

## Contents

- [1 Introduction](#)
- [2 Prerequisites](#)
- [3 Installing](#)
- [4 How to use](#)
  - ◆ [4.1 Properties](#)
  - ◆ [4.2 Events on check boxes](#)
    - ◇ [4.2.1 onChange](#)
  - ◆ [4.3 Creating and setting up a button](#)
  - ◆ [4.4 Formatting the label of the CheckBox](#)
- [5 Downloads](#)
- [6 Conclusion](#)

## Introduction

The default CheckBox component of the Flash CS3/CS4 is very useful to web pages and desktop applications, but it's not true to mobile applications because of the small screen and big resolution, mainly in newer devices (5800, N97). The component shown here is a custom CheckBox, where the user can provide your own MovieClip as the button.

The check boxes are combined in groups and the user can select one or more check boxes in a group at one time. This behaviour is different of radio buttons, where clicking on another causes the existing one to become unchecked.

## Prerequisites

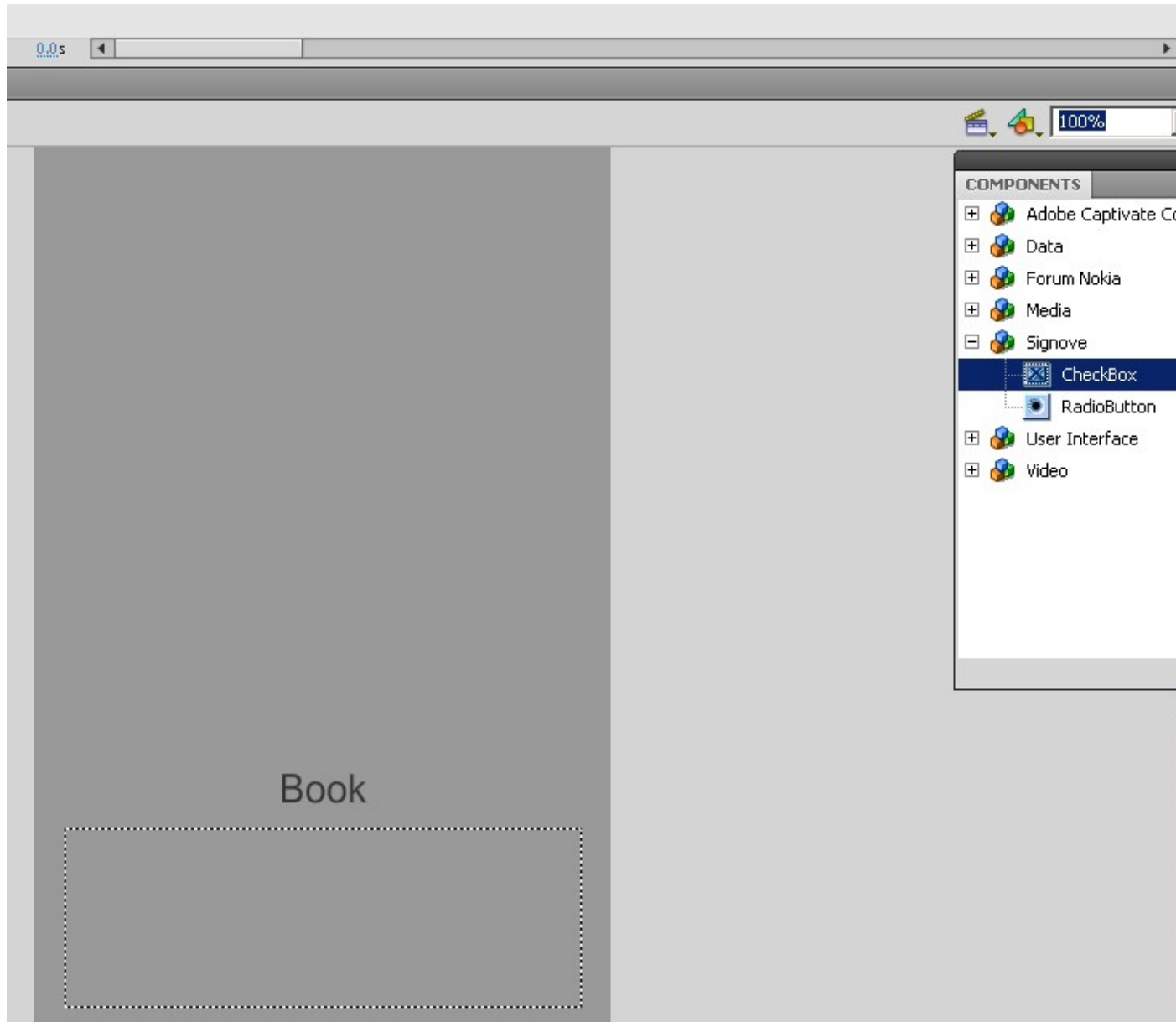
This component was developed in Flash Player 8 and ActionScript 2.0. It was tested on Nokia Xpress Music 5800.

