

# colorPicker

User Guide version 1.1  
[www.flashloaded.com](http://www.flashloaded.com)

# Table of Contents

<b>Installation</b>	<b>3</b>
<b>Getting started</b>	<b>4</b>
<b>Component Inspector Parameters</b>	<b>5</b>
<b>Using the colorPicker</b>	<b>6</b>
Basic and advanced mode	6
<b>Advanced mode:</b>	<b>6</b>
Creating a custom user defined color palette	7
<b>Defining the basic palette XML</b>	<b>8</b>
<b>Creating pre-defined custom palettes</b>	<b>10</b>
<b>Skinning the colorPicker</b>	<b>12</b>
<b>ActionScript 2.0 API</b>	<b>13</b>
Events	13
Methods	14
Properties	16
<b>ActionScript 3.0 API</b>	<b>18</b>
Methods	19
Properties	21
<b>Help</b>	<b>23</b>

# Installation

You will need Adobe Extension Manager in order to install this component. Extension Manager should have been installed by default when you installed Flash. You may download the latest version of Extension Manager for free from the [Adobe website](#).

1. Ensure that Flash is closed before installing the colorPicker component.
2. Unzip/extract the colorPicker-AS2.zip or colorPicker-AS3.zip file that you downloaded. You will find a file called colorPicker-AS2.mxp or colorPicker-AS3.mxp. Double click on this file in order to install the component using Extension Manager.

The colorPicker should now be installed in your Flash Components Panel.

# Getting started

1. Having installed the colorPicker using the Adobe Extension Manager, start a new or open an existing Flash file.
2. Locate the colorPicker component in the components panel and drag it onto the stage. The color square represents the selected color.
3. Open the Component Inspector (shift+F7) and set the parameters. Please refer to the [Component Inspector Parameters](#) section for an explanation of each parameter.

## Component Inspector Parameters

Parameter	Description	Example
basicPaletteXML	The path and filename of the XML file where the custom basic color palette is defined. Enter <i>default</i> to use the built-in default color palette.	default
defaultColor	The default selected color that the colorPicker displays.	5
defaultPaletteColor	The default name for the first custom palette.	-180
mode	Sets whether to display the colorPicker initially in basic or advanced mode.	basic
palettesFamily	Optionally enter a word here. Enter the same word in other instances of the colorPicker that you would like to share the same user built custom color palettes.	palettes
palettesXML	The path and filename of the XML file containing a predefined set of color palettes.	colors.xml
skinBgColor	The background color of the colorPicker component.	#FFFFFF
skinBorderColor	The border color around the colorPicker and the elements inside the colorPicker.	#C6C6C6
skinButtonTextColor	The color of the text labels for the buttons in the colorPicker.	#000000
skinTextColor	The color of the general text labels in the colorPicker.	#333333

# Using the colorPicker

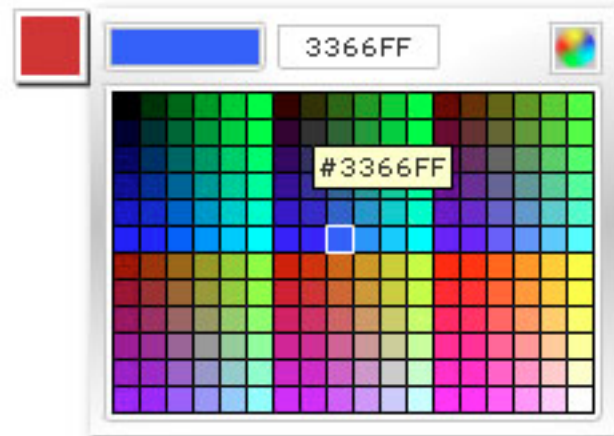
## Basic and advanced mode

The colorPicker can be set to appear in **basic** or **advanced** mode. Setting it in basic mode allows the user to switch to advanced mode by clicking on the small color wheel icon located in the top right corner of the basic palette:

### Basic mode:

In basic mode, the colors can be selected using any of the following methods:

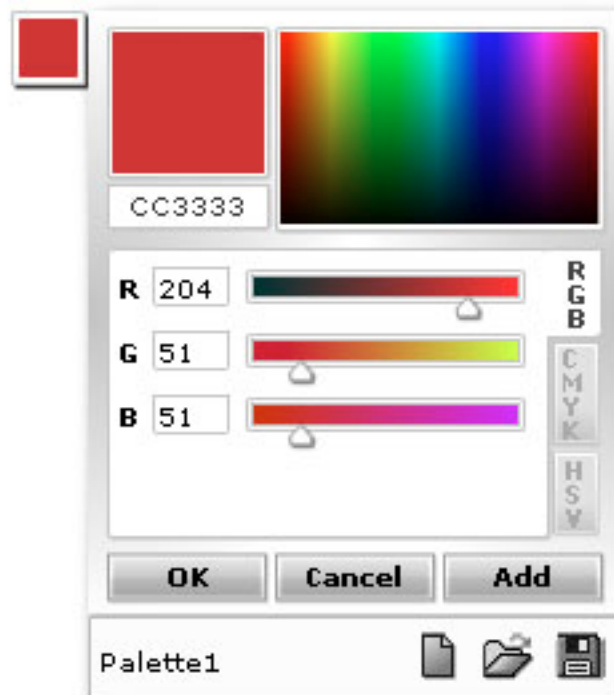
1. Select a color from the palette.
2. Enter the HEX code of the desired color in the textfield at the top of the palette.
3. Use the color dropper cursor to select any color anywhere on the stage.



### Advanced mode:

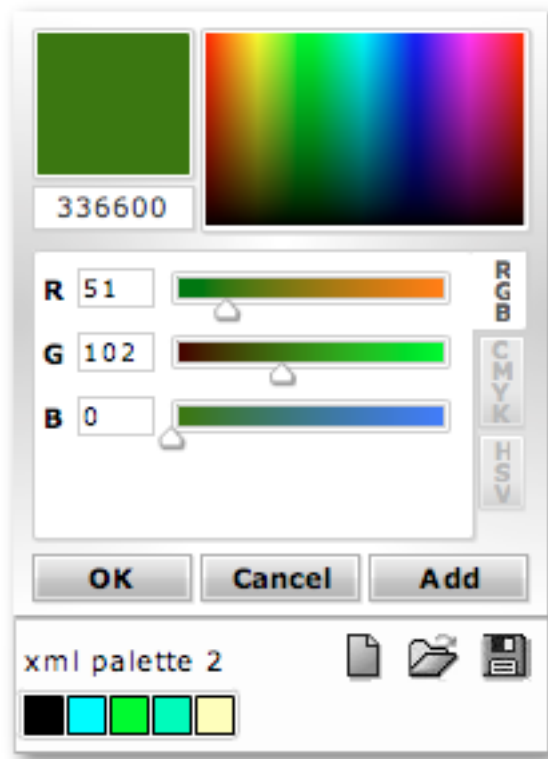
In advanced mode, the colors can be selected using any of the following methods:


1. Select the color from the color window.
2. Click on the RGB, CMYK or HSV tabs and use the sliders or textfields.
3. Select the color from the custom palette (see the next section for more information).
4. Enter the HEX code of the desired color in the textfield at the top of the palette.
5. Use the color dropper cursor to select any color anywhere on the stage.




## Creating a custom user defined color palette

Users can create their own sets of custom color palettes in advanced mode. To do this, select the color and click the **Add** button to add the color to the custom palette. The color will appear at the bottom of the color picker. To delete a color from the custom palette, press **ctrl** (Windows) or **cmd** (Mac) while clicking on the color from the custom palette.



The color palette can be saved for future reference. You can save multiple custom palettes. Enter a name for the custom palette just above the palette colors (or use the default name) and click on the save icon: 

Click on the open icon  to load a palette from the list of saved palettes.

Click on the new icon  to start a new empty custom palette.

