.17.1	Detailed Description	40
5.17.2	Constructor & Destructor Documentation	41
5.17.2.1	BMLPosture()	41
5.17.3	Member Function Documentation	41
5.17.3.1	Parse()	41
5.18	BMLRealizer.BMLPostureShift Class Reference	41
5.18.1	Detailed Description	42
5.18.2	Constructor & Destructor Documentation	42
5.18.2.1	BMLPostureShift()	42
5.18.3	Member Function Documentation	42
5.18.3.1	Parse()	42
5.19	BMLRealizer.BMLSpeech Class Reference	43
5.19.1	Detailed Description	43
5.19.2	Constructor & Destructor Documentation	43
5.19.2.1	BMLSpeech()	43



5.19.3 Member Function Documentation . . . . .	44
5.19.3.1 Parse() . . . . .	44
5.19.4 Member Data Documentation . . . . .	44
5.19.4.1 text . . . . .	44
5.20 BMLRealizer.BMLStance Class Reference . . . . .	44
5.20.1 Detailed Description . . . . .	45
5.20.2 Member Enumeration Documentation . . . . .	45
5.20.2.1 Type . . . . .	45
5.20.3 Constructor & Destructor Documentation . . . . .	45
5.20.3.1 BMLStance() . . . . .	45
5.20.4 Member Function Documentation . . . . .	45
5.20.4.1 Parse() . . . . .	45
5.20.5 Member Data Documentation . . . . .	46
5.20.5.1 type . . . . .	46
5.21 BMLRealizer.BMLSyncPoint Class Reference . . . . .	46
5.21.1 Detailed Description . . . . .	47
5.21.2 Constructor & Destructor Documentation . . . . .	47
5.21.2.1 BMLSyncPoint() . . . . .	47
5.21.3 Member Function Documentation . . . . .	47
5.21.3.1 IsCompleted() . . . . .	47
5.21.3.2 IsTimerSafe() . . . . .	48
5.21.3.3 TriggerSyncPoint() . . . . .	48
5.21.3.4 Update() . . . . .	48
5.22 BMLRealizer.BMLWait Class Reference . . . . .	49
5.22.1 Detailed Description . . . . .	49
5.22.2 Constructor & Destructor Documentation . . . . .	49
5.22.2.1 BMLWait() . . . . .	49
5.22.3 Member Function Documentation . . . . .	49
5.22.3.1 Parse() . . . . .	49
5.22.4 Member Data Documentation . . . . .	50

5.22.4.1	duration	50
5.23	AssetPackage.RageBMLRealizer Class Reference	50
5.23.1	Detailed Description	51
5.23.2	Constructor & Destructor Documentation	51
5.23.2.1	RageBMLRealizer()	51
5.23.3	Member Function Documentation	51
5.23.3.1	GetBehaviorFromId()	51
5.23.3.2	SyncPointCompleted()	52
5.23.3.3	TriggerSyncPoint()	52
5.23.3.4	Update()	53
5.23.4	Property Documentation	53
5.23.4.1	ScheduledBlocks	53
5.23.4.2	Settings	53
5.23.4.3	Timer	53
5.24	AssetPackage.RageBMLRealizerSettings Class Reference	54
5.24.1	Detailed Description	54
5.24.2	Constructor & Destructor Documentation	54
5.24.2.1	RageBMLRealizerSettings()	54
<b>Index</b>		<b>55</b>

# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">AssetPackage</a>	.....	<a href="#">7</a>
<a href="#">BMLRealizer</a>	.....	<a href="#">7</a>



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

BaseAsset	
AssetPackage.RageBMLRealizer . . . . .	50
BaseSettings	
AssetPackage.RageBMLRealizerSettings . . . . .	54
BMLRealizer.BMLBlock . . . . .	11
BMLRealizer.BMLBehavior . . . . .	9
BMLRealizer.BMLFace . . . . .	17
BMLRealizer.BMLFaceFacs . . . . .	18
BMLRealizer.BMLFaceLexeme . . . . .	20
BMLRealizer.BMLFaceShift . . . . .	22
BMLRealizer.BMLGaze . . . . .	23
BMLRealizer.BMLGazeShift . . . . .	26
BMLRealizer.BMLGesture . . . . .	27
BMLRealizer.BMLHead . . . . .	29
BMLRealizer.BMLHeadDirectionShift . . . . .	32
BMLRealizer.BMLLocomotion . . . . .	34
BMLRealizer.BMLPointing . . . . .	35
BMLRealizer.BMLPose . . . . .	37
BMLRealizer.BMLPosture . . . . .	40
BMLRealizer.BMLPostureShift . . . . .	41
BMLRealizer.BMLSpeech . . . . .	43
BMLRealizer.BMLStance . . . . .	44
BMLRealizer.BMLWait . . . . .	49
BMLRealizer.BMLBml . . . . .	15
BMLRealizer.BMLFeedback . . . . .	23
BMLRealizer.BMLSyncPoint . . . . .	46



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">BMLRealizer.BMLBehavior</a>	9
BML behavior class. all behavior need to derived from this class . . . . .	
<a href="#">BMLRealizer.BMLBlock</a>	11
abstract class of BML block all block need to be derived from this class . . . . .	
<a href="#">BMLRealizer.BMLBml</a>	15
<a href="#">BMLRealizer.BMLFace</a>	
Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit. . . . .	17
<a href="#">BMLRealizer.BMLFaceFacs</a>	
This behavior provides control of the face through single Action Units from the Facial Action Coding Scheme. It is an Core Extension, that is, not every BML Compliant Realizer has to implement this behavior, but if a Realizer offers FACS based face control, they should adhere to the specification of this <faceFacs> behavior . . . . .	18
<a href="#">BMLRealizer.BMLFaceLexeme</a>	20
Show a (partial) face expression from a predefined lexicon. . . . .	
<a href="#">BMLRealizer.BMLFaceShift</a>	
Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit, where the specified compound face expression becomes the new BASE state of the ECAs face. . . . .	22
<a href="#">BMLRealizer.BMLFeedback</a>	23
<a href="#">BMLRealizer.BMLGaze</a>	
Temporarily directs the gaze of the character towards a target. This behavior causes the character to temporarily direct its gaze to the requested target. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera. . . . .	23
<a href="#">BMLRealizer.BMLGazeShift</a>	
Permanently change the gaze direction of the character towards a certain target. This behavior causes the character to direct its gaze to the requested target. This changes the default state of the ECA: after completing this behavior, the new target is the default gaze direction of the ECA. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera. . . . .	26
<a href="#">BMLRealizer.BMLGesture</a>	
Currently, BML offers two types of gesture behaviors. The first provides a set of gestures recalled by name from a gesticon; the second provides simple pointing gestures. Coordinated movement with arms and hands, recalled from a gesticon by requesting the corresponding lexeme . . . .	27

<a href="#">BMLRealizer.BMLHead</a>	
Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. The stroke phase of the head motion (from strokeStart till strokeEnd is the "meaningful" part of the head motion. The stroke sync point is the "peak" moment of the motion. If repetition > 1, the meaning of the stroke sync point becomes undefined . . . . .	29
<a href="#">BMLRealizer.BMLHeadDirectionShift</a>	
Orient the head towards a target referenced by the target attribute. Permanently orient the head in a certain direction. . . . .	32
<a href="#">BMLRealizer.BMLLocomotion</a>	
Move the body of the character from one location to another. This behavior causes the character to move to the requested target in the manner described. . . . .	34
<a href="#">BMLRealizer.BMLPointing</a>	
Deictic gesture towards the target specified by the target attribute . . . . .	35
<a href="#">BMLRealizer.BMLPose</a>	
Child element of <posture> and <postureShift> behaviors, defines additions to the global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines additions that modify the global body posture of the ECA. For each value of the part attribute, only one <pose> child is expected to be present. A BML Realizer may define any number of lexemes beyond the ones specified above. . . . .	37
<a href="#">BMLRealizer.BMLPosture</a>	
Temporarily change the posture of the ECA. Temporarily change the posture of the ECA. After the <posture> behavior has ended, return to the BASE posture. . . . .	40
<a href="#">BMLRealizer.BMLPostureShift</a>	
Permanently change the BASE posture of the ECA. . . . .	41
<a href="#">BMLRealizer.BMLSpeech</a>	
Utterance to be spoken by a character. Realization of the <speech> element generates both speech audio (or text) and speech movement, for example using a speech synthesizer and viseme morphing. The <speech> element requires a sub-element. This sub-element is a <text> element that contains the text to be spoken, with optionally embedded <sync> elements for alignment with other behaviors. . . . .	43
<a href="#">BMLRealizer.BMLStance</a>	
Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. This global posture may then be modified by one or more <pose> siblings. . . . .	44
<a href="#">BMLRealizer.BMLSyncPoint</a>	
BML Sync Point class possible format: behavior_id:sync_id [+/- offset] A reference to a sync point of another behavior, optionally with a float offset in seconds. By default, this is a behavior in the same <bml> block that the syncref is contained in; if optional prefix block_id: is present, the syncref specifies a sync point of a behavior in the <bml> block with that ID.) offset: A positive float offset in seconds relative to the start time of the surrounding <bml> block . . . . .	46
<a href="#">BMLRealizer.BMLWait</a>	
	49
<a href="#">AssetPackage.RageBMLRealizer</a>	
An <a href="#">BMLRealizer</a> Rage asset . . . . .	50
<a href="#">AssetPackage.RageBMLRealizerSettings</a>	
An asset settings . . . . .	54



## Chapter 4

# Namespace Documentation

### 4.1 AssetPackage Namespace Reference

#### Classes

- class [RageBMLRealizer](#)  
*An [BMLRealizer](#) Rage asset*
- class [RageBMLRealizerSettings](#)  
*An asset settings.*

### 4.2 BMLRealizer Namespace Reference

#### Classes

- class [BMLBehavior](#)  
*BML behavior class. all behavior need to derived from this class*
- class [BMLBlock](#)  
*abstract class of BML block all block need to be derived from this class*
- class [BMLBml](#)
- class [BMLFace](#)  
*Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit.*
- class [BMLFaceFacs](#)  
*This behavior provides control of the face through single Action Units from the Facial Action Coding Scheme. It is an Core Extension, that is, not every BML Compliant Realizer has to implement this behavior, but if a Realizer offers FACS based face control, they should adhere to the specification of this <faceFacs> behavior*
- class [BMLFaceLexeme](#)  
*Show a (partial) face expression from a predefined lexicon.*
- class [BMLFaceShift](#)  
*Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit, where the specified compound face expression becomes the new BASE state of the ECAs face.*
- class [BMLFeedback](#)
- class [BMLGaze](#)  
*Temporarily directs the gaze of the character towards a target. This behavior causes the character to temporarily direct its gaze to the requested target. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.*

- class [BMLGazeShift](#)

*Permanently change the gaze direction of the character towards a certain target. This behavior causes the character to direct its gaze to the requested target. This changes the default state of the ECA: after completing this behavior, the new target is the default gaze direction of the ECA. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.*
- class [BMLGesture](#)

