



Super Menu (v1) manual

Thank you for purchasing this product at www.3dbeat.com

This manual describes the usage of this component.

Note: At the moment you can just use one instance of this component into the stage.

1- What is the super menu?

This is a flash V3 component cascading XML base menu which has a lot of interactivity. As you can see, at the below picture it contains some text nodes, menu option (in the Gold Edition version), and handles. If you need an interactive menu flash component it is yours!

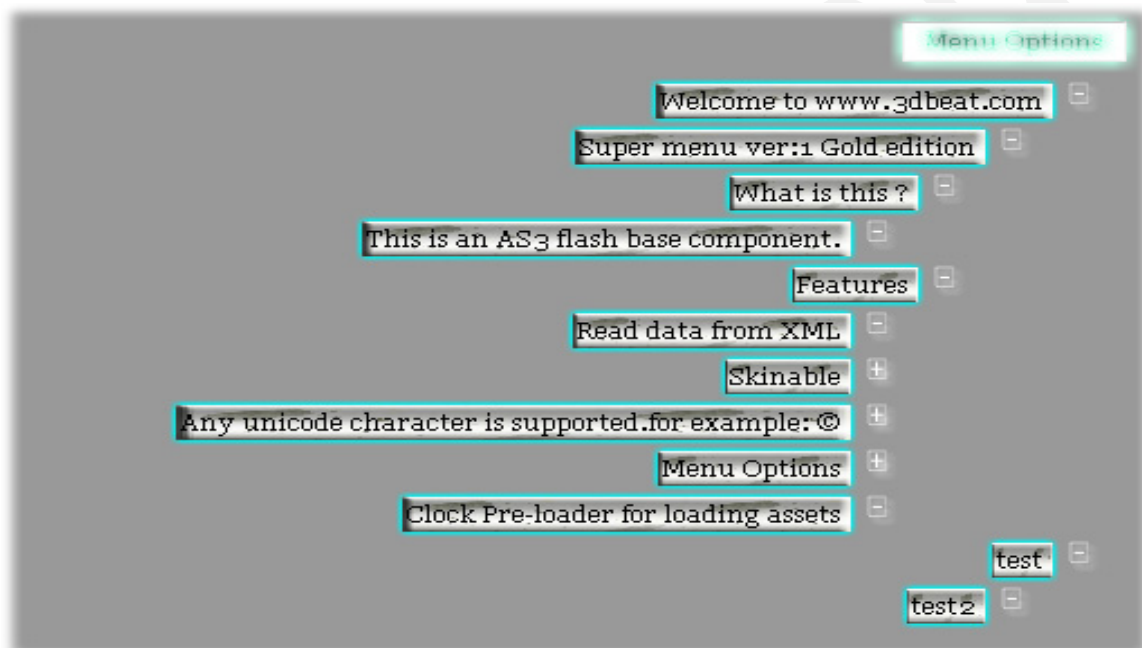


Figure 1: 3dBeat super menu Gold edition

2- What is the difference between Super Menu Gold and Super Menu Silver?

The Gold version has a menu option which can control most of the input properties such as color cascading, sound volumes, speed properties, etc.

The most important feature of the Gold version is Cascading scrollbar. By this you can open or close the children of the text field's roots easily especially when you have a lot of text boxes.

This feature exists in the Sort tab of the menu option.



