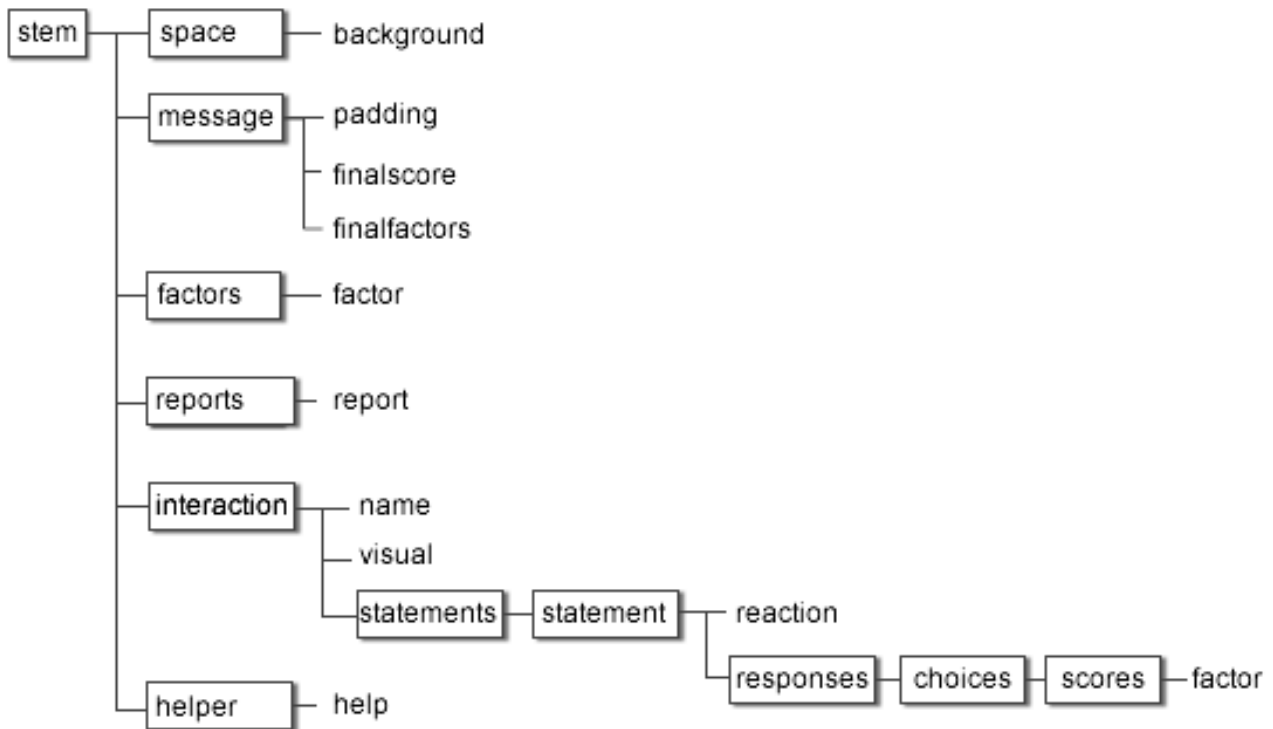


## Stem Reference - Version 0.7

This section contains a reference for all STEM elements, in the order that they appear in the stem.xml document. The class tree for the elements is shown below.

**Note:** Click an element name in this diagram to jump to its entry in the reference section.



Because the files are in XML grammar and file format, tag names are case-sensitive and must appear exactly as shown here:

### <stem>

#### Syntax

```
<stem>  
</stem>
```

#### Description

This is the root element; all other elements must be contained inside it.

### <space>

#### Syntax

```
<space>  
  <background src="media/foyer.swf" x="0" y="0"> </  
background>  
</space>
```

## Description

The **space** element specifies where the interaction occurs. In a future version, this element might be used to define rooms and other spaces.

## Children

`<background>`

## `<background>`

### Syntax

```
<space>  
  <background src="media/foyer.swf" x="0" y="0"> </  
background>  
</space>
```

## Description

The **background** element specifies the background image or media for the interaction. Often, this is used to place an image of a room or other space in the background of an interaction.