*Currently, BML offers two types of gesture behaviors. The first provides a set of gestures recalled by name from a gesticon; the second provides simple pointing gestures. Coordinated movement with arms and hands, recalled from a gesticon by requesting the corresponding lexeme*
- class [BMLHead](#)

*Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. The stroke phase of the head motion (from stroke↔ Start till strokeEnd is the "meaningful" part of the head motion. The stroke sync point is the "peak" moment of the motion. If repetition > 1, the meaning of the stroke sync point becomes undefined*
- class [BMLHeadDirectionShift](#)

*Orient the head towards a target referenced by the target attribute. Permanently orient the head in a certain direction.*
- class [BMLLocomotion](#)

*Move the body of the character from one location to another. This behavior causes the character to move to the requested target in the manner described.*
- class [BMLPointing](#)

*Deictic gesture towards the target specified by the target attribute*
- class [BMLPose](#)

*Child element of <posture> and <postureShift> behaviors, defines additions to the global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines additions that modify the global body posture of the ECA. For each value of the part attribute, only one <pose> child is expected to be present. A BML Realizer may define any number of lexemes beyond the ones specified above.*
- class [BMLPosture](#)

*Temporarily change the posture of the ECA. Temporarily change the posture of the ECA. After the <posture> behavior has ended, return to the BASE posture.*
- class [BMLPostureShift](#)

*Permanently change the BASE posture of the ECA.*
- class [BMLSpeech](#)

*Utterance to be spoken by a character. Realization of the <speech> element generates both speech audio (or text) and speech movement, for example using a speech synthesizer and viseme morphing. The<speech> element requires a sub-element. This sub-element is a<text> element that contains the text to be spoken, with optionally embedded<sync> elements for alignment with other behaviors.*
- class [BMLStance](#)

*Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. This global posture may then be modified by one or more <pose> siblings.*
- class [BMLSyncPoint](#)

*BML Sync Point class possible format: behavior\_id:sync\_id [+/- offset] A reference to a sync point of another behavior, optionally with a float offset in seconds. By default, this is a behavior in the same <bml> block that the syncref is contained in; if optional prefix block\_id: is present, the syncref specifies a sync point of a behavior in the <bml> block with that ID.) offset: A positive float offset in seconds relative to the start time of the surrounding <bml> block.*
- class [BMLWait](#)

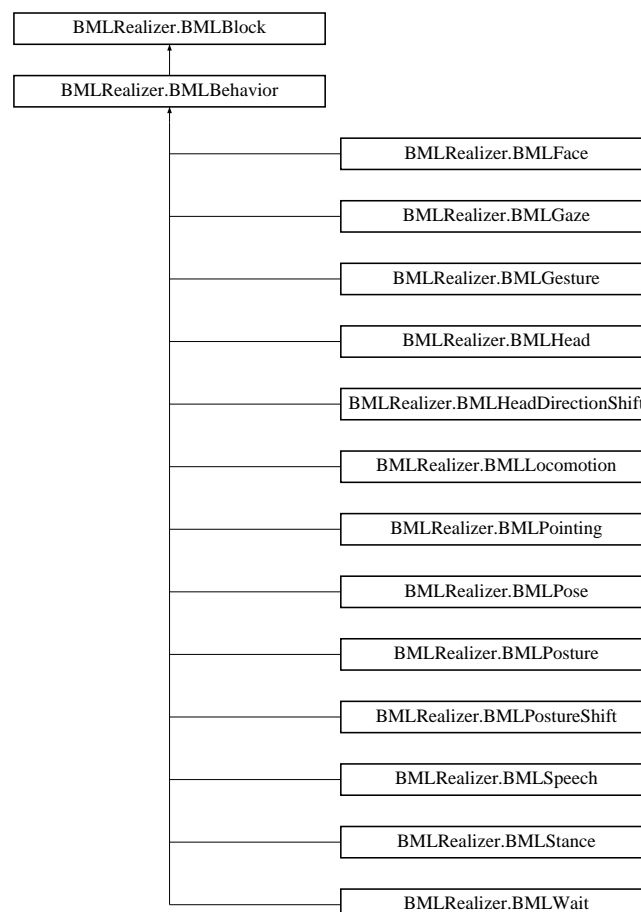
## Chapter 5

# Class Documentation

### 5.1 BMLRealizer.BMLBehavior Class Reference

BML behavior class. all behavior need to derived from this class

Inheritance diagram for BMLRealizer.BMLBehavior:



## Public Member Functions

- [BMLBehavior](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: id*

## Additional Inherited Members

### 5.1.1 Detailed Description

BML behavior class. all behavior need to derived from this class

Definition at line 25 of file BMLBehavior.cs.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 BMLBehavior()

```
BMLRealizer.BMLBehavior.BMLBehavior ( )
```

constructor

Definition at line 31 of file BMLBehavior.cs.

### 5.1.3 Member Function Documentation

#### 5.1.3.1 Parse()

```
override void BMLRealizer.BMLBehavior.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id

Parameters

<i>reader</i>	
---------------	--

Implements [BMLRealizer.BMLBlock](#).

Reimplemented in [BMLRealizer.BMLGaze](#), [BMLRealizer.BMLPose](#), [BMLRealizer.BMLFaceLexeme](#), [BMLRealizer.BMLGesture](#), [BMLRealizer.BMLHead](#), [BMLRealizer.BMLPointing](#), [BMLRealizer.BMLFace](#), [BMLRealizer.BMLLocomotion](#), [BMLRealizer.BMLStance](#), [BMLRealizer.BMLSpeech](#), [BMLRealizer.BMLHeadDirectionShift](#), [BMLRealizer.BMLGazeShift](#), [BMLRealizer.BMLPosture](#), [BMLRealizer.BMLWait](#), and [BMLRealizer.BMLPostureShift](#).

Definition at line 40 of file BMLBehavior.cs.

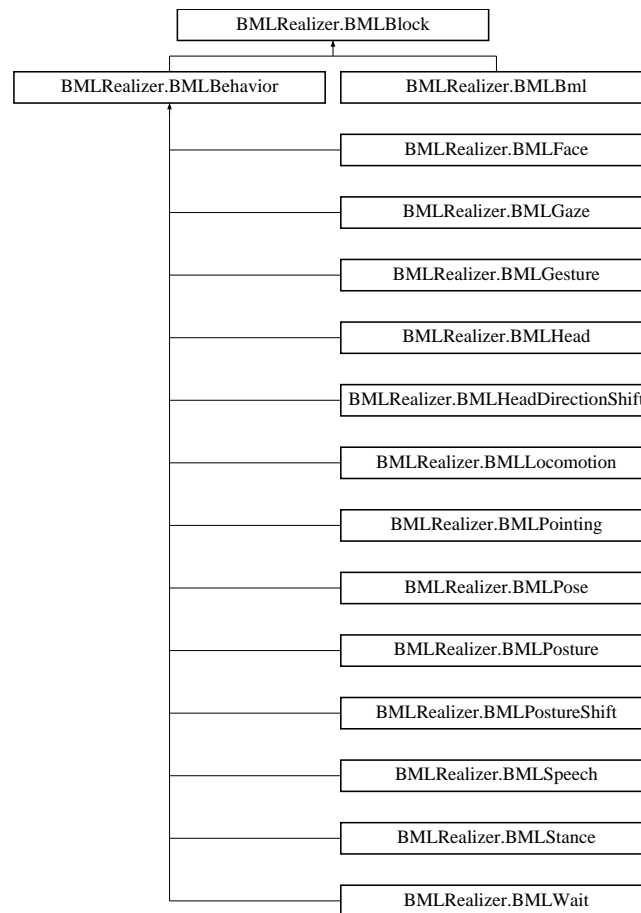
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLBehavior.cs

## 5.2 BMLRealizer.BMLBlock Class Reference

abstract class of BML block all block need to be derived from this class

Inheritance diagram for BMLRealizer.BMLBlock:



### Public Member Functions

- [BMLBlock](#) ()  
*empty constructor*
- abstract void [Parse](#) (XmlReader reader)  
*all child class need to implement their own parsing standard*
- void **Dispose** ()
- String **getCharacterId** ()

### Public Attributes

- string [id](#)  
*Unique ID that allows referencing to a particular bml block. The id 'bml' is reserved.*
- Dictionary< string, [BMLSyncPoint](#) > [syncPoints](#) = new Dictionary<string, [BMLSyncPoint](#)>()  
*Sync Point collection of this block*
- [BMLBml](#) [parentBml](#)  
*parent bml tag*

## Protected Member Functions

- `T TryParseAttribute< T >` (XmlReader reader, string attributeName, T defaultValue, bool required=true)  
*helper function to parse the attribute from XML*
- `bool TryParseSyncPoint` (XmlReader reader, string eventName)  
*helper function to parse the sync point attribute we do not need to check whether we found the attribute or not. The [BMLSyncPoint](#) class will use those value (null or not null).*

### 5.2.1 Detailed Description

abstract class of BML block all block need to be derived from this class

Definition at line 31 of file BMLBlock.cs.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 BMLBlock()

```
BMLRealizer.BMLBlock.BMLBlock ( )
```

empty constructor

Definition at line 52 of file BMLBlock.cs.

### 5.2.3 Member Function Documentation

#### 5.2.3.1 Parse()

```
abstract void BMLRealizer.BMLBlock.Parse (
    XmlReader reader ) [pure virtual]
```

all child class need to implement their own parsing standard

#### Parameters

<i>reader</i>	
---------------	--

Implemented in [BMLRealizer.BMLGaze](#), [BMLRealizer.BMLPose](#), [BMLRealizer.BMLBml](#), [BMLRealizer.BMLFace](#), [BMLRealizer.BMLLexeme](#), [BMLRealizer.BMLGesture](#), [BMLRealizer.BMLHead](#), [BMLRealizer.BMLPointing](#), [BMLRealizer.BMLFace](#), [BMLRealizer.BMLLocomotion](#), [BMLRealizer.BMLStance](#), [BMLRealizer.BMLSpeech](#), [BMLRealizer.BMLHead](#), [BMLRealizer.BMLDirectionShift](#), [BMLRealizer.BMLGazeShift](#), [BMLRealizer.BMLPosture](#), [BMLRealizer.BMLWait](#), [BMLRealizer.BMLBehavior](#), and [BMLRealizer.BMLPostureShift](#).

#### 5.2.3.2 TryParseAttribute< T >()

```
T BMLRealizer.BMLBlock.TryParseAttribute< T > (
    XmlReader reader,
```

```

    string attributeName,
    T defaultValue,
    bool required = true ) [protected]

```

helper function to parse the attribute from XML

#### Template Parameters

<i>T</i>	
----------	--

#### Parameters

<i>reader</i>	
---------------	--

#### XMLReader

#### Parameters

<i>attributeName</i>	
----------------------	--

the attribute name that we need to parse

#### Parameters

<i>defaultValue</i>	
---------------------	--

the value when we do not find the attribute

#### Parameters

<i>required</i>	
-----------------	--

do you require this attribute or not ?

#### Returns

Definition at line 72 of file BMLBlock.cs.

#### 5.2.3.3 TryParseSyncPoint()