Super Menu (v1) manual

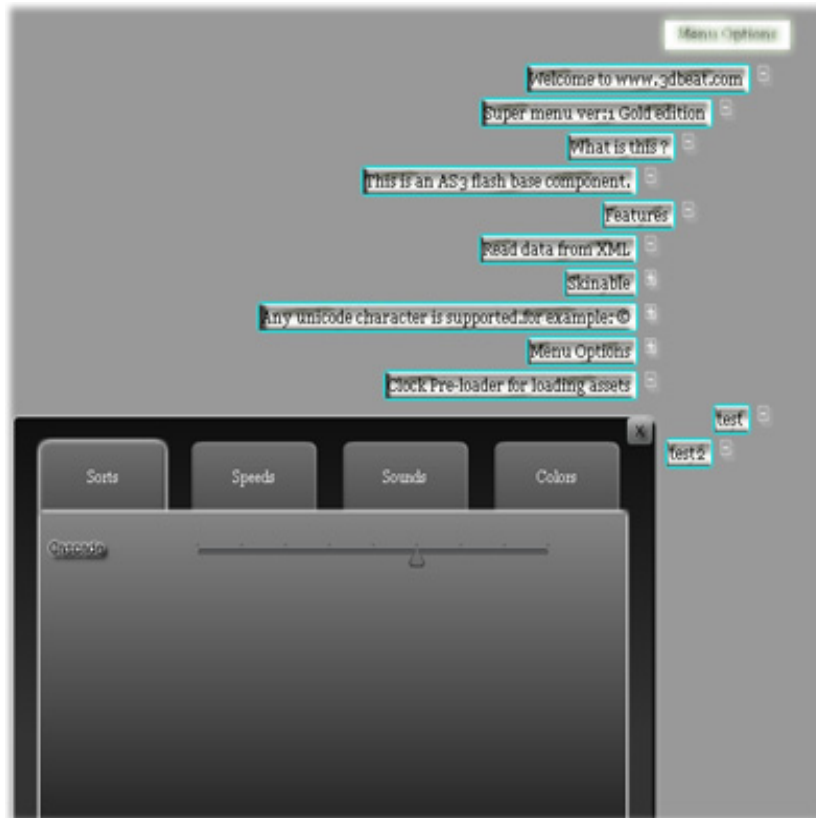


Figure 2: 3dBeat super menu Gold edition

3- What are in this package?

By purchasing this product you will receive:

MXP file:

It can be used by Adobe Flash users to install it through Adobe Extension Manager CS3.

MXP Gold Edition version 1 file size: 146 KB

MXP Silver Edition version 1 file size: 137 KB

By installing the MXP file some other files and folders will be created on your hard drive at BOOT DRIVE:
\\Documents and Settings\\(your username)\\Local Settings\\Application Data\\Adobe\\Flash
CS3\\en\\Configuration\\Components\\3dBeat\\supermenu for windows systems and similarly for Macintosh systems.
You can copy the contents of this folder to wherever you want (except the SWC file which should be presented in
the default directory)

These file are:

1- 3dbeat_xmltree.xml :

It contains the XML data for creating this component. This file is essential for both GOLD and SILVER edition.



Super Menu (v1) manual

2-Image directory:

It contains background.jpg and mySwf.swf (and the source file mySwf fla) that can be used as the background of buttons. (Please look at the end section of this document, if you want to use SWF as the text boxes background)

3- The sound directory which contains two sample sound files as the mp3 format (sound1.mp3, sound2.mp3)

Instructions

1- 3dbeat_xmltree.xml :

This file has two significant information categories for the super menu.

a) Interface tag :

```
E.g. <interface>
      <type alignment="vertical_right"></type>
      <backgroundButton
      ButtonImageName="Images\background.jpg"></backgroundButton>
      <PropertiesOfText TextFont="Georgia" size="12"
      color="0x000000"></PropertiesOfText>
      <position x="710" y="6" distanceBetweenChildButtons="-30"
      distanceBetweenParents="6" opened="true"></position>
      <speed buttonsMovement="3" fade="5"></speed>
      <color button="0xfffff" borderColorTextField="0x00eeee"
      borderColorClosingTextField="0xa428a4" borderColorMouseOver="0x5080FD"
      borderColorMouseClicked="0xccff00"></color>
      <sound buttonTextFiledRollOver="sound\sound2.mp3"
      buttonTextFiledMouseDown="sound\sound1.mp3"></sound>
</interface>
```

This part can be directly edited by the user to change the interface of the super menu. Except the first line (license) you can change the value of corresponding xml data. It is clear you cannot change the tag's name (license, alignment, etc). Also do not replace the positions of attributes. The `type` tag has two variations. The `"vertical_right"` (default) and `"vertical_left"`.

Super Menu (v1) manual

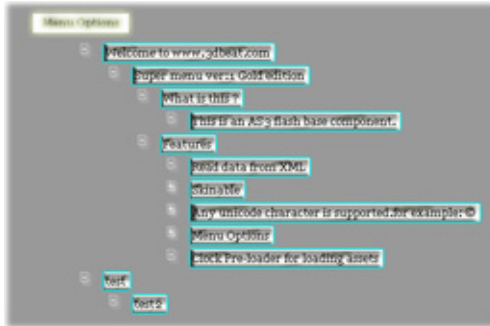


Figure 3: "vertical_left"

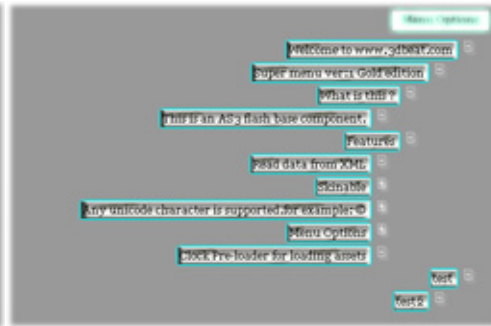


Figure 4: "vertical_right"

ButtonImageName : you can identify the path of the background which can be JPG, PNG or SWF file (Read about the SWF file at the end section).

Note: You can leave empty string ("") if you want to have default background.

e.g. `<backgroundButton ButtonImageName=""></backgroundButton>`

PropertiesOfText: It contains some Text properties. For color attribute you should write the 32 bit HEX or decimal of the color.

"0xFFFFFFFF" or "#FFFFFF" means white "0x000000" or "#000000" is black.