## Installing

It's very simple to install the component, just download the MXP file and open it in the Adobe Extension Manager.

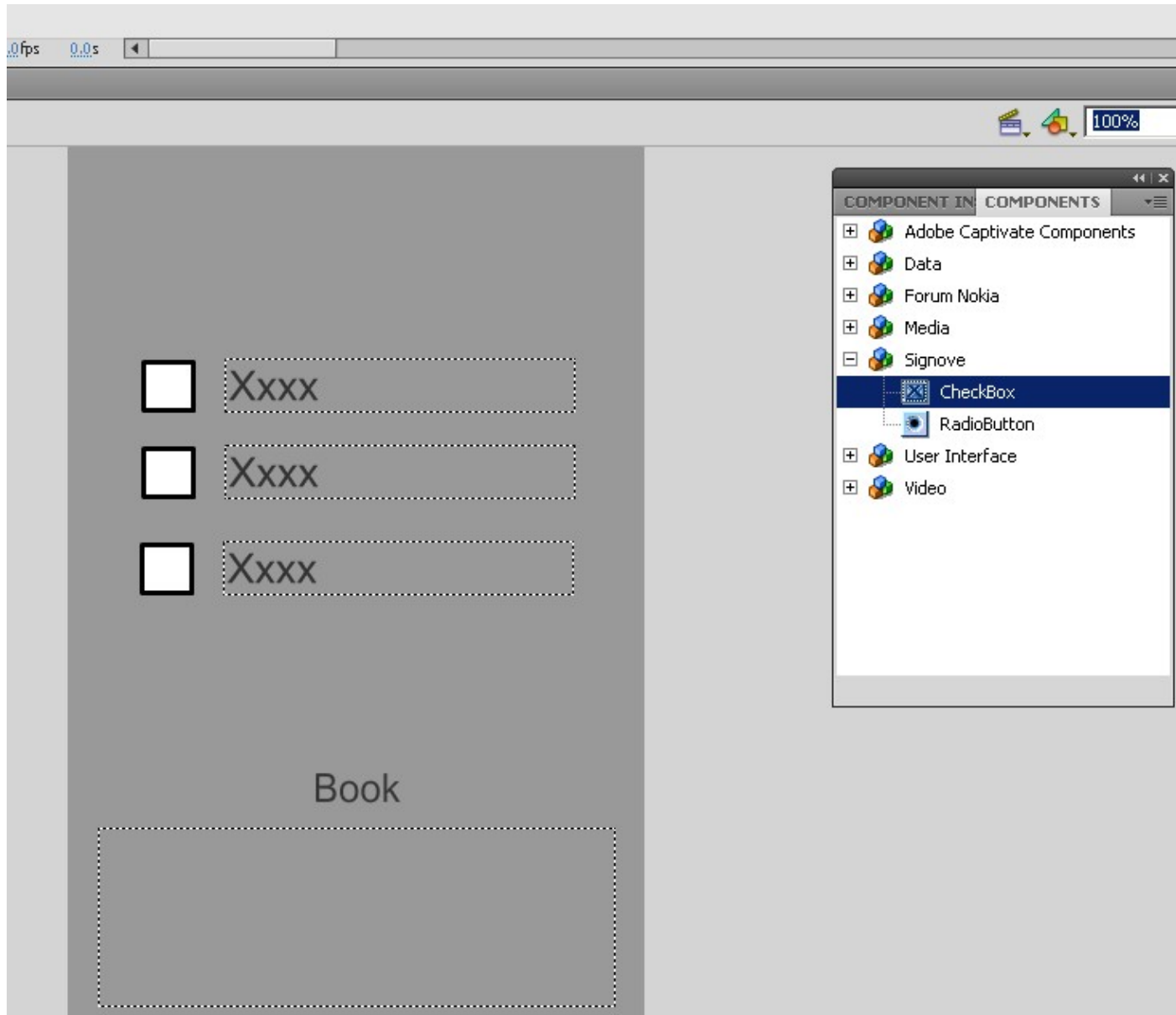
## How to use

After the installation, the component appears in the Component panel (Window -> Components -> Signove -> CheckBox) like shown in the below figure. Now, it's just drag and drop the component to the Stage and configure the properties.



## Properties

To view the properties of the component go to (Window -> Component Inspector) and select the component in the Stage. There are just five properties in the component, as shown in the below figure.



- **buttonMovieClip** : This movie clip is the button of the check box. We'll see more details about it in "Creating and setting up a button" section.
- **data** : The value of the check box. Must be unique in a group.