```

bool BMLRealizer.BMLBlock.TryParseSyncPoint (
    XmlReader reader,
    string eventName ) [protected]

```

helper function to parse the sync point attribute we do not need to check whether we found the attribute or not. The [BMLSyncPoint](#) class will use those value (null or not null).

**Parameters**

<i>reader</i>	
---------------	--

**XMLReader****Parameters**

<i>eventName</i>	
------------------	--

the name of sync point (start, ready, strokeStart, attackPeak, stroke, strokeEnd, relax, end)

**Returns**

Definition at line 126 of file BMLBlock.cs.

**5.2.4 Member Data Documentation****5.2.4.1 id**

```
string BMLRealizer.BMLBlock.id
```

Unique ID that allows referencing to a particular bml block. The id 'bml' is reserved.

Definition at line 36 of file BMLBlock.cs.

**5.2.4.2 parentBml**

```
BMLBml BMLRealizer.BMLBlock.parentBml
```

parent bml tag

Definition at line 46 of file BMLBlock.cs.

**5.2.4.3 syncPoints**

```
Dictionary<string, BMLSyncPoint> BMLRealizer.BMLBlock.syncPoints = new Dictionary<string, BMLSyncPoint>()
```

Sync Point collection of this block

Definition at line 41 of file BMLBlock.cs.

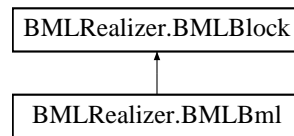
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLBlock.cs



## 5.3 BMLRealizer.BMLBml Class Reference

Inheritance diagram for BMLRealizer.BMLBml:



### Public Types

- enum [Composition](#) { **MERGE**, **APPEND**, **REPLACE** }  
*one among [MERGE,APPEND,REPLACE], defines the composition policy to apply if the current <bml> block overlaps with previous <bml> blocks (see below).*

### Public Member Functions

- [BMLBml](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml*
- void **SetGlobalStartTrigger** (string value)
- void **IncreaseChild** ()
- bool **IncreaseEndChild** ()

### Public Attributes

- string [characterId](#)  
*a reference towards the controlled character*
- string **xmlns**
- [Composition](#) *composition*  
*one among [MERGE,APPEND,REPLACE], defines the composition policy to apply if the current <bml> block overlaps with previous <bml> blocks (see below).*

### Additional Inherited Members

#### 5.3.1 Detailed Description

Definition at line 25 of file BMLBml.cs.

#### 5.3.2 Member Enumeration Documentation

##### 5.3.2.1 Composition

```
enum BMLRealizer.BMLBml.Composition [strong]
```

one among [MERGE,APPEND,REPLACE], defines the composition policy to apply if the current <bml> block overlaps with previous <bml> blocks (see below).

Definition at line 47 of file BMLBml.cs.

### 5.3.3 Constructor & Destructor Documentation

#### 5.3.3.1 BMLBml()

```
BMLRealizer.BMLBml.BMLBml ( )
```

constructor

Definition at line 57 of file BMLBml.cs.

### 5.3.4 Member Function Documentation

#### 5.3.4.1 Parse()

```
override void BMLRealizer.BMLBml.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml

Parameters

<i>reader</i>	
---------------	--

Implements [BMLRealizer.BMLBlock](#).

Definition at line 74 of file BMLBml.cs.

### 5.3.5 Member Data Documentation

#### 5.3.5.1 characterId

```
string BMLRealizer.BMLBml.characterId
```

a reference towards the controlled character

Definition at line 31 of file BMLBml.cs.

#### 5.3.5.2 composition

```
Composition BMLRealizer.BMLBml.composition
```

one among [MERGE,APPEND,REPLACE], defines the composition policy to apply if the current <bml> block overlaps with previous <bml> blocks (see below).

Definition at line 38 of file BMLBml.cs.

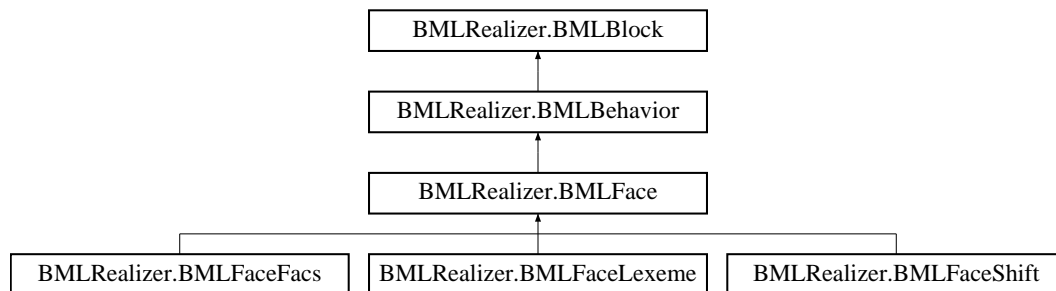
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BML↔Realizer/BMLBml.cs

## 5.4 BMLRealizer.BMLFace Class Reference

Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit.

Inheritance diagram for BMLRealizer.BMLFace:



### Public Member Functions

- [BMLFace](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: id, amount, overshoot synx attribute: start, attackPeak, relax, end*

### Public Attributes

- float [amount](#)  
*A float value between 0..1 to indicate the amount to which the expression should be shown on the face, 0 meaning 'not at all' and 1 meaning 'maximum, highly exaggerated'*
- float [overshoot](#)  
*Fraction of overshoot of the attack peak, relative to amount (which defines the level of the sustain phase)*

### Additional Inherited Members

#### 5.4.1 Detailed Description

Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit.

Definition at line 24 of file BMLFace.cs.

#### 5.4.2 Constructor & Destructor Documentation

##### 5.4.2.1 BMLFace()

```
BMLRealizer.BMLFace.BMLFace ( )
```

constructor

Definition at line 39 of file BMLFace.cs.

#### 5.4.3 Member Function Documentation

##### 5.4.3.1 Parse()

```
override void BMLRealizer.BMLFace.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id, amount, overshoot synx attribute: start, attackPeak, relax, end

**Parameters**

<i>reader</i>	
---------------	--

XMLReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Reimplemented in [BMLRealizer.BMLFaceLexeme](#).

Definition at line 51 of file BMLFace.cs.

## 5.4.4 Member Data Documentation

### 5.4.4.1 amount

```
float BMLRealizer.BMLFace.amount
```

A float value between 0..1 to indicate the amount to which the expression should be shown on the face, 0 meaning 'not at all' and 1 meaning 'maximum, highly exaggerated'

Definition at line 29 of file BMLFace.cs.

### 5.4.4.2 overshoot

```
float BMLRealizer.BMLFace.overshoot
```

Fraction of overshoot of the attack peak, relative to amount (which defines the level of the sustain phase)

Definition at line 34 of file BMLFace.cs.

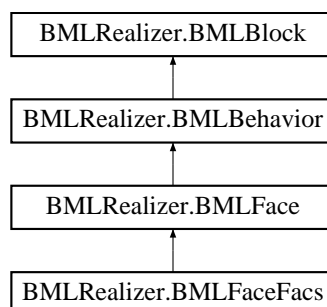
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLFace.cs

## 5.5 BMLRealizer.BMLFaceFacs Class Reference

This behavior provides control of the face through single Action Units from the Facial Action Coding Scheme. It is an Core Extension, that is, not every BML Compliant Realizer has to implement this behavior, but if a Realizer offers FACS based face control, they should adhere to the specification of this <faceFacs> behavior

Inheritance diagram for BMLRealizer.BMLFaceFacs:



## Public Types

- enum [Side](#) { **LEFT**, **RIGHT**, **BOTH** }

## Public Member Functions

- [BMLFaceFacs](#) ()

*Constructor*

## Public Attributes

- int [au](#)

*The number of the FACS Action Unit to be displayed*

- [Side](#) *side*

*Which side of the face to display the action unit on. Possible values: [LEFT,RIGHT,BOTH] Note that for some Action Units, BOTH is the only possible value*

## Additional Inherited Members

### 5.5.1 Detailed Description

This behavior provides control of the face through single Action Units from the Facial Action Coding Scheme. It is an Core Extension, that is, not every BML Compliant Realizer has to implement this behavior, but if a Realizer offers FACS based face control, they should adhere to the specification of this <faceFacs> behavior

Definition at line 22 of file BMLFaceFacs.cs.

### 5.5.2 Member Enumeration Documentation

#### 5.5.2.1 Side

```
enum BMLRealizer.BMLFaceFacs.Side [strong]
```

Definition at line 37 of file BMLFaceFacs.cs.

### 5.5.3 Constructor & Destructor Documentation

#### 5.5.3.1 BMLFaceFacs()

```
BMLRealizer.BMLFaceFacs.BMLFaceFacs ( )
```

Constructor

Definition at line 47 of file BMLFaceFacs.cs.

## 5.5.4 Member Data Documentation

### 5.5.4.1 au

```
int BMLRealizer.BMLFaceFacs.au
```

The number of the FACS Action Unit to be displayed

Definition at line 27 of file BMLFaceFacs.cs.

### 5.5.4.2 side

```
Side BMLRealizer.BMLFaceFacs.side
```

Which side of the face to display the action unit on. Possible values: [LEFT,RIGHT,BOTH] Note that for some Action Units, BOTH is the only possible value

Definition at line 32 of file BMLFaceFacs.cs.

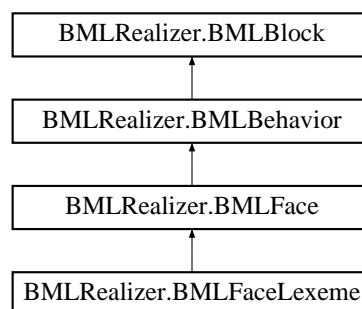
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLFaceFacs.cs

## 5.6 BMLRealizer.BMLFaceLexeme Class Reference

Show a (partial) face expression from a predefined lexicon.

Inheritance diagram for BMLRealizer.BMLFaceLexeme:



### Public Types

- enum [Lexeme](#) {  
**NONE, OBLIQUE\_BROWS, RAISE\_BROWS, RAISE\_LEFT\_BROW, RAISE\_RIGHT\_BROW, LOWER\_BROWS, LOWER\_LEFT\_BROW, LOWER\_RIGHT\_BROW, LOWER\_MOUTH\_CORNERS, LOWER\_LEFT\_MOUTH\_CORNER, LOWER\_RIGHT\_MOUTH\_CORNER, RAISE\_MOUTH\_CORNERS, RAISE\_RIGHT\_MOUTH\_CORNER, RAISE\_LEFT\_MOUTH\_CORNER, OPEN\_MOUTH, OPEN\_LIPS, WIDEN\_EYES, CLOSE\_EYES** }

## Public Member Functions

- [BMLFaceLexeme](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: lexeme*

## Public Attributes

- [Lexeme](#) *lexeme*

## Additional Inherited Members

### 5.6.1 Detailed Description

Show a (partial) face expression from a predefined lexicon.

Definition at line 24 of file BMLFaceLexeme.cs.

### 5.6.2 Member Enumeration Documentation

#### 5.6.2.1 Lexeme

```
enum BMLRealizer.BMLFaceLexeme.Lexeme [strong]
```

Definition at line 35 of file BMLFaceLexeme.cs.

### 5.6.3 Constructor & Destructor Documentation

#### 5.6.3.1 BMLFaceLexeme()