The **positions** tag contains the position of the first text input X and Y coordinate in pixel.(In the Gold edition for the Y position please consider 42 pixels for menu options too)

distanceBetweenChildButtons attribute: It is the horizontal position among text inputs in pixel. If you use positive numbers it will be set -20 automatically in the SWF file.

distanceBetweenParents attribute: It is the vertical distance among text inputs in pixel. If you use any negative number it will be round to +6.

Opened attribute: Identify the first situation of the text inputs. It can be a Boolean value true or false. If you use true the text inputs will open at the fist of loading, else they will close. If you type something wrong instead of Boolean (true/false) it means false.

The Speed tag:

buttonsMovement : Identify the speed of text inputs movement when the handle is pressed in order to close or open the child text inputs.

Fade: The fading speed of text input children when the handle presses.

The color tag:

button: The color of handles

borderColorTextField: the border color of text fields

borderColorClosingTextField: The border color of text fields when they are closing

borderColorMouseOver: The border color of text fields when the mouse is over them

borderColorMouseClicked: The border color of clicked text filed

Note: If you leave something wrong such as the text or leave an empty string the color will be assumed as BLACK color.

The sound tag:



Super Menu (v1) manual

buttonTextFiledRollOver: The playing sound when the mouse is over any text fields.

buttonTextFieldMouseDown: The playing sound when the user clicks on the clicked text field

Note: You can leave empty string ("") if you want to have no sound.

e.g. `<sound buttonTextFiledRollOver="" buttonTextFieldMouseDown ="" ></sound>`

Button Texts:

It contains all of the texts which presented as the text of buttons.

The structure of writing your texts is simple. Just use any appropriate tag (e.g. P) and set the sub menu as the child of your selected tag. Do this procedure to reach your favor number of child text boxes. *Also do not forget to close each tag for each opened tag.*

E.g.

```
<p>This is the first text
```

```
    <p>This is the second text as the child of previous text</p>
```

```
    <p>This is the third text as the child of previous text and as the sibling of second text</p>
```

```
</p>
```

```
<p>This is the end text as the sibling of the first text
```

```
</p>
```

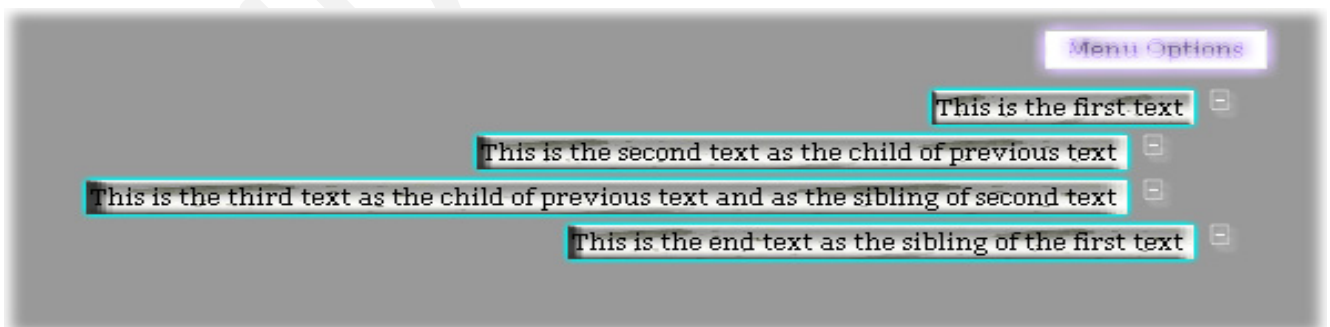


Figure 5: Sample of text nodes

2- Installing:



Super Menu (v1) manual

We have installed this component via Extension manager version 1.8.169. But it seems it can be installed with lower version too.

You can download the last version of Adobe Extension manager for windows systems from:
http://download.macromedia.com/pub/dw_exchange/extension_manager/win/m_install.exe
Or

http://download.macromedia.com/pub/dw_exchange/extension_manager/mac/install.dmg for
Mac systems.

3- Component usage:

After installing the component by Adobe Extension manager, select the 'Components' as you see in the below picture. (Fig 6)