## Attributes

- **src** Is the relative path for the background. This path is relative to the stem.swf file, not the stem.xml file in the data folder. The file can be in any image format that Flash supports, e.g. .swf, .jpg, .png. Because .swf files can be used, animation, sound and other interactions can be added to the background.

- 

### Attributes

- **x** The x position of the background. This is normally set to 0.
- **y** The y position of the background. This is normally set to 0.

## Parent

`<space>`

## `<messages>`

### Syntax

```
<messages>  
  <padding x="10"></padding>  
  <finalscore x="420" y="50" displayduring="yes">Score</  
finalscore>  
  <finalfactors display="no"></finalfactors>  
</messages>
```

## Description

The **message** element contains the other elements that control the scoring and feedback systems of STEM.

## Children

```
<padding>
<finalscore>
<finalfactors>
```

## <padding>

### Syntax

```
<messages>
  <padding x="10"></padding>
  <finalscore x="420" y="50" displayduring="yes">Score</
finalscore>
  <finalfactors display="no"></finalfactors>
</messages>
```

### Description

The **padding** element controls how much visual space there is between each of the parts in final the feedback display, e.g. the bar charts show the scores. These elements are stacked vertically, and the padding element only controls the padding between the vertical elements.

### Attributes

- **x** The amount of vertical padding between parts in the message box. This is normally set to 10.

- 

### Parent

```
<messages>
```

## <finalscore>

### Syntax

```
<messages>
  <padding x="10"></padding>
  <finalscore x="420" y="50" displayduring="yes">Score</
finalscore>
  <finalfactors display="no"></finalfactors>
</messages>
```

### Description

The **finalscore** element controls the visual position of the score message, when the messages are shown and what the score is labelled.

### Example

```
<finalscore x="420" y="50" displayduring="yes">Progress</
finalscore>
```

In this example, the score is shown in the top right hand corner with the word progress showing during the whole interaction.

### Example

```
<finalscore x="420" y="50" displayduring="no">Score</finalscore>
```

The score is shown in the top right hand corner only when the end branch is chosen by the user and the word **Score** is used in the label.

### Attributes

- **x** The x position of the score.
- **y** The y position of the score.
- **displayduring**. This can be set to **yes** or **no**. If you only want the score to show at the end, it should be set to **displayduring=no**.

### Parent

```
<messages>
```

## <finalfactors>

### Syntax

```
<messages>  
  <padding x="10"></padding>  
  <finalscore x="420" y="50" displayduring="yes">Score</  
finalscore>  
  <finalfactors display="no"></finalfactors>  
</messages>
```

### Description

The **finalfactors** element controls if feedback is given on how the user's choices performed against each of the factors that are defined. STEM will create simple bar graphs for each of these factors, at the end of the interaction. There is more about the factors feature of STEM below.

### Attributes

- **display** This can be set to **yes** or **no**. If you only want the score to show at the end, it should be set to **displayduring=no**.

### Parent

```
<messages>
```

## <factors>

### Syntax

```
<factors>  
  <factor>Customer Service</factor>  
  <factor>Technical</factor>
```

`</factors>`

## Description

The **factors** element contains other elements that define how the scoring system works. Each choice a user makes has a score for each of the factors. This means that they might be able to make a choice that has a high score for customer service but is not technically correct. In that case, the score for the customer service factor would go up and the technical factor score would go down. The overall score is a combination of all the scores from all the factors. The report shown at the end shows how the user got on against factors. Every choice needs to be rated for all the factors.

There are no limits to the amount of factors, but 3 seems to be a good amount to work with.

## Children

`<factor>`

## `<factor>`

### Syntax

```
<factor>Technical</factor>
```

## Description

The **factor** element simply contains the name of each of scoring factor. For more about factors see 'factors' above.

## Parent

`<factors>`

## `<reports>`

### Syntax

```
<reports>  
  <report></report>  
</reports>
```

## Description

The **reports** element contains the report elements that define how the score is reported at the end.

## Children

`<report>`

## `<report>`

### Syntax

```
<reports>
```

```
<report>scorm</report>
</reports>
```

## Description

The **report** specifies how the score is reported. The default value is blank, and the score is only shown to the user.