```
BMLRealizer.BMLFaceLexeme.BMLFaceLexeme ( )
```

constructor

Definition at line 60 of file BMLFaceLexeme.cs.

### 5.6.4 Member Function Documentation

#### 5.6.4.1 Parse()

```
override void BMLRealizer.BMLFaceLexeme.Parse (  
    XmlReader reader ) [virtual]
```

parsing the xml attribute: lexeme

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLFace](#).

Definition at line 70 of file BMLFaceLexeme.cs.

## 5.6.5 Member Data Documentation

### 5.6.5.1 lexeme

[Lexeme](#) `BMLRealizer.BMLFaceLexeme.lexeme`

Definition at line 30 of file BMLFaceLexeme.cs.

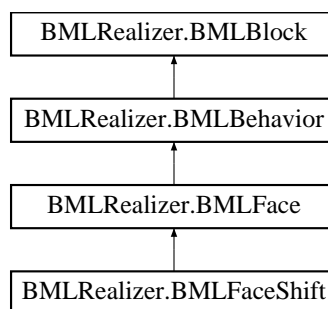
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLFaceLexeme.cs

## 5.7 BMLRealizer.BMLFaceShift Class Reference

Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit, where the specified compound face expression becomes the new BASE state of the ECAs face.

Inheritance diagram for BMLRealizer.BMLFaceShift:



### Additional Inherited Members

#### 5.7.1 Detailed Description

Compound behavior to specify the timing and alignment of several (partial) face expressions as one unit, where the specified compound face expression becomes the new BASE state of the ECAs face.

Definition at line 22 of file BMLFaceShift.cs.

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

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLFaceShift.cs



## 5.8 BMLRealizer.BMLFeedback Class Reference

### 5.8.1 Detailed Description

Definition at line 19 of file BMLFeedback.cs.

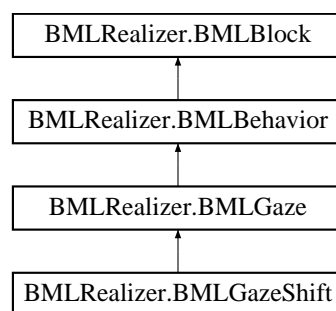
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLFeedback.cs

## 5.9 BMLRealizer.BMLGaze Class Reference

Temporarily directs the gaze of the character towards a target. This behavior causes the character to temporarily direct its gaze to the requested target. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.

Inheritance diagram for BMLRealizer.BMLGaze:



### Public Types

- enum [Influence](#) {  
**NONE, EYES, HEAD, SHOULDER, WAIST, WHOLE** }  
*Determines what parts of the body to move to effect the gaze direction.*
- enum [Direction](#) {  
**RIGHT, LEFT, UP, DOWN, UPRIGHT, UPLEFT, DOWNLEFT, DOWNRIGHT** }  
*Direction of the offsetDirection angle.*

### Public Member Functions

- [BMLGaze](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: target, influence, offsetAngle, offsetDirection sync point: start, ready, relax, end*

## Public Attributes

- string [target](#)  
*A reference towards a target instance that represents the target direction of the gaze.*
- [Influence influence](#)  
*Determines what parts of the body to move to effect the gaze direction.*
- float [offsetAngle](#)  
*Adds an angle degrees offset to gaze direction relative to the target in the direction specified in the offsetDirection*
- [Direction offsetDirection](#)  
*Direction of the offsetDirection angle*

## Additional Inherited Members

### 5.9.1 Detailed Description

Temporarily directs the gaze of the character towards a target. This behavior causes the character to temporarily direct its gaze to the requested target. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.

Definition at line 26 of file BMLGaze.cs.

### 5.9.2 Member Enumeration Documentation

#### 5.9.2.1 Direction

```
enum BMLRealizer.BMLGaze.Direction [strong]
```

Direction of the offsetDirection angle.

Definition at line 64 of file BMLGaze.cs.

#### 5.9.2.2 Influence

```
enum BMLRealizer.BMLGaze.Influence [strong]
```

Determines what parts of the body to move to effect the gaze direction.

Definition at line 51 of file BMLGaze.cs.

### 5.9.3 Constructor & Destructor Documentation

#### 5.9.3.1 BMLGaze()

```
BMLRealizer.BMLGaze.BMLGaze ( )
```

constructor

Definition at line 79 of file BMLGaze.cs.

### 5.9.4 Member Function Documentation

#### 5.9.4.1 Parse()

```
override void BMLRealizer.BMLGaze.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: target, influence, offsetAngle, offsetDirection sync point: start, ready, relax, end

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Reimplemented in [BMLRealizer.BMLGazeShift](#).

Definition at line 91 of file BMLGaze.cs.

## 5.9.5 Member Data Documentation

### 5.9.5.1 influence

[Influence](#) `BMLRealizer.BMLGaze.influence`

Determines what parts of the body to move to effect the gaze direction.

Definition at line 36 of file BMLGaze.cs.

### 5.9.5.2 offsetAngle

`float BMLRealizer.BMLGaze.offsetAngle`

Adds an angle degrees offset to gaze direction relative to the target in the direction specified in the `offsetDirection`

Definition at line 41 of file BMLGaze.cs.

### 5.9.5.3 offsetDirection

[Direction](#) `BMLRealizer.BMLGaze.offsetDirection`

Direction of the `offsetDirection` angle

Definition at line 46 of file BMLGaze.cs.

### 5.9.5.4 target

`string BMLRealizer.BMLGaze.target`

A reference towards a target instance that represents the target direction of the gaze.

Definition at line 31 of file BMLGaze.cs.

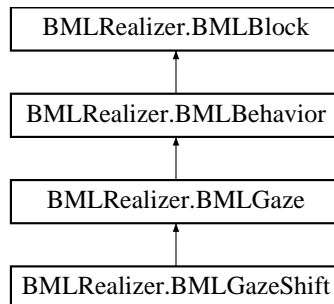
The documentation for this class was generated from the following file:

- `C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLGaze.cs`

## 5.10 BMLRealizer.BMLGazeShift Class Reference

Permanently change the gaze direction of the character towards a certain target. This behavior causes the character to direct its gaze to the requested target. This changes the default state of the ECA: after completing this behavior, the new target is the default gaze direction of the ECA. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.

Inheritance diagram for BMLRealizer.BMLGazeShift:



### Public Member Functions

- [BMLGazeShift](#) (  
*constructor sync attribute: start, end*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: sync point: start, end*

### Additional Inherited Members

#### 5.10.1 Detailed Description

Permanently change the gaze direction of the character towards a certain target. This behavior causes the character to direct its gaze to the requested target. This changes the default state of the ECA: after completing this behavior, the new target is the default gaze direction of the ECA. The influence parameter is read as follows: EYE means 'use only the eyes'; HEAD means 'use only head and eyes to change the gaze direction', etcetera.

Definition at line 25 of file BMLGazeShift.cs.

#### 5.10.2 Constructor & Destructor Documentation

##### 5.10.2.1 BMLGazeShift()

```
BMLRealizer.BMLGazeShift.BMLGazeShift ( )
```

constructor sync attribute: start, end

Definition at line 31 of file BMLGazeShift.cs.

#### 5.10.3 Member Function Documentation

##### 5.10.3.1 Parse()

```
override void BMLRealizer.BMLGazeShift.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: sync point: start, end

## Parameters

<i>reader</i>	
---------------	--

XMLReader

Reimplemented from [BMLRealizer.BMLGaze](#).

Definition at line 42 of file BMLGazeShift.cs.

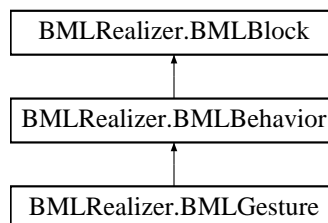
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLGazeShift.cs

## 5.11 BMLRealizer.BMLGesture Class Reference

Currently, BML offers two types of gesture behaviors. The first provides a set of gestures recalled by name from a gesticon; the second provides simple pointing gestures. Coordinated movement with arms and hands, recalled from a gesticon by requesting the corresponding lexeme

Inheritance diagram for BMLRealizer.BMLGesture:



### Public Types

- enum [Mode](#) { **NONE**, **LEFT\_HAND**, **RIGHT\_HAND**, **BOTH\_HANDS** }  
*What hand/arm is being used*
- enum [Lexeme](#) { **BEAT** }  
*Refers to an animation or a controller to realize this particular gesture.*

### Public Member Functions

- [BMLGesture](#) ()  
*constructor*
- override void [Parse](#) (XMLReader reader)  
*parsing the xml attribute: mode, lexeme sync point: start, ready, strokeStart, stroke, strokeEnd, relax, end*

## Public Attributes

- [Mode mode](#)  
*What hand/arm is being used*
- [Lexeme lexeme](#)  
*Refers to an animation or a controller to realize this particular gesture.*

## Additional Inherited Members

### 5.11.1 Detailed Description

Currently, BML offers two types of gesture behaviors. The first provides a set of gestures recalled by name from a gesticon; the second provides simple pointing gestures. Coordinated movement with arms and hands, recalled from a gesticon by requesting the corresponding lexeme

Definition at line 25 of file BMLGesture.cs.

### 5.11.2 Member Enumeration Documentation

#### 5.11.2.1 Lexeme

```
enum BMLRealizer.BMLGesture.Lexeme [strong]
```

Refers to an animation or a controller to realize this particular gesture.

Definition at line 51 of file BMLGesture.cs.

#### 5.11.2.2 Mode

```
enum BMLRealizer.BMLGesture.Mode [strong]
```

What hand/arm is being used

Definition at line 40 of file BMLGesture.cs.

### 5.11.3 Constructor & Destructor Documentation

#### 5.11.3.1 BMLGesture()

```
BMLRealizer.BMLGesture.BMLGesture ( )
```

constructor

Definition at line 59 of file BMLGesture.cs.

### 5.11.4 Member Function Documentation

#### 5.11.4.1 Parse()

```
override void BMLRealizer.BMLGesture.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: mode, lexeme sync point: start, ready, strokeStart, stroke, strokeEnd, relax, end

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 70 of file BMLGesture.cs.

## 5.11.5 Member Data Documentation

### 5.11.5.1 lexeme

[Lexeme](#) `BMLRealizer.BMLGesture.lexeme`

Refers to an animation or a controller to realize this particular gesture.

Definition at line 35 of file BMLGesture.cs.

### 5.11.5.2 mode

[Mode](#) `BMLRealizer.BMLGesture.mode`

What hand/arm is being used

Definition at line 30 of file BMLGesture.cs.