# Defining the basic palette XML

The basic color palette can be redefined by using an XML file.

1. Open your favourite plain text editor (for example Notepad on Windows or TextEdit on Mac) and start a new file. *Note: If you are using TextEdit on Mac, choose Format > Make Plain Text*
2. Begin your file with the following line:

```
<?xml version="1.0" encoding="utf-8"?>
```

This is the standard xml declaration.

3. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<basicpalette itemsize="50" cols="4" border="2">  
</basicpalette>
```

Edit the basicpalette entry as follows:

The **itemsize** element defines size (in pixels) for each color square

The **cols** element defines the number of columns to display the palette colors

The **border** element defines space between the colors in the palette

4. Add color tags to the XML file for each color that you wish to add to the palette (the bold lines are the new additions):

```
<?xml version="1.0" encoding="utf-8"?>  
<basicpalette itemsize="50" cols="4" border="2">  
  <color value="0xAA0000"/>  
  <color value="0xAAAA00"/>  
  <color value="0xAA00AA"/>  
  <color value="0x00AAAA"/>  
  <color value="0xFFFF00"/>  
  <color value="0x0FFF00"/>  
  <color value="0x00FFF0"/>  
  <color value="0x000FFF"/>  
</basicpalette>
```

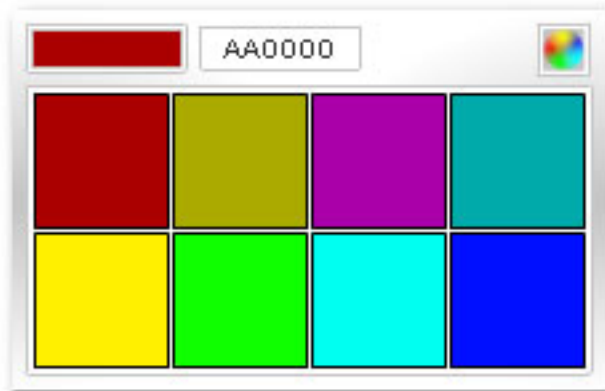
**color** specified the color using HEX value. For example: #FF0000 would be specified at 0xFF0000

5. Save the XML file to the same folder as your Flash file. In this example, we have given the XML file the name: *basicpalette.xml*

6. Return to your Flash file. Enter the name and path to the XML file that you just created in the **basicPaletteXML** parameter of the colorPicker that's on the stage.

*Note: If your .swf file will be in a different folder to the HTML file in which it is embedded, you should enter the path to the XML file, relative to the location of the .html file.*

This is how the basic color palette specified in this example will look:



# Creating pre-defined custom palettes

You can specify a set of pre-defined custom palettes which can be loaded in advanced mode.

1. Open your favourite plain text editor (for example Notepad on Windows or TextEdit on Mac) and start a new file. *Note: If you are using TextEdit on Mac, choose Format > Make Plain Text*
2. Begin your file with the following line:

```
<?xml version="1.0" encoding="utf-8"?>
```

This is the standard xml declaration.

3. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<palettes>  
</palettes>
```

4. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<palettes>  
    <palette paletteName="xml palette 1">  
    </palette>  
</palettes>
```

The **paletteName** element defines the name of the palette which will be displayed in load palette popup menu.

5. Add color tags for each color that you wish to add to this palette (the bold lines are the new additions):