## Possible values

**scorm** - When this is set, the following is reported to the Learning Management System via SCORM:

- The total final score.
- The score for each of the factors.
- The choices the users made.

For the SCORM reporting to work, the SCORM\_API\_wrapper.js needs to be included in the html. There is an example of how these dependencies work together in main.scorm.html.

This hasn't been extensively tested in multiple Learning Management Systems. The reports element is also at the end of the setup section of the xml file.

## Children

```
<reports>
```

## <interaction>

### Syntax

```
<interaction>
</interaction>
```

## Description

The **interaction** element is a container element for the main elements that define the branches and choices. At the moment, STEM is set up for only one interaction element per XML file; in the future it may support multiple interactive elements. The first two children elements **name** and **visual** are setup elements.

## Children

```
<name>
<visual>
<statements>
```

## <name>

### Syntax

```
<interaction>
  <name type ="image" ></name>
```

```
<visual x="100" y="50">media/alison.swf</visual>
```

## Description

The **name** element is only used to define what type of media the interaction uses. The actual value of the name is not used at the moment.

## Attributes

### • type

- The possible values for **type** are:
  - **image**: This is the base for all the types of interaction. The file can be any image type that flash supports including .swf. If the image is an .swf, it will play at the end before the responses are shown. An example of how the image type is the based type is the sound type. When the sound element is defined, STEM still uses the base image that is defined in the visual element (see below). STEM will look for an image attribute in each reaction. If an image is not found with the reaction element, the image that is shown to the user will not be changed.
  - **sound**: Means that a sound file such as .mp3 can be defined for each of the reactions. An image still needs be defined in the visual element.
  - **video**: Means that a video file can be defined for each of the reactions. The video is played and then the choices are shown. There are no playback controls for video.

## <visual>

### Syntax

```
<interaction>  
  <name type ="image" ></name>  
  <visual x="100" y="50">media/alison.swf</visual>  
  <statements>
```

## Description

The **visual** element is used to define what is shown when the interaction starts. The user clicks on this image, and then the statements start. This can be used to build introduction screens. If later responses don't define new images, then the image that has been set here stays for the whole interaction.

The path to the file is relative to the stem.swf file. It is suggested that you keep all you images, sounds and video in the media folder.

## Attributes

- **x**: Sets the x position of the image.
- **y**: Sets the y position of the image.

•The height and width are taken from the source file. STEM does not resize images.

## •<statements>

### Syntax

```
<statements>
  <statement id="0" name="intro" x="200" y="80" >
    <reaction >We are building a new house, a bit out of town
and we need to know the cost of getting the power on</reaction>
    <responses x="0" y="290" width="600" height="140"
choiceheight="20">
      <choice branch="1"><statement>Do you have pole
out the front</statement>
        <scores>
          <factor name="Supplier Requirements"
score="1"></factor>
        </scores>
      </choice>
    </responses>
  </statement>
</statements>
```

### Description

The **statements** element is a container for each of the statements/branches. The order and id number statements need to be kept consistent, e.g. you can't have a statement with an id=0 at the end of the list of statements and it expect to work. The numbering system starts from 0.

The metaphor that STEM uses for its interactions is that there is a series of statements with reactions and responses. The idea is that STEM has a "reaction", and then the user has the choice to "respond".

### Children

```
<statement>
```

## <statement>

### Syntax

```
<statements>
  <statement id="0" name="intro" help="0" x="200" y="80" >
    <reaction >We are building a new house, a bit out of town
and we need to know the cost of getting the power on</reaction>
    <responses x="0" y="290" width="600" height="140"
choiceheight="20">
      <choice branch="1"><statement>Do you have pole
out the front</statement>
        <scores>
```

```

        <factor name="Supplier Requirements"
score="1"></factor>
        </scores>
    </choice>
</responses>
</statement>
</statements>

```

## Description

The **statement** element is the container element for each stage of a branch.