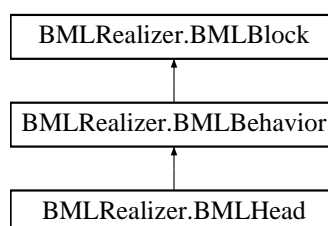
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLGesture.cs

## 5.12 BMLRealizer.BMLHead Class Reference

Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. The stroke phase of the head motion (from strokeStart till strokeEnd is the "meaningful" part of the head motion. The stroke sync point is the "peak" moment of the motion. If repetition > 1, the meaning of the stroke sync point becomes undefined

Inheritance diagram for BMLRealizer.BMLHead:



## Public Types

- enum [Lexeme](#) { **NONE**, **NOD**, **SHAKE** }

*Refers to an animation or a controller to realize this particular head behavior. Minimum set offered by all realizers: [NOD, SHAKE]*

## Public Member Functions

- [BMLHead](#) ()

*constructor*

- override void [Parse](#) (XmlReader reader)

*parsing the xml attribute: id, amount, overshoot sync attribute: start, ready, strokeStart, stroke, strokeEnd, relax, end*

## Public Attributes

- [Lexeme](#) [lexeme](#)

*Refers to an animation or a controller to realize this particular head behavior. Minimum set offered by all realizers: [NOD, SHAKE]*

- int [repetition](#)

*Number of times the basic head motion is repeated.*

- float [amount](#)

*How intense is the head nod? 0 means immeasurable small; 0.5 means "moderate"; 1 means maximally large*

## Additional Inherited Members

### 5.12.1 Detailed Description

Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. Movement of the head, recalled from a gesticon by requesting the corresponding lexeme. The stroke phase of the head motion (from strokeStart till strokeEnd is the "meaningful" part of the head motion. The stroke sync point is the "peak" moment of the motion. If repetition > 1, the meaning of the stroke sync point becomes undefined

Definition at line 25 of file BMLHead.cs.

### 5.12.2 Member Enumeration Documentation

#### 5.12.2.1 Lexeme

```
enum BMLRealizer.BMLHead.Lexeme [strong]
```

Refers to an animation or a controller to realize this particular head behavior. Minimum set offered by all realizers: [NOD, SHAKE]

Definition at line 45 of file BMLHead.cs.



### 5.12.3 Constructor & Destructor Documentation

#### 5.12.3.1 BMLHead()

```
BMLRealizer.BMLHead.BMLHead ( )
```

constructor

Definition at line 55 of file BMLHead.cs.

### 5.12.4 Member Function Documentation

#### 5.12.4.1 Parse()

```
override void BMLRealizer.BMLHead.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id, amount, overshoot sync attribute: start, ready, strokeStart, stroke, strokeEnd, relax, end

Parameters

<i>reader</i>	
---------------	--

XMLReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 65 of file BMLHead.cs.

### 5.12.5 Member Data Documentation

#### 5.12.5.1 amount

```
float BMLRealizer.BMLHead.amount
```

How intense is the head nod? 0 means immeasurable small; 0.5 means "moderate"; 1 means maximally large

Definition at line 40 of file BMLHead.cs.

#### 5.12.5.2 lexeme

```
Lexeme BMLRealizer.BMLHead.lexeme
```

Refers to an animation or a controller to realize this particular head behavior. Minimum set offered by all realizers: [NOD, SHAKE]

Definition at line 30 of file BMLHead.cs.

### 5.12.5.3 repetition

```
int BMLRealizer.BMLHead.repetition
```

Number of times the basic head motion is repeated.

Definition at line 35 of file BMLHead.cs.

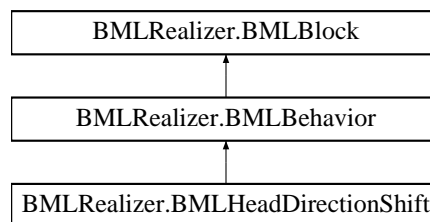
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLHead.cs

## 5.13 BMLRealizer.BMLHeadDirectionShift Class Reference

Orient the head towards a target referenced by the target attribute. Permanently orient the head in a certain direction.

Inheritance diagram for BMLRealizer.BMLHeadDirectionShift:



### Public Member Functions

- [BMLHeadDirectionShift](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: id, amount, overshoot sync attribute: start, end*

### Public Attributes

- string [target](#)  
*target towards which the head is oriented*

### Additional Inherited Members

### 5.13.1 Detailed Description

Orient the head towards a target referenced by the target attribute. Permanently orient the head in a certain direction.

Definition at line 25 of file BMLHeadDirectionShift.cs.

## 5.13.2 Constructor & Destructor Documentation

### 5.13.2.1 BMLHeadDirectionShift()

```
BMLRealizer.BMLHeadDirectionShift.BMLHeadDirectionShift ( )
```

constructor

Definition at line 35 of file BMLHeadDirectionShift.cs.

## 5.13.3 Member Function Documentation

### 5.13.3.1 Parse()

```
override void BMLRealizer.BMLHeadDirectionShift.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id, amount, overshoot sync attribute: start, end

Parameters

<i>reader</i>	
---------------	--

XMLReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 46 of file BMLHeadDirectionShift.cs.

## 5.13.4 Member Data Documentation

### 5.13.4.1 target

```
string BMLRealizer.BMLHeadDirectionShift.target
```

target towards which the head is oriented

Definition at line 30 of file BMLHeadDirectionShift.cs.

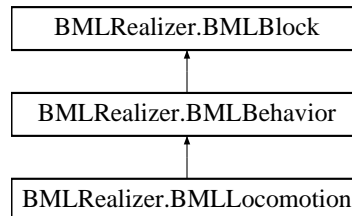
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLHeadDirectionShift.cs

## 5.14 BMLRealizer.BMLLocomotion Class Reference

Move the body of the character from one location to another. This behavior causes the character to move to the requested target in the manner described.

Inheritance diagram for BMLRealizer.BMLLocomotion:



### Public Member Functions

- [BMLLocomotion](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing xml attribute: id, target, manner sync point: start, end*

### Public Attributes

- string [target](#)  
*A reference towards a target instance that represents the end location of the locomotive behavior.*
- string [manner](#)  
*The general manner of locomotion [WALK, RUN, STRAFE ...] (WALK is the only mandatory element in the set)*

### Additional Inherited Members

#### 5.14.1 Detailed Description

Move the body of the character from one location to another. This behavior causes the character to move to the requested target in the manner described.

Definition at line 25 of file BMLLocomotion.cs.

#### 5.14.2 Constructor & Destructor Documentation

##### 5.14.2.1 BMLLocomotion()

```
BMLRealizer.BMLLocomotion.BMLLocomotion ( )
```

constructor

Definition at line 40 of file BMLLocomotion.cs.

#### 5.14.3 Member Function Documentation

##### 5.14.3.1 Parse()

```
override void BMLRealizer.BMLLocomotion.Parse (
    XmlReader reader ) [virtual]
```

parsing xml attribute: id, target, manner sync point: start, end

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 51 of file BMLLocomotion.cs.

## 5.14.4 Member Data Documentation

### 5.14.4.1 manner

```
string BMLRealizer.BMLLocomotion.manner
```

The general manner of locomotion [WALK, RUN, STRAFE ...] (WALK is the only mandatory element in the set)

Definition at line 35 of file BMLLocomotion.cs.

### 5.14.4.2 target

```
string BMLRealizer.BMLLocomotion.target
```

A reference towards a target instance that represents the end location of the locomotive behavior.

Definition at line 30 of file BMLLocomotion.cs.

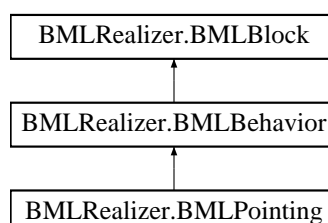
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLLocomotion.cs

## 5.15 BMLRealizer.BMLPointing Class Reference

Deictic gesture towards the target specified by the target attribute

Inheritance diagram for BMLRealizer.BMLPointing:



## Public Types

- enum [Mode](#) { **NONE**, **LEFT\_HAND**, **RIGHT\_HAND**, **BOTH\_HANDS** }  
*What hand/arm is being used*

## Public Member Functions

- [BMLPointing](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml attribute: id, target, mode sync point: start, ready, strokeStart, stroke, strokeEnd, relax, end*

## Public Attributes

- [Mode](#) [mode](#)  
*What hand/arm is being used*
- string [target](#)  
*The gesture is directed towards this target*

## Additional Inherited Members

### 5.15.1 Detailed Description

Deictic gesture towards the target specified by the target attribute

Definition at line 24 of file BMLPointing.cs.

### 5.15.2 Member Enumeration Documentation

#### 5.15.2.1 Mode

```
enum BMLRealizer.BMLPointing.Mode [strong]
```

What hand/arm is being used

Definition at line 39 of file BMLPointing.cs.

### 5.15.3 Constructor & Destructor Documentation

#### 5.15.3.1 BMLPointing()

```
BMLRealizer.BMLPointing.BMLPointing ( )
```

constructor

Definition at line 50 of file BMLPointing.cs.

### 5.15.4 Member Function Documentation

#### 5.15.4.1 Parse()

```
override void BMLRealizer.BMLPointing.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id, target, mode sync point: start, ready, strokeStart, stroke, strokeEnd, relax, end

## Parameters

<i>reader</i>	
---------------	--

XMLReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 61 of file BMLPointing.cs.

## 5.15.5 Member Data Documentation

## 5.15.5.1 mode

[Mode](#) BMLRealizer.BMLPointing.mode

What hand/arm is being used

Definition at line 29 of file BMLPointing.cs.

## 5.15.5.2 target

string BMLRealizer.BMLPointing.target

The gesture is directed towards this target

Definition at line 34 of file BMLPointing.cs.

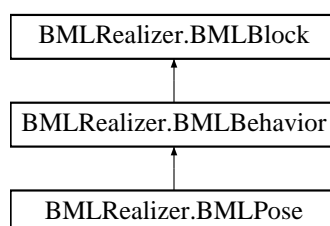
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLPointing.cs

## 5.16 BMLRealizer.BMLPose Class Reference

Child element of <posture> and <postureShift> behaviors, defines additions to the global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines additions that modify the global body posture of the ECA. For each value of the part attribute, only one <pose> child is expected to be present. A BML Realizer may define any number of lexemes beyond the ones specified above.

Inheritance diagram for BMLRealizer.BMLPose:



## Public Types

- enum [Part](#) {  
**NONE, ARMS, LEFT\_ARM, RIGHT\_ARM,**  
**LEGS, LEFT\_LEG, RIGHT\_LEG, HEAD,**  
**WHOLEBODY }**  
*What part of the body is affected? Possible values are [ARMS, LEFT\_ARM, RIGHT\_ARM, LEGS, LEFT\_LEG, RIGHT\_LEG, HEAD, WHOLEBODY]*
- enum [Lexeme](#) {  
**NONE, ARMS\_AKIMBO, ARMS\_CROSSED, ARMS\_NEUTRAL,**  
**ARMS\_OPEN, LEGS\_CROSSED, LEGS\_NEUTRAL, LEGS\_OPEN,**  
**LEANING\_FORWARD, LEANING\_BACKWARD }**  
*What configuration is set to the given part? Some possible values are [ARMS\_AKIMBO, ARMS\_CROSSED, ARMS\_NEUTRAL, ARMS\_OPEN, LEGS\_CROSSED, LEGS\_NEUTRAL, LEGS\_OPEN, LEANING\_FORWARD, LEANING\_BACKWARD, ...]*

## Public Member Functions

- [BMLPose](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing xml attribute: part, lexeme*

## Public Attributes

- [Part](#) part  
*What part of the body is affected? Possible values are [ARMS, LEFT\_ARM, RIGHT\_ARM, LEGS, LEFT\_LEG, RIGHT\_LEG, HEAD, WHOLEBODY]*
- [Lexeme](#) lexeme  
*What configuration is set to the given part? Some possible values are [ARMS\_AKIMBO, ARMS\_CROSSED, ARMS\_NEUTRAL, ARMS\_OPEN, LEGS\_CROSSED, LEGS\_NEUTRAL, LEGS\_OPEN, LEANING\_FORWARD, LEANING\_BACKWARD, ...]*

## Additional Inherited Members

### 5.16.1 Detailed Description

Child element of <posture> and <postureShift> behaviors, defines additions to the global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines additions that modify the global body posture of the ECA. For each value of the part attribute, only one <pose> child is expected to be present. A BML Realizer may define any number of lexemes beyond the ones specified above.

Definition at line 25 of file BMLPose.cs.

### 5.16.2 Member Enumeration Documentation

#### 5.16.2.1 Lexeme