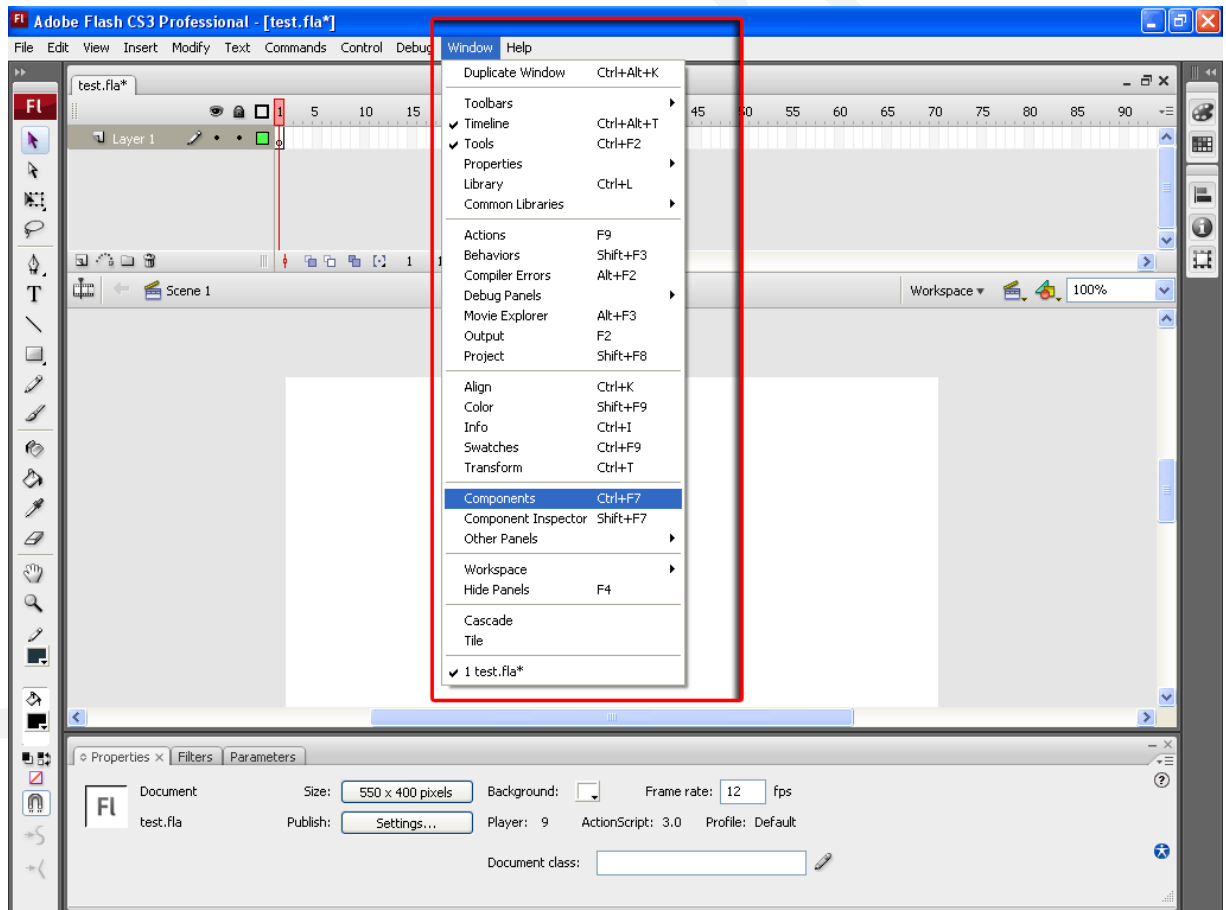


Figure 6: Selecting the Components menu

If your installation is done OK you should see 3dBeat component(s) into your Adobe Flash CS3 Components window. If you have installed this component via Adobe Extension Manager when your



Super Menu (v1) manual

Adobe Flash CS3 is Open, You need to restart your Adobe Flash or choose the Reload button as you see in the below picture in order to refresh the components list in your Adobe Flash Components window. (Fig 7)