## Attributes

- **id**: Each statement needs to be given an id number. These id numbers are used to make the link between which choice is made and what statement happens next. The id numbers **need to start from 0**. The order and id number statements need to be keep consistent, e.g. you can't have a statement with an id=0 at the end of the list of statements and expect it work.
- **name**: This is not used in STEM at this stage, but it giving each statement a name will make the file easier to read.
- **help**: Is optional and in most case isn't used. A setting such as `help="0"` means that the help element at the position 0 of the helper element (which is later in the XML file) will be shown. This is often used when the user has made bad choice.
- **x**: Sets the x position of the statement.
- **y**: Sets the y position of the statement.

## Children

```

<reaction>
<responses>

```

## <reaction>

### Syntax

```

<statements>
    <statement id="0" name="intro" x="200" y="80" >
        <reaction >We are building a new house, a bit out of town
and we need to know the cost of getting the power on</reaction>
        <responses x="0" y="290" width="600" height="140"
choiceheight="20">
            <choice branch="1"><statement>Do you have pole
out the front</statement>
                <scores>
                    <factor name="Supplier Requirements"
score="1"></factor>
                </scores>
            </choice>
        </responses>
    </statement>

```

```
</statements>
```

## Description

The **reaction** element controls what is shown at each of the branches.

The value reaction element placed in a text box contained in another image. The image is a ReactBackground.swf file, and this can be modified by changing the ReactBackground.fla file. The default is a speech bubble. Different media types can be used as attributes if that type is set in the **name** element.

HTML text can be used as a value here as well if it's contained in **<![CDATA** `[ "<HTML text goes here>" ] ]>`

The x and y position is taken from the statement element.

## Example

```
<reaction ><![CDATA["<strong>Don't speak to me like that</strong>"]]></reaction>
```

**Attributes - all are optional and depend on the type that has been set in the name.**

- **image:** This is the relative path to an image file. The file type can be any image format that Flash supports. Not every statement needs an image; if an image is not defined for a statement then the last image used will remain.

## Example

```
<reaction image="media/jill.jpg" >Hi </reaction>
```

In this example the image file jill.jpg is loaded and shown, and the text "Hi" is shown.

```
<reaction image="media/jill_sayinghi.swf"> </reaction>
```

In this example the image file jill\_sayinghi.swf is loaded and shown, but no text is shown. If the .swf has a timeline, the timeline is played before the responses are shown, e.g. .swf can be an animation lip-synced to audio.

- **video:** This is the relative path to a video file. The file type can be any video format that Flash supports.

## Example

```
<reaction video="media/DockingSaw_OP_8_Never_walk_away_while_machine_is_running.f4v" ></reaction>
```

In this example, the video file DockingSaw\_OP\_8\_Never\_walk\_away\_while\_machine\_is\_running.f4v is played before the responses are shown. There is no text shown. If the position or size of the video moves between statements, you may see a white flash.

- **sound:** This is the relative path to a sound file. The file type can be any sound format that Flash supports.

### Example

```
<reaction sound="media/audio/1went_into_work_job_after40_years.mp3"></reaction>
```

In this example, the sound file 1went\_into\_work\_job\_after40\_years.mp3 is played before the responses are shown. There is no text shown.

## <responses>

### Syntax

```
<responses x="0" y="290" width="600" height="140" choiceheight="20">
  <choice branch="1"><statement>Look here, don't talk to me like that </statement>
    <scores>
      <factor name="Customers Services" score="-5"></factor>
      <factor name="Sales" score="-5"></factor>
    </scores>
  </choice>
</responses>
```

### Description

The **response** element is a container element for the list of choices the user is given at each stage. The choices are listed in a Flash List component. The attributes for the response element set the visual position, and the height and width of that list. The look and feel of this can be changed in the .fla in same way any Flash component can be styled.

### Attributes

- **x:** Sets the x position of the responses.
- **y:** Sets the y position of the responses.
- **width:** Sets the width of the responses.
- **height:** Sets the height of all the responses.
- **choiceheight:** Sets the height of the actual choices. This is useful when the choices are long and wrap across multiple lines.

