

SmileText Component API

© 2006/07 TUFaT.com, All Rights Reserved

To use the SmileText and SmileTextImport components, you must first install them by opening smiletext.mxp and smiletextimport.mxp. You must have the Macromedia Extension Manager, which you can download from www.macromedia.com. After successful installation, both components should be available in the Flash IDE Components Panel.

Why are there TWO components in this package? What's the difference?

'smiletext' – a textarea component with smilies

'smiletextimport' – a component into which text can be typed, and then sent to the 'smiletext' component

The reason that these have been split into two packages is for design flexibility. You may, for example, not wish to have the input box directly below the textarea. By separating the input area from the actual text component, you have the freedom to choose any layout that you wish. The components, for example, can be in different movie clips, or in different locations in the same clip.

IMPORTANT: To show the smiles in the text component (smiletext), you **MUST** have in your Flash library the smile movie clips, or load the smiles from an external SWF file. The example file illustrates this.

NOTE: To begin working with the SmileText and SmileTextImport components, you must call the following methods:

1. For SmileText:
`smiletext_instance.setInputMovie(smiletextimport_instance);`
2. For SmileTextImport:
`smiletextimport_instance.setOutputMovie(smiletext_instance);`
3. Be sure that in your Flash library, you have placed the folder called "SMILIES" with all smile content.

- **SmileText component set text align (left, right):**

Usage

```
smiletext_instance.setRTL(boolean_value);
```

Sample

```
smiletext_instance.setRTL(true);//right alignment  
smiletext_instance.setRTL(false);//left alignment
```

- **SmileText component background color:**

Usage

```
smiletext_instance.setBackgroundColor(COLOR, ALPHA);
```

Sample

```
smiletext_instance.setBackgroundColor(0x00FF22, 10);
```

- **SmileText component frame color:**

Usage

```
smiletext_instance.setBorderColor(COLOR, ALPHA);
```

Sample

```
smiletext_instance.setBorderColor(0x00FF22, 10);
```

- **SmileText component size:**

Usage

```
smiletext_instance.setSize(WIDTH, HEIGHT);
```

Sample

```
smiletext_instance.setSize(200, 200);
```

- **SmileText component all text deletion:**

Usage

```
smiletext_instance.clear();
```

- **SmileText component text format:**

Usage

```
smiletext_instance.setupTextFormat(FONT, SIZE);
```

Sample

```
smiletext_instance.setupTextFormat('Arial', 12);
```

- **SmileText component show/hide smilies:**

Usage

```
smiletext_instance.setShowSmilies(boolean_value);
```

Sample

```
smiletext_instance.setShowSmilies(true);
```

- **Add text in SmileText component. First parameter is heading, second parameter is text (that can include smilies), third parameter is text colour, fourth is user ID (useful in creating multi-user chat applications).**

Usage

```
smiletext_instance.addText(string_value, string_value, COLOR, user_id);
```

Sample

```
smiletext_instance.addText("AVATAR[UserName] 9:58 am: ", "Some text...", 0x000000, 4);
```

```
smiletext_instance.addText("AVATAR[UserName] 9:58 am: ", ":roll: :tennis:", 0x000000, 5);
```

- **Determination of function, to which user's ID is sent, and which brings back a line with the smile. Smiletext component changes AVATAR line on function return result:**

Usage

```
smiletext_instance.setIdAvatarFunction(function_value);
```

Sample

```
var inFunction = function(USER_ID) {
    if(USER_ID == 3)
        return ":baby:";
    else
        if(USER_ID == 5)
            return ":roll:";
        else
            return "";
}
```

```
smiletext_instance.setIdAvatarFunction(inFunction);
```

- **SmileTextImport component set text align (left, right):**

Usage

```
smiletextimport_instance.setRTL(boolean_value);
```

Sample

```
smiletextimport_instance.setRTL(true);//right alignment
smiletextimport_instance.setRTL(false);//left alignment
```

- **Specify label that will be send to smiletext component lines:**

Usage

```
smiletextimport_instance.setLabel(string_value);
```

Sample

```
smiletextimport_instance.setLabel("AVATAR[UserName] 9:58 am: ");
```

- **Setup of user's ID in smiletextimport, that will be send to smiletext component:**

Usage

```
smiletextimport_instance.setUserID(number_value);
```

Sample

```
smiletextimport_instance.setUserID(5);
```

- **Setup smiles selection component (which activates the selection panel or combobox):**

Usage

```
smiletextimport_instance.setSmiliesComponent(string_value);
```

Sample

```
smiletextimport_instance.setSmiliesComponent("Button");
```

```
smiletextimport_instance.setSmiliesComponent("ComboBox");
```

- **Setup background color of smiles selection panel:**

Usage

```
smiletextimport_instance.setSmiAreaBGColor(COLOR);
```

Sample

```
smiletextimport_instance.setSmiAreaBGColor(0xFF0000);
```

- **Setup frame color of smiles selection panel:**

Usage

```
smiletextimport_instance.setBackgroundColor (COLOR);
```

Sample

```
smiletextimport_instance.setBackgroundColor (0xFF0000);
```

- **Setup frame color of smiletextimport component:**

Usage

```
smiletextimport_instance.setBorderColor(COLOR);
```

Sample

```
smiletextimport_instance.setBorderColor(0xFF0000);
```

- **Delete smile:**

Usage

```
smiletext_instance.RemoveSmile(smile_name);
```

Sample

```
smiletext_instance.RemoveSmile(":roll:");
```

- **Add smile:**

Usage

```
smiletext_instance.AddSmile(library_name, smile_text, WIDTH, HEIGHT, XSCALE, YSCALE);
```

Sample

```
smiletext_instance.AddSmile("smi_roll", ":roll:", 20, 15, 50, 50);
```

- **Set call-back function – calls when user send message. First param – movieclip, that contain call-back function, second param – function name:**

Usage

```
smiletextimport_instance.setSendFunction(movie_clip, function_name);
```

Sample

```
smiletextimport_instance.setSendFunction(_root, "call_back_func");
```