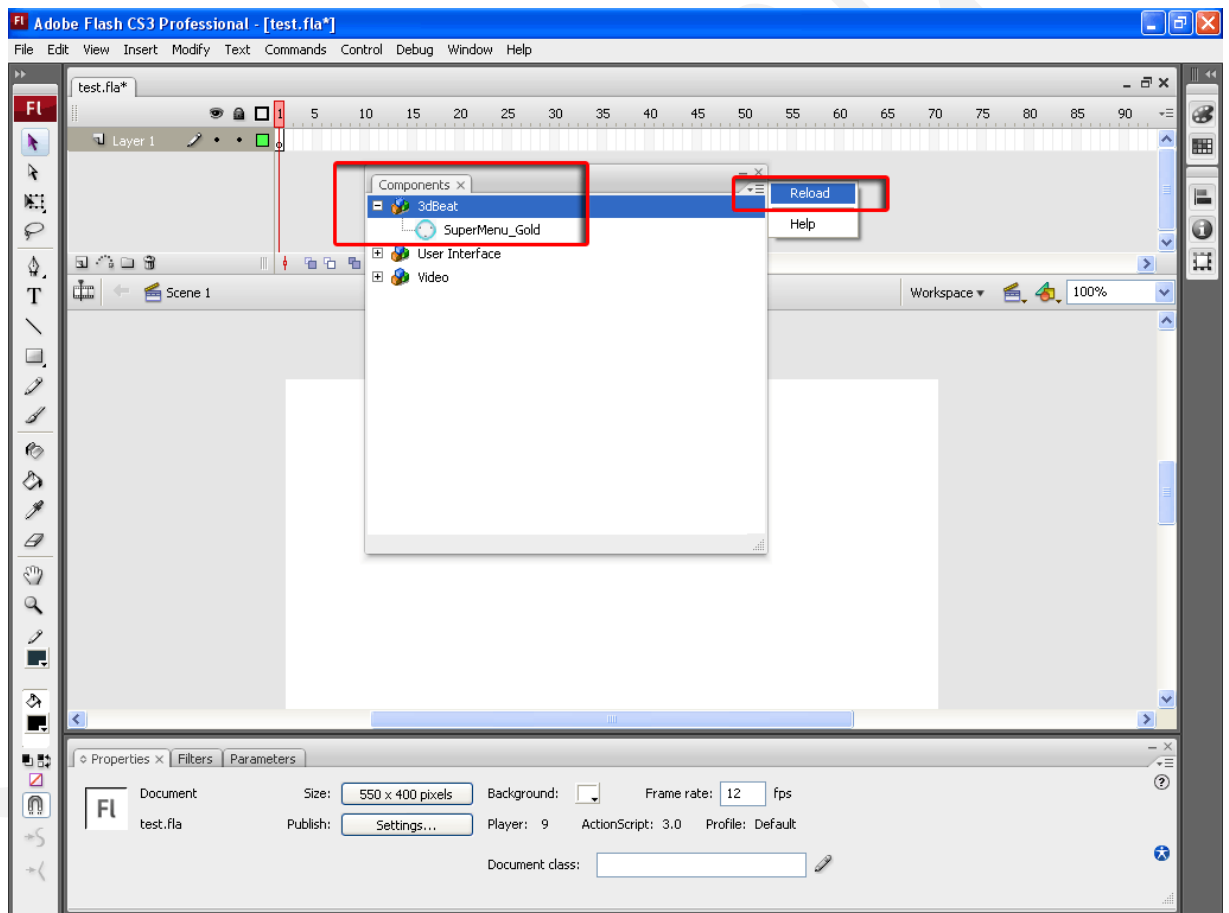


Figure 7: Super Menu Component



Super Menu (v1) manual

Then, you can drag the SuperMenu_Gold or SuperMenu_Silver into the stage. The next step is naming the dragged component. Select the dragged component into the stage and select the properties tab as you see in the below picture and name it by your favor name (e.g. your movie). (Fig 8)