```
<?xml version="1.0" encoding="utf-8"?>  
<palettes>  
    <palette paletteName="xml palette 1">  
        <color id="0" value="0xAA0000"/>  
        <color id="1" value="0xAAAA00"/>  
        <color id="2" value="0xAA00AA"/>  
    </palette>  
</palettes>
```

**id** is a unique identified for the color

**color** specified the color using HEX value. For example: #FF0000 would be specified at 0xFF0000

6. Add more palettes in the same pattern. For example:

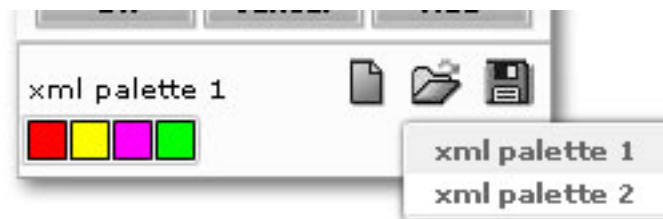
```
<?xml version="1.0" encoding="utf-8"?>
<palettes>
  <palette paletteName="xml palette 1">
    <color id="0" value="0xAA0000"/>
    <color id="1" value="0xAAAA00"/>
    <color id="2" value="0xAA00AA"/>
  </palette>
  <palette paletteName="xml palette 2">
    <color id="0" value="0x0FFF00"/>
    <color id="1" value="0x00FFF0"/>
    <color id="2" value="0x000FFF"/>
  </palette>
</palettes>
```

7. Save the XML file to the same folder as your Flash file. In this example, we have given the XML file the name: *palettes.xml*

8. Return to your Flash file. Enter the name and path to the XML file that you just created in the **palettesXML** parameter of the colorPicker that's on the stage.

*Note: If your .swf file will be in a different folder to the HTML file in which it is embedded, you should enter the path to the XML file, relative to the location of the .html file.*

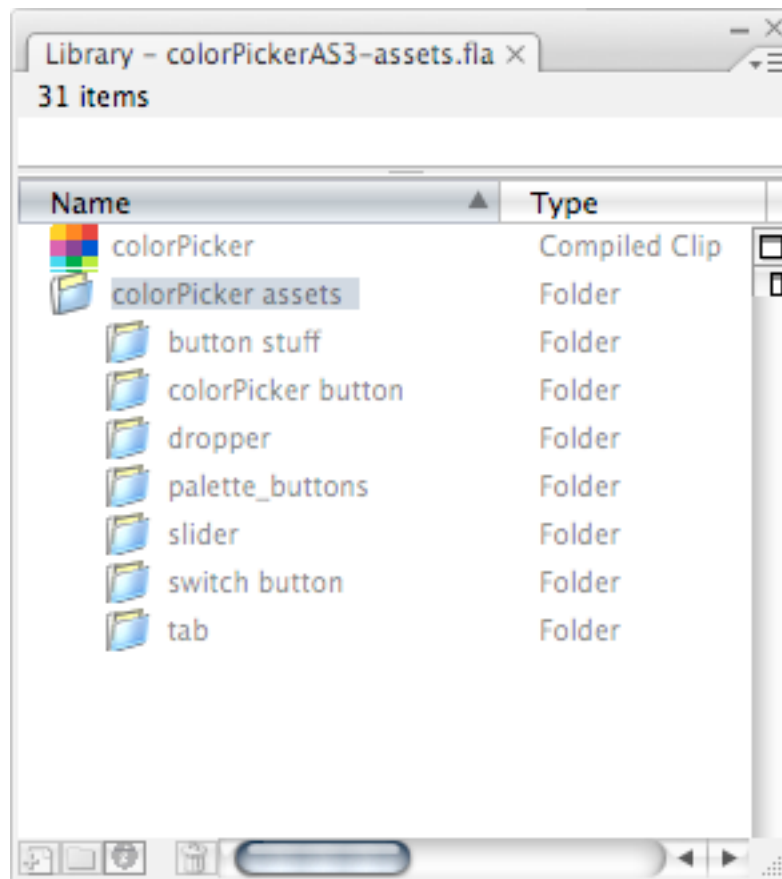
The palette will be added to the list of custom palettes like this:



## Skinning the colorPicker

Besides changing the colors of the background and labels through the Component Inspector, you can also change the look of the sliders, icons, tabs and dropper cursor. This is how you would change any of these elements:

1. Ensure that the library panel is open: *ctrl+L* (win) or *cmd+L* (mac)
2. From the menu in Flash select: *Window > Common Libraries > colorPickerAS2-assets* (for AS2) or *Window > Common Libraries > colorPickerAS3-assets* (for AS3).
3. You should see a folder called **colorPicker assets**. Double click on this folder to expand it and to show all of the available editable assets.



4. Drag and drop the folder containing the elements that you wish to edit from the common library into your current library. You may also drag the complete **colorPicker assets** folder into your library if you wish.
5. Once the folders are in your library, you may double click on the folders to view and edit the the movie clips or images for those elements.

# ActionScript 2.0 API

## Events

### colorOk

#### Availability

Flash Player 8

#### Description

Event; called when a color is selected. Event is dispatched to the component

#### Example