```
enum BMLRealizer, BMLPose.Lexeme [strong]
```

What configuration is set to the given part? Some possible values are [ARMS\_AKIMBO, ARMS\_CROSSED, ARMS\_NEUTRAL, ARMS\_OPEN, LEGS\_CROSSED, LEGS\_NEUTRAL, LEGS\_OPEN, LEANING\_FORWARD, LEANING\_BACKWARD, ...]

Definition at line 56 of file BMLPose.cs.



### 5.16.2.2 Part

```
enum BMLRealizer.BMLPose.Part [strong]
```

What part of the body is affected? Possible values are [ARMS, LEFT\_ARM, RIGHT\_ARM, LEGS, LEFT\_LEG, RIGHT\_LEG, HEAD, WHOLEBODY]

Definition at line 40 of file BMLPose.cs.

## 5.16.3 Constructor & Destructor Documentation

### 5.16.3.1 BMLPose()

```
BMLRealizer.BMLPose.BMLPose ( )
```

constructor

Definition at line 73 of file BMLPose.cs.

## 5.16.4 Member Function Documentation

### 5.16.4.1 Parse()

```
override void BMLRealizer.BMLPose.Parse (
    XmlReader reader ) [virtual]
```

parsing xml attribute: part, lexeme

Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 83 of file BMLPose.cs.

## 5.16.5 Member Data Documentation

### 5.16.5.1 lexeme

[Lexeme](#) BMLRealizer.BMLPose.lexeme

What configuration is set to the given part? Some possible values are [ARMS\_AKIMBO, ARMS\_CROSSED, ARMS\_NEUTRAL, ARMS\_OPEN, LEGS\_CROSSED, LEGS\_NEUTRAL, LEGS\_OPEN, LEANING\_FORWARD, LEANING\_BACKWARD, ...]

Definition at line 35 of file BMLPose.cs.

### 5.16.5.2 part

`Part BMLRealizer.BMLPose.part`

What part of the body is affected? Possible values are [ARMS, LEFT\_ARM, RIGHT\_ARM, LEGS, LEFT\_LEG, RIGHT\_LEG, HEAD, WHOLEBODY]

Definition at line 30 of file BMLPose.cs.

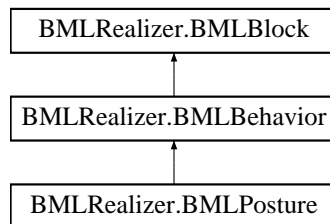
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLPose.cs

## 5.17 BMLRealizer.BMLPosture Class Reference

Temporarily change the posture of the ECA. Temporarily change the posture of the ECA. After the <posture> behavior has ended, return to the BASE posture.

Inheritance diagram for BMLRealizer.BMLPosture:



### Public Member Functions

- `BMLPosture()`  
*constructor*
- `override void Parse(XmlReader reader)`  
*parsing xml attribute: id sync point: start, ready, relax, end*

### Additional Inherited Members

#### 5.17.1 Detailed Description

Temporarily change the posture of the ECA. Temporarily change the posture of the ECA. After the <posture> behavior has ended, return to the BASE posture.

Definition at line 25 of file BMLPosture.cs.

## 5.17.2 Constructor & Destructor Documentation

### 5.17.2.1 BMLPosture()

BMLRealizer.BMLPosture.BMLPosture ( )

constructor

Definition at line 30 of file BMLPosture.cs.

## 5.17.3 Member Function Documentation

### 5.17.3.1 Parse()

```
override void BMLRealizer.BMLPosture.Parse (
    XmlReader reader ) [virtual]
```

parsing xml attribute: id sync point: start, ready, relax, end

Parameters

<i>reader</i>	
---------------	--

XmlReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 41 of file BMLPosture.cs.

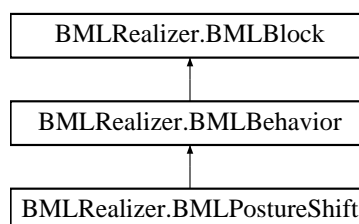
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLPosture.cs

## 5.18 BMLRealizer.BMLPostureShift Class Reference

Permanently change the BASE posture of the ECA.

Inheritance diagram for BMLRealizer.BMLPostureShift:



## Public Member Functions

- [BMLPostureShift](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing xml attribute: id sync point: start, end*

## Additional Inherited Members

### 5.18.1 Detailed Description

Permanently change the BASE posture of the ECA.

Definition at line 24 of file BMLPostureShift.cs.

### 5.18.2 Constructor & Destructor Documentation

#### 5.18.2.1 BMLPostureShift()

```
BMLRealizer.BMLPostureShift.BMLPostureShift ( )
```

constructor

Definition at line 29 of file BMLPostureShift.cs.

### 5.18.3 Member Function Documentation

#### 5.18.3.1 Parse()

```
override void BMLRealizer.BMLPostureShift.Parse (
    XmlReader reader ) [virtual]
```

parsing xml attribute: id sync point: start, end

Parameters

<i>reader</i>	
---------------	--

XmlReader

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 40 of file BMLPostureShift.cs.

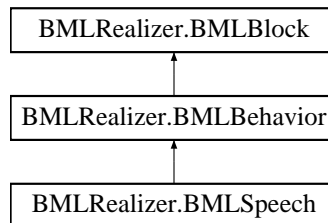
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLPostureShift.cs

## 5.19 BMLRealizer.BMLSpeech Class Reference

Utterance to be spoken by a character. Realization of the `<speech>` element generates both speech audio (or text) and speech movement, for example using a speech synthesizer and viseme morphing. The `<speech>` element requires a sub-element. This sub-element is a `<text>` element that contains the text to be spoken, with optionally embedded `<sync>` elements for alignment with other behaviors.

Inheritance diagram for BMLRealizer.BMLSpeech:



### Public Member Functions

- [BMLSpeech](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the xml child node: text sync point: start, end*

### Public Attributes

- string [text](#)  
*the text that need to be spoken*

### Additional Inherited Members

#### 5.19.1 Detailed Description

Utterance to be spoken by a character. Realization of the `<speech>` element generates both speech audio (or text) and speech movement, for example using a speech synthesizer and viseme morphing. The `<speech>` element requires a sub-element. This sub-element is a `<text>` element that contains the text to be spoken, with optionally embedded `<sync>` elements for alignment with other behaviors.

Definition at line 29 of file BMLSpeech.cs.

#### 5.19.2 Constructor & Destructor Documentation

##### 5.19.2.1 BMLSpeech()

```
BMLRealizer.BMLSpeech.BMLSpeech ( )
```

constructor

Definition at line 39 of file BMLSpeech.cs.

### 5.19.3 Member Function Documentation

#### 5.19.3.1 Parse()

```
override void BMLRealizer.BMLSpeech.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml child node: text sync point: start, end

##### Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 50 of file BMLSpeech.cs.

### 5.19.4 Member Data Documentation

#### 5.19.4.1 text

```
string BMLRealizer.BMLSpeech.text
```

the text that need to be spoken

Definition at line 34 of file BMLSpeech.cs.

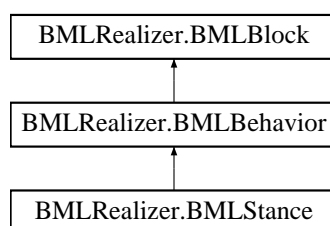
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLSpeech.cs

## 5.20 BMLRealizer.BMLStance Class Reference

Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. This global posture may then be modified by one or more <pose> siblings.

Inheritance diagram for BMLRealizer.BMLStance:



## Public Types

- enum [Type](#) { **SITTING**, **CROUCHING**, **STANDING**, **LYING** }  
Global body posture. Possible values are [SITTING, CROUCHING, STANDING, LYING]

## Public Member Functions

- [BMLStance](#) ()  
constructor
- override void [Parse](#) (XmlReader reader)  
parsing the xml attribute: id

## Public Attributes

- [Type](#) type  
Global body posture. Possible values are [SITTING, CROUCHING, STANDING, LYING]

## Additional Inherited Members

### 5.20.1 Detailed Description

Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. Child element of <posture> and <postureShift> behaviors, defines global body posture of the ECA. This global posture may then be modified by one or more <pose> siblings.

Definition at line 25 of file BMLStance.cs.

### 5.20.2 Member Enumeration Documentation

#### 5.20.2.1 Type

```
enum BMLRealizer.BMLStance.Type [strong]
```

Global body posture. Possible values are [SITTING, CROUCHING, STANDING, LYING]

Definition at line 35 of file BMLStance.cs.

### 5.20.3 Constructor & Destructor Documentation

#### 5.20.3.1 BMLStance()

```
BMLRealizer.BMLStance.BMLStance ( )
```

constructor

Definition at line 46 of file BMLStance.cs.

### 5.20.4 Member Function Documentation

#### 5.20.4.1 Parse()

```
override void BMLRealizer.BMLStance.Parse (
    XmlReader reader ) [virtual]
```

parsing the xml attribute: id

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 51 of file BMLStance.cs.

## 5.20.5 Member Data Documentation

### 5.20.5.1 type

Type `BMLRealizer.BMLStance.type`

Global body posture. Possible values are [SITTING, CROUCHING, STANDING, LYING]

Definition at line 30 of file BMLStance.cs.

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

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLStance.cs

## 5.21 BMLRealizer.BMLSyncPoint Class Reference

BML Sync Point class possible format: behavior\_id:sync\_id [+/- offset] A reference to a sync point of another behavior, optionally with a float offset in seconds. By default, this is a behavior in the same <bml> block that the syncref is contained in; if optional prefix block\_id: is present, the syncref specifies a sync point of a behavior in the <bml> block with that ID.) offset: A positive float offset in seconds relative to the start time of the surrounding <bml> block.