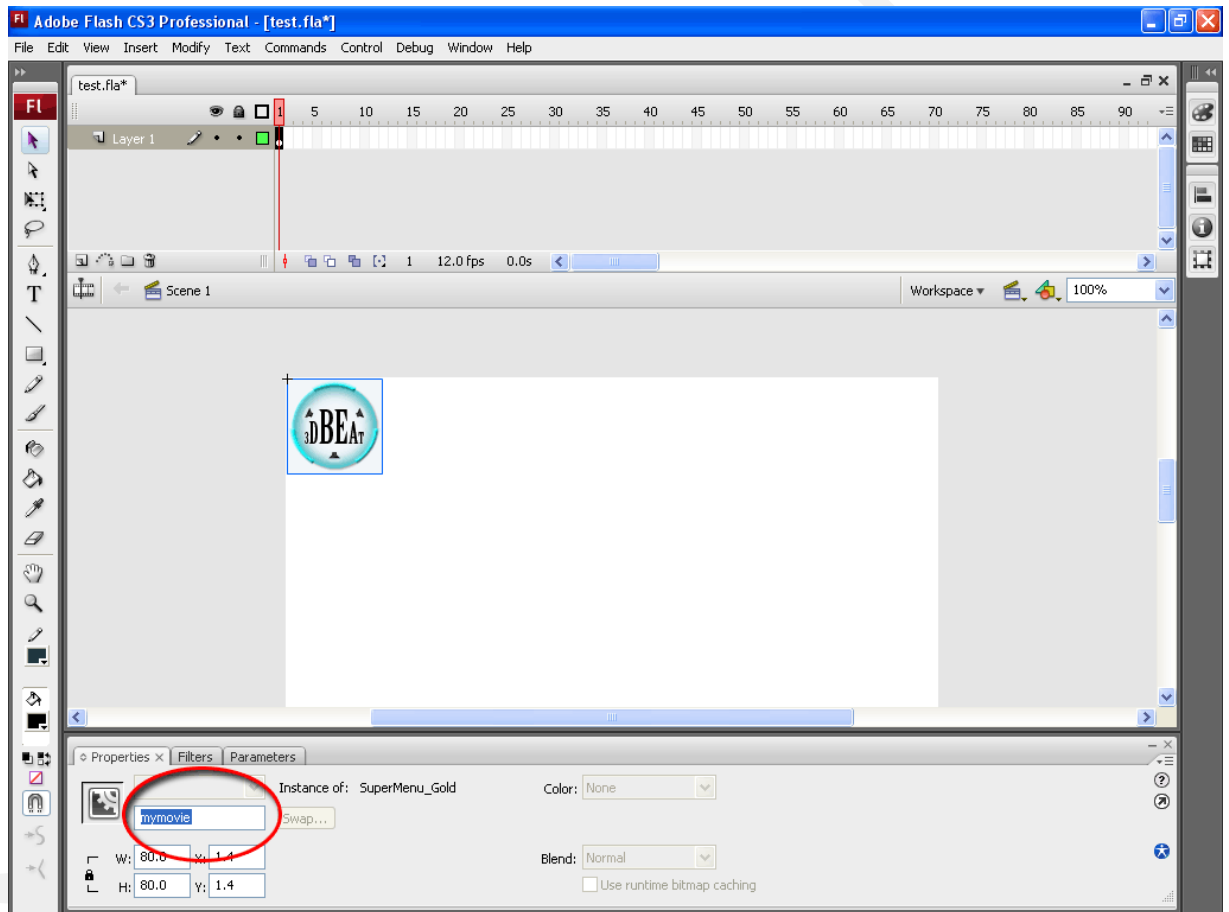


Figure 8: Naming your movie



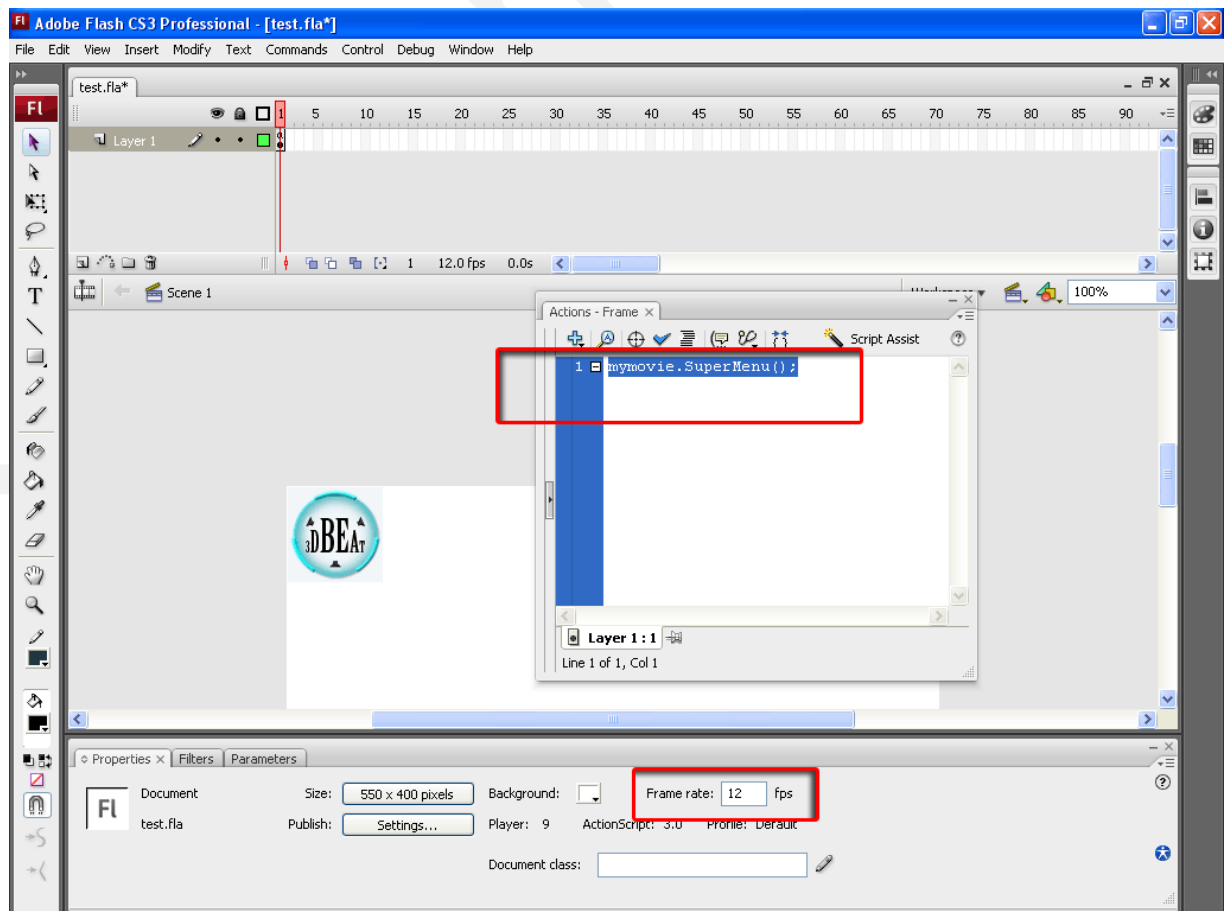
Super Menu (v1) manual

Then select the Actions Panel and type: `mymovie.SuperMenu()`; Also you can increase the frame rate of your movie to make your SWF file more smoothly. (Fig 9)

Note: If you want to put this component to the stage by coding (Not drag and drop) use the following codes:

```
var mymovie:SuperMenu_Gold=new SuperMenu_Gold();  
  
addChild(mymovie); //adding your component as a visible movie  
  
mymovie.SuperMenu(); // initialize the inner parts of the component
```

It is very important to use `addChild` method before `SuperMenu` method. If you use the `SuperMenu` method at first you will encounter an error.