```
var listener = new Object();
listener.colorOk = function(event)
{
    trace(event.color); // detect which color was selected
}
colorPicker.addEventListener("colorOk" , listener.colorOk);
```

### colorChanged

#### Availability

Flash Player 8

#### Description

Event; called when a color in preview section of open color picker changes. Event is dispatched to the component. This event might be used for some preview action implementations

#### Example

```
var listener = new Object();
listener.colorChanged = function(event)
{
    trace(event.color); // detect which color is being previewed
}
colorPicker.addEventListener("colorChanged" , listener.colorChanged);
```

## colorCancel

### Availability

Flash Player 8

### Description

Event; called when a color selection was canceled. Event is dispatched to the component.

### Example

```
var listener = new Object();
listener.colorCancel = function(event)
{
    trace(event.color); // detect an old color to cancel to
}

colorPicker.addEventListener("colorCancel" , listener.colorCancel);
```

## Methods

### getPopup

#### Availability

Flash Player 8

#### Description

Method; allows you to get a popup object of the color picker. Accept string as a parameter, which indicates the mode of popup.

#### Example

```
var advPopup = colorPicker.getPopup(colorPicker.MODEADVANCED);
var basicPopup = colorPicker.getPopup(colorPicker.MODEBASIC);
```

### showPopup

#### Availability

Flash Player 8

#### Description

Method; shows the popup.

#### Example

```
colorPicker.showPopup(false); // the current popup will be hidden
```

## positionPopup

### Availability

Flash Player 8

### Description

Method; positions a shown popup at default location.

### Example

```
colorPicker.positionPopup();
```

## getColor

### Availability

Flash Player 8

### Description

Method; returns the color currently selected in color picker.

### Example

```
var curColor:Number = colorPicker.getColor();
```

## getHexColor

### Availability

Flash Player 8

### Description

Method; returns the color currently selected in color picker as HEX string.

### Example

```
var curColor:String = colorPicker.getHexColor();
```

## setColor

### Availability

Flash Player 8

### Description

Method; sets the selected color for color picker

### Example

```
colorPicker.setColor(0xFF0000); //sets the color to red
```

# Properties

## mode

### Availability

Flash Player 8

### Description

Property; controls the mode of color picker's popup. Could be basic or advanced.

The default value is "basic"

### Example

```
colorPicker.mode = colorPicker.MODEADVANCED; // set the advanced mode  
colorPicker.mode = colorPicker.MODEBASIC; // set the basic mode
```

## defaultColor

### Availability

Flash Player 8

### Description

Property; controls the default color of color picker

### Example

```
advColorPicker.defaultColor = 0xFF0000;
```

## palettesXml

### Availability

Flash Player 8

### Description

Property; specifies the xml file with predefined palettes for advanced mode.

### Example

```
advColorPicker.palettesXml = "palettes.xml";
```

## palettesFamily

### Availability

Flash Player 8

### Description

Property; specifies the palettes family name. All palettes saved within a similar palettes family will share the palettes. But these palettes will not be available to color pickers with other palette family name

### Example

```
advColorPicker.palettesFamily = "My Palettes";
```

## defaultPaletteName

### Availability

Flash Player 8

### Description

Property; specifies default name of palette. If the palette with specified name exist it is automatically loaded, otherwise this value is used for new palette name

### Example

```
advColorPicker.defaultPaletteName = "Load me first";
```

## basicPaletteXML

### Availability

Flash Player 8

### Description

Property; specifies the path to xml file, where the basic palette might be defined.

### Example