### Children

```
<choice>
```

## <choice>

### Syntax

```
<responses x="0" y="290" width="600" height="140"
choiceheight="20">
  <choice branch="1"><statement>Look here, don't talk to me
like that </statement>
  <scores>
    <factor name="Customers Services" score="-5"></factor>
    <factor name="Sales" score="-5"></factor>
  </scores>
</choice>
</responses>
```

### Description

The **choice** element is the key element that shows what the choices are and where they go. The branch attribute controls which statement is shown next.

The statement element controls what is shown in the list of choices.

The score element controls how the choice is scored.

### Attributes

- **branch:** This is the number of the next statement. The actual positions are in the list of statements, not the id number you have defined. The other possible value is **end**. The end value means that the final score will be shown with the messages, and the interaction will be over.

### Children

```
<scores>
<statement>
```

## <statement>

### Syntax

```
<responses x="0" y="290" width="600" height="140"
choiceheight="20">
  <choice branch="1"><statement>Look here, don't talk to me
like that </statement>
  <scores>
    <factor name="Customers Services" score="-5"></factor>
    <factor name="Sales" score="-5"></factor>
  </scores>
</choice>
</responses>
```

## Description

The **statement** element is a container element for what appears in the list of choices. The height of the element is defined in the choiceheight attribute of the responses element. The text inside this element can be HTML text and can be images. This is a simple way to make a list of image choices.

## Example

```
<statement ><![CDATA[""]]></statement>
```

## <scores>

### Syntax

```
<responses x="0" y="290" width="600" height="140"
choiceheight="20">
  <choice branch="1"><statement>Look here, don't talk to me
like that </statement>
  <scores>
    <factor name="Customers Services" score="-5"></factor>
    <factor name="Sales" score="-5"></factor>
  </scores>
</choice>
</responses>
```

## Description

Each choice needs to have a **<scores>** element that holds the scores for each of the factor elements.

## Attributes

## Children

```
<factor>
```

## <factor>

### Syntax

```
<responses x="0" y="290" width="600" height="140"
choiceheight="20">
  <choice branch="1"><statement>Look here, don't talk to me
like that </statement>
  <scores>
    <factor name="Customers Services" score="-5"></factor>
    <factor name="Sales" score="-5"></factor>
  </scores>
</choice>
</responses>
```

## Description

For each choice, you need to define what the score is for each of the factors. These are the factors that were defined at the start of the document in the factors element.

## Attributes

- **name:** This is the exact name of the factors.
- **score:** Each of the factors needs to have a score; this can be a positive or negative number.

## <helper>

### Syntax

<helper>

```
<help id="0" type="text" x="100" y="100" width="300"
height="20" background="0x000000" alpha=".8" color="0xFFFFFFFF">You
shouldn't be giving advice like that out without checking without
first. </help>
<help id="1" x="200" y="200" width="100" height="110"
background="0x000000" alpha=".8" color="0xFFFFFFFF"> That is not a
suitable response</help>
```

</helper>

## Description

The **helper** element is a container element that appears at the end of the document. It contains a series of help elements that can be shown during the interaction.

## Children

<help>

## <help>

### Syntax

<helper>

```
<help id="0" type="text" x="100" y="100" width="300"
height="20" background="0x000000" alpha=".8" color="0xFFFFFFFF">You
shouldn't be giving advice like that out without checking without
first. </help>
<help id="1" type="text" x="200" y="200" width="100"
height="110" background="0x000000" alpha=".8" color="0xFFFFFFFF">
That is not a suitable response</help>
```

</helper>

## Description

The **help** element contains feedback that can be given during the interaction. The attributes on the element control the appearance of the feedback.

## Attributes

- **id**: Is a number that reflects the help position in the helper element. This is the number that is used in the statement element.
- **type**: Currently the only type that has been implemented is the text type.
- **x**: Sets the x position of the help.
- **y**: Sets the y position of the help.
- **width**: Sets the width of the help.
- **height**: Sets the height of all the help.
- **background**: Sets the background colour of the help element; this is an RGB colour in written in HEX.
- **alpha**: Sets the transparency of the element. Because the help element is often put on top of the other elements, this can useful.
- **colour**: Sets the colour of the text; this is an RGB colour in written in HEX.