Super Menu (v1) manual



Figure 9: Declare your movie

It is finished. Just set the properties in the parameters panel and run your movie. The description of the properties are same as the description of 3dbeat_xmltree.xml . (Fig 10)

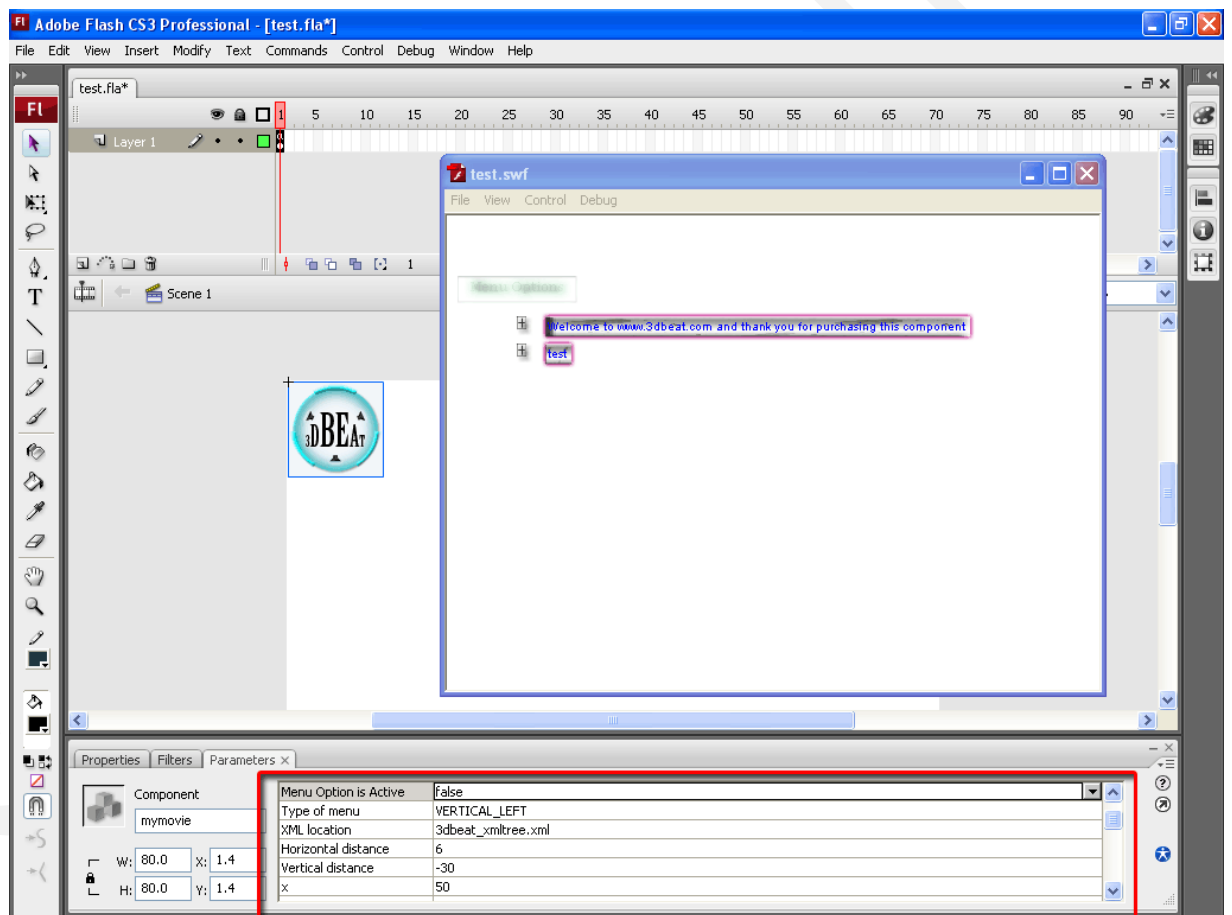


Figure10: Run your movie

All of the XML properties are available in the component inspector. Also the first property which is a Boolean type is "Ignore the XML "

If set to true the property inspector properties override the xml properties.

If set to false the XML properties is override the inspector property.

Super Menu (v1) manual



Note: you cannot remove the XML file in both cases.