```
advColorPicker.basicPaletteXML = basicPalette.xml;
```

# ActionScript 3.0 API

## Events

### colorOk

#### Availability

Flash Player 9

#### Description

Event; called when a color is selected. Event is dispatched to the component.

#### Example

```
function onColorOk(e:Object) {  
    trace('color selected:'+e.color)  
}  
colorPicker.addEventListener("colorOK", onColorOk);
```

### colorChanged

#### Availability

Flash Player 9

#### Description

Event; called when a color in preview section of open color picker changes. Event is dispatched to the component. This event might be used for some preview action implementations

#### Example

```
function onColorChange(e:Object) {  
    trace('color changed:'+e.color)  
}  
colorPicker.addEventListener("colorChanged", onColorChange);
```

## colorCancel

### Availability

Flash Player 9

### Description

Event; called when a color selection was canceled. Event is dispatched to the component.

### Example

```
function onColorCancel(e:Object) {
    trace('color changed:'+e.color)
}
colorPicker.addEventListener("colorCancel", onColorCancel );
```

## Methods

### getPopup

#### Availability

Flash Player 9

#### Description

Method; allows you to get a popup object of the color picker. Accept string as a parameter, which indicates the mode of popup.

#### Example

```
var advPopup = colorPicker.getPopup(colorPicker.MODEADVANCED);
var basicPopup = colorPicker.getPopup(colorPicker.MODEBASIC);
```

### showPopup

#### Availability

Flash Player 9

#### Description

Method; shows the popup.

#### Example

```
colorPicker.showPopup(false); // the current popup will be hidden
```

## positionPopup

### Availability

Flash Player 9

### Description

Method; positions a shown popup at default location.

### Example

```
colorPicker.positionPopup();
```

## getColor

### Availability

Flash Player 9

### Description

Method; returns the color currently selected in color picker.

### Example

```
var curColor:Number = colorPicker.getColor();
```

## getHexColor

### Availability

Flash Player 9

### Description

Method; returns the color currently selected in color picker as HEX string.

### Example

```
var curColor:String = colorPicker.getHexColor();
```

## setColor

### Availability

Flash Player 9

### Description

Method; sets the selected color for color picker

### Example

```
colorPicker.setColor(0xFF0000); //sets the color to red
```

# Properties

## mode

### Availability

Flash Player 9

### Description

Property; controls the mode of color picker's popup. Could be basic or advanced.

The default value is "basic"

### Example

```
colorPicker.mode = colorPicker.MODEADVANCED; // set the advanced mode  
colorPicker.mode = colorPicker.MODEBASIC; // set the basic mode
```

## defaultColor

### Availability

Flash Player 9

### Description

Property; controls the default color of color picker

### Example

```
advColorPicker.defaultColor = 0xFF0000;
```

## palettesXml

### Availability

Flash Player 9

### Description

Property; specifies the xml file with predefined palettes for advanced mode.

### Example

```
advColorPicker.palettesXml = "palettes.xml";
```

## palettesFamily

### Availability

Flash Player 9

### Description

Property; specifies the palettes family name. All palettes saved within a similar palettes family will share the palettes. But these palettes will not be available to color pickers with other palette family name

### Example

```
advColorPicker.palettesFamily = "My Palettes";
```

## defaultPaletteName

### Availability

Flash Player 9

### Description

Property; specifies default name of palette. If the palette with specified name exist it is automatically loaded, otherwise this value is used for new palette name

### Example

```
advColorPicker.defaultPaletteName = "Load me first";
```

## basicPaletteXML

### Availability

Flash Player 9

### Description

Property; specifies the path to xml file, where the basic palette might be defined.

### Example

```
advColorPicker.basicPaletteXML = basicPalette.xml;
```

# Help

This component is fully supported by the Flashloaded support team through our support forum. You will also find tips and additional information in the forum as well as announcements for version updates: [colorPicker Support Forum](#)

*Note: In order to post a question in the forum, you will need to [register](#) by creating a username and password. This registration differs from your account login.*