### Public Member Functions

- [BMLSyncPoint](#) ([BMLBlock](#) parent, string eventName, string value)  
*the constructor of BML Sync Point.*
- void [Update](#) (RageBMLNet bmlNet)  
*function that need to be called everytime the realizer update is called*
- bool [TriggerSyncPoint](#) ()  
*trigger this syncpoint to complete*
- bool [IsCompleted](#) ()  
*is this syncpoint already completed ?*
- bool [IsTimerSafe](#) (Dictionary< string, [BMLBlock](#) > blocks, float globalTimer)  
*function to check whether the timer variable is safe to used or not*



### 5.21.1 Detailed Description

BML Sync Point class possible format: behavior\_id:sync\_id [+/- offset] A reference to a sync point of another behavior, optionally with a float offset in seconds. By default, this is a behavior in the same <bml> block that the syncref is contained in; if optional prefix block\_id: is present, the syncref specifies a sync point of a behavior in the <bml> block with that ID.) offset: A positive float offset in seconds relative to the start time of the surrounding <bml> block.

Definition at line 33 of file BMLSyncPoint.cs.

### 5.21.2 Constructor & Destructor Documentation

#### 5.21.2.1 BMLSyncPoint()

```
BMLRealizer.BMLSyncPoint.BMLSyncPoint (
    BMLBlock parent,
    string eventName,
    string value )
```

the constructor of BML Sync Point.

##### Parameters

<i>eventName</i>	
------------------	--

the name of sync point event (start, ready, strokeStart, attackPeak, stroke, strokeEnd, relax, end)

##### Parameters

<i>value</i>	
--------------	--

the attribute value that we need to parse.

Definition at line 76 of file BMLSyncPoint.cs.

### 5.21.3 Member Function Documentation

#### 5.21.3.1 IsCompleted()

```
bool BMLRealizer.BMLSyncPoint.IsCompleted ( )
```

is this syncpoint already completed ?

##### Returns

Definition at line 187 of file BMLSyncPoint.cs.

### 5.21.3.2 IsTimerSafe()

```
bool BMLRealizer.BMLSyncPoint.IsTimerSafe (
    Dictionary< string, BMLBlock > blocks,
    float globalTimer )
```

function to check whether the timer variable is safe to used or not

#### Parameters

<i>realizer</i>	
-----------------	--

#### Returns

Definition at line 197 of file BMLSyncPoint.cs.

### 5.21.3.3 TriggerSyncPoint()

```
bool BMLRealizer.BMLSyncPoint.TriggerSyncPoint ( )
```

trigger this syncpoint to complete

#### Returns

Definition at line 171 of file BMLSyncPoint.cs.

### 5.21.3.4 Update()

```
void BMLRealizer.BMLSyncPoint.Update (
    RageBMLNet bmlNet )
```

function that need to be called everytime the realizer update is called

#### Parameters

<i>realizer</i>	
-----------------	--

The realizer

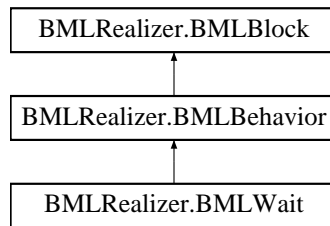
Definition at line 131 of file BMLSyncPoint.cs.

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

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BML↔  
Realizer/BMLSyncPoint.cs

## 5.22 BMLRealizer.BMLWait Class Reference

Inheritance diagram for BMLRealizer.BMLWait:



### Public Member Functions

- [BMLWait](#) ()  
*constructor*
- override void [Parse](#) (XmlReader reader)  
*parsing the XML attribute: duration*

### Public Attributes

- float [duration](#)  
*the duration of the wait in seconds*

### Additional Inherited Members

#### 5.22.1 Detailed Description

Definition at line 21 of file BMLWait.cs.

#### 5.22.2 Constructor & Destructor Documentation

##### 5.22.2.1 BMLWait()

```
BMLRealizer.BMLWait.BMLWait ( )
```

constructor

Definition at line 31 of file BMLWait.cs.

#### 5.22.3 Member Function Documentation

##### 5.22.3.1 Parse()

```
override void BMLRealizer.BMLWait.Parse (
    XmlReader reader ) [virtual]
```

parsing the XML attribute: duration

## Parameters

<i>reader</i>	
---------------	--

Reimplemented from [BMLRealizer.BMLBehavior](#).

Definition at line 41 of file BMLWait.cs.

## 5.22.4 Member Data Documentation

### 5.22.4.1 duration

```
float BMLRealizer.BMLWait.duration
```

the duration of the wait in seconds

Definition at line 26 of file BMLWait.cs.

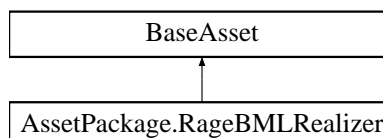
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/BMLRealizer/BMLWait.cs

## 5.23 AssetPackage.RageBMLRealizer Class Reference

An [BMLRealizer](#) Rage asset

Inheritance diagram for AssetPackage.RageBMLRealizer:



### Public Member Functions

- delegate void [SyncPointCompleted](#) (string id, string eventName)  
*callback function. it will be called when the specific sync point is completed*
- [RageBMLRealizer](#) ()  
*Initializes a new instance of the RageBMLRealizer.Asset class.*
- void **ParseFromFile** (string filename)
- void **ParseFromString** (string xml)
- void [Update](#) (float deltaTime)  
*update function will be called everytime when the program is run. it can be called inside Unity Update function*
- void [TriggerSyncPoint](#) (string id, string eventName)  
*this function can be called from outside library to trigger sync point.*
- [BMLBlock](#) [GetBehaviorFromId](#) (string id)  
*function to get behavior from ID*

## Public Attributes

- [SyncPointCompleted](#) **OnSyncPointCompleted**

## Properties

- override **ISettings** [Settings](#) [get, set]  
*Gets or sets options for controlling the operation.*
- Dictionary< string, [BMLBlock](#) > [ScheduledBlocks](#) [get]  
*the dictionary that hold the blocks / behavior that need to be run*
- float [Timer](#) [get]  
*global timer*

### 5.23.1 Detailed Description

An [BMLRealizer](#) Rage asset

Definition at line 37 of file RageBMLRealizer.cs.

### 5.23.2 Constructor & Destructor Documentation

#### 5.23.2.1 RageBMLRealizer()

```
AssetPackage.RageBMLRealizer.RageBMLRealizer ( )
```

Initializes a new instance of the RageBMLRealizer.Asset class.

Create Settings and let it's BaseSettings class assign Defaultvalues where it can.

Definition at line 76 of file RageBMLRealizer.cs.

### 5.23.3 Member Function Documentation

#### 5.23.3.1 GetBehaviorFromId()

```
BMLBlock AssetPackage.RageBMLRealizer.GetBehaviorFromId (
    string id )
```

function to get behavior from ID

#### Parameters

<i>id</i>	
-----------	--

**Returns**

Definition at line 231 of file RageBMLRealizer.cs.

**5.23.3.2 SyncPointCompleted()**

```
delegate void AssetPackage.RageBMLRealizer.SyncPointCompleted (
    string id,
    string eventName )
```

callback function. it will be called when the specific sync point is completed

**Parameters**

<i>id</i>	
-----------	--

the ID of block

**Parameters**

<i>eventName</i>	
------------------	--

the event name of sync point (start, ready, strokeStart, attackPeak, stroke, strokeEnd, relax, end)

**5.23.3.3 TriggerSyncPoint()**

```
void AssetPackage.RageBMLRealizer.TriggerSyncPoint (
    string id,
    string eventName )
```

this function can be called from outside library to trigger sync point.

**Parameters**

<i>id</i>	
-----------	--

the ID of the block where the sync point is resided

**Parameters**

<i>eventName</i>	
------------------	--

the event name of sync point (start, ready, strokeStart, attackPeak, stroke, strokeEnd, relax, end)

Definition at line 211 of file RageBMLRealizer.cs.

#### 5.23.3.4 Update()

```
void AssetPackage.RageBMLRealizer.Update (
    float deltaTime )
```

update function will be called everytime when the program is run. it can be called inside Unity Update function

##### Parameters

<i>deltaTime</i>	
------------------	--

the time from last called

Definition at line 193 of file RageBMLRealizer.cs.

### 5.23.4 Property Documentation

#### 5.23.4.1 ScheduledBlocks

```
Dictionary<string, BMLBlock> AssetPackage.RageBMLRealizer.ScheduledBlocks [get]
```

the dictionary that hold the blocks / behavior that need to be run

Definition at line 150 of file RageBMLRealizer.cs.

#### 5.23.4.2 Settings

```
override ISettings AssetPackage.RageBMLRealizer.Settings [get], [set]
```

Gets or sets options for controlling the operation.

Besides the toXml() and fromXml() methods, we never use this property but use it's correctly typed backing field 'settings' instead.

This property should go into each asset having Settings of its own.

The actual class used should be derived from BaseAsset (and not directly from ISetting).

The settings.

Definition at line 135 of file RageBMLRealizer.cs.

#### 5.23.4.3 Timer

```
float AssetPackage.RageBMLRealizer.Timer [get]
```

global timer

Definition at line 161 of file RageBMLRealizer.cs.

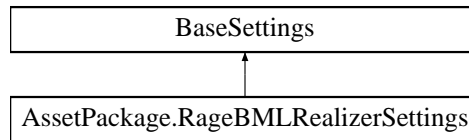
The documentation for this class was generated from the following file:

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/RageBMLRealizer.cs

## 5.24 AssetPackage.RageBMLRealizerSettings Class Reference

An asset settings.

Inheritance diagram for AssetPackage.RageBMLRealizerSettings:



### Public Member Functions

- [RageBMLRealizerSettings](#) ()  
*Initializes a new instance of the RageBMLRealizer.AssetSettings class.*

#### 5.24.1 Detailed Description

An asset settings.

BaseSettings contains the (de-)serialization methods.

Definition at line 28 of file RageBMLRealizerSettings.cs.

#### 5.24.2 Constructor & Destructor Documentation

##### 5.24.2.1 RageBMLRealizerSettings()

```
AssetPackage.RageBMLRealizerSettings.RageBMLRealizerSettings ( )
```

Initializes a new instance of the RageBMLRealizer.AssetSettings class.

Definition at line 35 of file RageBMLRealizerSettings.cs.