Extra Functions:

You can assign any functions for clicked buttons:

At first define your function in any scope that you want by passing one argument as an object. (This argument returns to the text input which is clicked)

```
function myFunction(e:Object){  
    dosomething();  
}
```

Then register it by using registerFunction method of the Super Menu. For argument you can simply use the name of the function that you declared already.

```
Yourmove.registerFunction(myFunction);  
Or  
Yourmove.registerFunction(functionScope.myFunction);
```

To assign your function on your favor text input, there is an "id" property for each text inputs that starts from 1 to N (N = the number of text inputs)

So you can assign your function to a specific text input through below code:

```
function myFunction(e:Object){  
  
    if( e.id==1){  
        dosomething1();  
    }  
    // this line identifies that the selected text button is the first  
    //text button and so you can assign your function on it easily.
```

Super Menu (v1) manual



```
}else if ( e.id==2){
```

```
// this line identifies that the selected text button is the  
second
```

```
Dosomething2(e);
```

```
}
```

```
}
```

For example the codes that is written in the below picture (Fig 10) shows the id of clicked button.

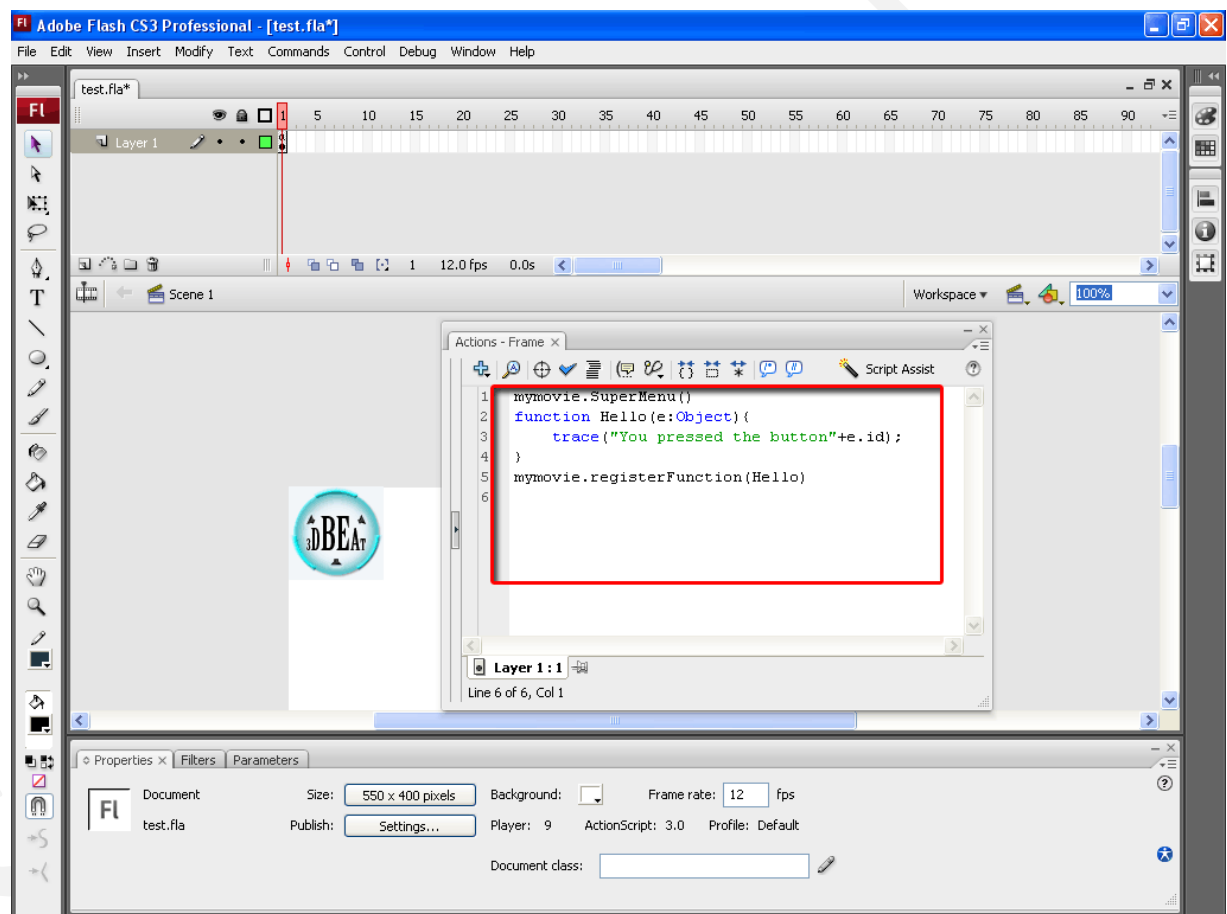


Figure11: Extra function

4- Using SWF file as the background of each text inputs:

According to the sample (images/mySwf.FLA) the flash movie should have a 'BackGroundClip' movie clip. It is essential for your SWF that has this clip name. What you want to happen in the background of text inputs should be placed at this movie clip.



Super Menu (v1) manual

The reason of creating this movie clip returns to a bug issue that we had by document class of SWF files at the time of the component creation.

© 2008 www.3dbeat.com

Super menu