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

- C:/Users/Chris021/OneDrive - Universiteit Utrecht/project/rage/BML-Realizer/RageBMLRealizer/RageBMLRealizerSettings.cs



# Index

- amount
  - BMLRealizer::BMLFace, [18](#)
  - BMLRealizer::BMLHead, [31](#)
- AssetPackage, [7](#)
- AssetPackage.RageBMLRealizer, [50](#)
- AssetPackage.RageBMLRealizerSettings, [54](#)
- AssetPackage::RageBMLRealizer
  - GetBehaviorFromId, [51](#)
  - RageBMLRealizer, [51](#)
  - ScheduledBlocks, [53](#)
  - Settings, [53](#)
  - SyncPointCompleted, [52](#)
  - Timer, [53](#)
  - TriggerSyncPoint, [52](#)
  - Update, [52](#)
- AssetPackage::RageBMLRealizerSettings
  - RageBMLRealizerSettings, [54](#)
- au
  - BMLRealizer::BMLFaceFacs, [20](#)
- BMLBehavior
  - BMLRealizer::BMLBehavior, [10](#)
- BMLBlock
  - BMLRealizer::BMLBlock, [12](#)
- BMLBml
  - BMLRealizer::BMLBml, [16](#)
- BMLFace
  - BMLRealizer::BMLFace, [17](#)
- BMLFaceFacs
  - BMLRealizer::BMLFaceFacs, [19](#)
- BMLFaceLexeme
  - BMLRealizer::BMLFaceLexeme, [21](#)
- BMLGaze
  - BMLRealizer::BMLGaze, [24](#)
- BMLGazeShift
  - BMLRealizer::BMLGazeShift, [26](#)
- BMLGesture
  - BMLRealizer::BMLGesture, [28](#)
- BMLHead
  - BMLRealizer::BMLHead, [31](#)
- BMLHeadDirectionShift
  - BMLRealizer::BMLHeadDirectionShift, [33](#)
- BMLLocomotion
  - BMLRealizer::BMLLocomotion, [34](#)
- BMLPointing
  - BMLRealizer::BMLPointing, [36](#)
- BMLPose
  - BMLRealizer::BMLPose, [39](#)
- BMLPosture
  - BMLRealizer::BMLPosture, [41](#)
- BMLPostureShift
  - BMLRealizer::BMLPostureShift, [42](#)
- BMLRealizer, [7](#)
- BMLRealizer.BMLBehavior, [9](#)
- BMLRealizer.BMLBlock, [11](#)
- BMLRealizer.BMLBml, [15](#)
- BMLRealizer.BMLFace, [17](#)
- BMLRealizer.BMLFaceFacs, [18](#)
- BMLRealizer.BMLFaceLexeme, [20](#)
- BMLRealizer.BMLFaceShift, [22](#)
- BMLRealizer.BMLFeedback, [23](#)
- BMLRealizer.BMLGaze, [23](#)
- BMLRealizer.BMLGazeShift, [26](#)
- BMLRealizer.BMLGesture, [27](#)
- BMLRealizer.BMLHead, [29](#)
- BMLRealizer.BMLHeadDirectionShift, [32](#)
- BMLRealizer.BMLLocomotion, [34](#)
- BMLRealizer.BMLPointing, [35](#)
- BMLRealizer.BMLPose, [37](#)
- BMLRealizer.BMLPosture, [40](#)
- BMLRealizer.BMLPostureShift, [41](#)
- BMLRealizer.BMLSpeech, [43](#)
- BMLRealizer.BMLStance, [44](#)
- BMLRealizer.BMLSyncPoint, [46](#)
- BMLRealizer.BMLWait, [49](#)
- BMLRealizer::BMLBehavior
  - BMLBehavior, [10](#)
  - Parse, [10](#)
- BMLRealizer::BMLBlock
  - BMLBlock, [12](#)
  - id, [14](#)
  - parentBml, [14](#)
  - Parse, [12](#)
  - syncPoints, [14](#)
  - TryParseAttribute< T >, [12](#)
  - TryParseSyncPoint, [13](#)
- BMLRealizer::BMLBml
  - BMLBml, [16](#)
  - characterId, [16](#)
  - Composition, [15](#)
  - composition, [16](#)
  - Parse, [16](#)
- BMLRealizer::BMLFace
  - amount, [18](#)
  - BMLFace, [17](#)
  - overshoot, [18](#)
  - Parse, [17](#)
- BMLRealizer::BMLFaceFacs
  - au, [20](#)

- BMLFaceFacs, [19](#)
- Side, [19](#)
- side, [20](#)
- BMLRealizer::BMLFaceLexeme
  - BMLFaceLexeme, [21](#)
  - Lexeme, [21](#)
  - lexeme, [22](#)
  - Parse, [21](#)
- BMLRealizer::BMLGaze
  - BMLGaze, [24](#)
  - Direction, [24](#)
  - Influence, [24](#)
  - influence, [25](#)
  - offsetAngle, [25](#)
  - offsetDirection, [25](#)
  - Parse, [24](#)
  - target, [25](#)
- BMLRealizer::BMLGazeShift
  - BMLGazeShift, [26](#)
  - Parse, [26](#)
- BMLRealizer::BMLGesture
  - BMLGesture, [28](#)
  - Lexeme, [28](#)
  - lexeme, [29](#)
  - Mode, [28](#)
  - mode, [29](#)
  - Parse, [28](#)
- BMLRealizer::BMLHead
  - amount, [31](#)
  - BMLHead, [31](#)
  - Lexeme, [30](#)
  - lexeme, [31](#)
  - Parse, [31](#)
  - repetition, [31](#)
- BMLRealizer::BMLHeadDirectionShift
  - BMLHeadDirectionShift, [33](#)
  - Parse, [33](#)
  - target, [33](#)
- BMLRealizer::BMLLocomotion
  - BMLLocomotion, [34](#)
  - manner, [35](#)
  - Parse, [34](#)
  - target, [35](#)
- BMLRealizer::BMLPointing
  - BMLPointing, [36](#)
  - Mode, [36](#)
  - mode, [37](#)
  - Parse, [36](#)
  - target, [37](#)
- BMLRealizer::BMLPose
  - BMLPose, [39](#)
  - Lexeme, [38](#)
  - lexeme, [39](#)
  - Parse, [39](#)
  - Part, [38](#)
  - part, [39](#)
- BMLRealizer::BMLPosture
  - BMLPosture, [41](#)
  - Parse, [41](#)
- BMLRealizer::BMLPostureShift
  - BMLPostureShift, [42](#)
  - Parse, [42](#)
- BMLRealizer::BMLSpeech
  - BMLSpeech, [43](#)
  - Parse, [44](#)
  - text, [44](#)
- BMLRealizer::BMLStance
  - BMLStance, [45](#)
  - Parse, [45](#)
  - Type, [45](#)
  - type, [46](#)
- BMLRealizer::BMLSyncPoint
  - BMLSyncPoint, [47](#)
  - IsCompleted, [47](#)
  - IsTimerSafe, [47](#)
  - TriggerSyncPoint, [48](#)
  - Update, [48](#)
- BMLRealizer::BMLWait
  - BMLWait, [49](#)
  - duration, [50](#)
  - Parse, [49](#)
- BMLSpeech
  - BMLRealizer::BMLSpeech, [43](#)
- BMLStance
  - BMLRealizer::BMLStance, [45](#)
- BMLSyncPoint
  - BMLRealizer::BMLSyncPoint, [47](#)
- BMLWait
  - BMLRealizer::BMLWait, [49](#)
- characterId
  - BMLRealizer::BMLBml, [16](#)
- Composition
  - BMLRealizer::BMLBml, [15](#)
- composition
  - BMLRealizer::BMLBml, [16](#)
- Direction
  - BMLRealizer::BMLGaze, [24](#)
- duration
  - BMLRealizer::BMLWait, [50](#)
- GetBehaviorFromId
  - AssetPackage::RageBMLRealizer, [51](#)
- id
  - BMLRealizer::BMLBlock, [14](#)
- Influence
  - BMLRealizer::BMLGaze, [24](#)
- influence
  - BMLRealizer::BMLGaze, [25](#)
- IsCompleted
  - BMLRealizer::BMLSyncPoint, [47](#)
- IsTimerSafe
  - BMLRealizer::BMLSyncPoint, [47](#)
- Lexeme

- BMLRealizer::BMLFaceLexeme, [21](#)
- BMLRealizer::BMLGesture, [28](#)
- BMLRealizer::BMLHead, [30](#)
- BMLRealizer::BMLPose, [38](#)
- lexeme
  - BMLRealizer::BMLFaceLexeme, [22](#)
  - BMLRealizer::BMLGesture, [29](#)
  - BMLRealizer::BMLHead, [31](#)
  - BMLRealizer::BMLPose, [39](#)
- manner
  - BMLRealizer::BMLLocomotion, [35](#)
- Mode
  - BMLRealizer::BMLGesture, [28](#)
  - BMLRealizer::BMLPointing, [36](#)
- mode
  - BMLRealizer::BMLGesture, [29](#)
  - BMLRealizer::BMLPointing, [37](#)
- offsetAngle
  - BMLRealizer::BMLGaze, [25](#)
- offsetDirection
  - BMLRealizer::BMLGaze, [25](#)
- overshoot
  - BMLRealizer::BMLFace, [18](#)
- parentBml
  - BMLRealizer::BMLBlock, [14](#)
- Parse
  - BMLRealizer::BMLBehavior, [10](#)
  - BMLRealizer::BMLBlock, [12](#)
  - BMLRealizer::BMLBml, [16](#)
  - BMLRealizer::BMLFace, [17](#)
  - BMLRealizer::BMLFaceLexeme, [21](#)
  - BMLRealizer::BMLGaze, [24](#)
  - BMLRealizer::BMLGazeShift, [26](#)
  - BMLRealizer::BMLGesture, [28](#)
  - BMLRealizer::BMLHead, [31](#)
  - BMLRealizer::BMLHeadDirectionShift, [33](#)
  - BMLRealizer::BMLLocomotion, [34](#)
  - BMLRealizer::BMLPointing, [36](#)
  - BMLRealizer::BMLPose, [39](#)
  - BMLRealizer::BMLPosture, [41](#)
  - BMLRealizer::BMLPostureShift, [42](#)
  - BMLRealizer::BMLSpeech, [44](#)
  - BMLRealizer::BMLStance, [45](#)
  - BMLRealizer::BMLWait, [49](#)
- Part
  - BMLRealizer::BMLPose, [38](#)
- part
  - BMLRealizer::BMLPose, [39](#)
- RageBMLRealizer
  - AssetPackage::RageBMLRealizer, [51](#)
- RageBMLRealizerSettings
  - AssetPackage::RageBMLRealizerSettings, [54](#)
- repetition
  - BMLRealizer::BMLHead, [31](#)
- ScheduledBlocks
  - AssetPackage::RageBMLRealizer, [53](#)
- Settings
  - AssetPackage::RageBMLRealizer, [53](#)
- Side
  - BMLRealizer::BMLFaceFacs, [19](#)
- side
  - BMLRealizer::BMLFaceFacs, [20](#)
- SyncPointCompleted
  - AssetPackage::RageBMLRealizer, [52](#)
- syncPoints
  - BMLRealizer::BMLBlock, [14](#)
- target
  - BMLRealizer::BMLGaze, [25](#)
  - BMLRealizer::BMLHeadDirectionShift, [33](#)
  - BMLRealizer::BMLLocomotion, [35](#)
  - BMLRealizer::BMLPointing, [37](#)
- text
  - BMLRealizer::BMLSpeech, [44](#)
- Timer
  - AssetPackage::RageBMLRealizer, [53](#)
- TriggerSyncPoint
  - AssetPackage::RageBMLRealizer, [52](#)
  - BMLRealizer::BMLSyncPoint, [48](#)
- TryParseAttribute< T >
  - BMLRealizer::BMLBlock, [12](#)
- TryParseSyncPoint
  - BMLRealizer::BMLBlock, [13](#)
- Type
  - BMLRealizer::BMLStance, [45](#)
- type
  - BMLRealizer::BMLStance, [46](#)
- Update
  - AssetPackage::RageBMLRealizer, [52](#)
  - BMLRealizer::BMLSyncPoint, [48](#)