- **groupName** : The name of the group. Check boxes in the same group must have the same groupName property.
- **label** : The text shown in the check box.
- **selected** : If the check box must appears selected or not when the application starts.

## Events on check boxes

### onChange

There is just one piece of ActionScript code needed in this event. It's shown below.

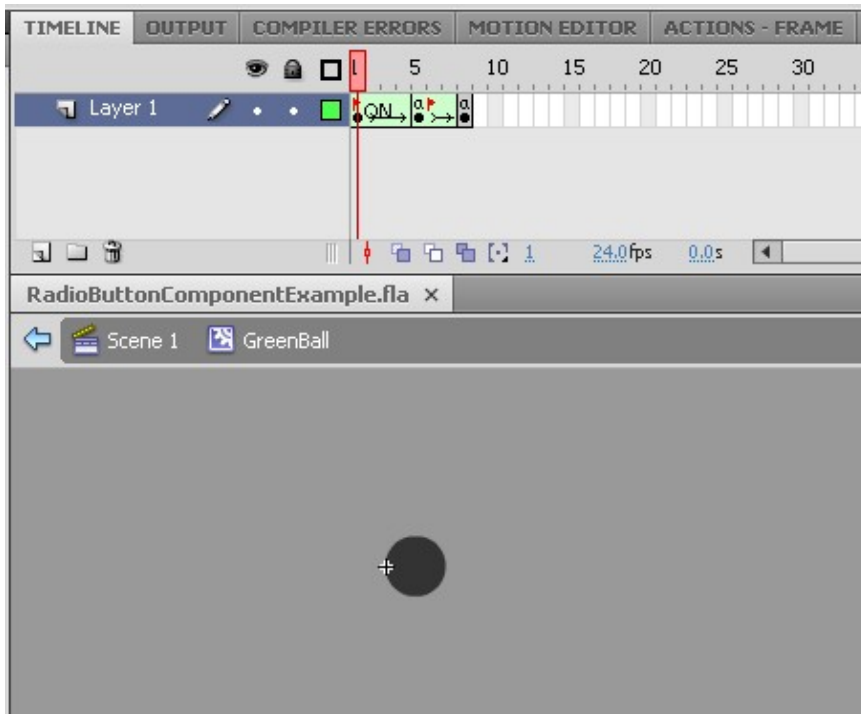
```
//Import the CheckBoxGroup class.
import com.signove.CheckBox.CheckBoxGroupClass

//Set a listener to the onChange event. When the user click on a check box, the onChange method i
var onChangeListener:Object = new Object();
onChangeListener.onChange = function(evt):Void{
    option_value.text = evt.target.selectedData;
}
//Retrieve the group and set the listener.
var group = CheckBoxGroupClass.getGroup('writers');
group.addEventListener('onChange', onChangeListener);
```

The code shown above is about events on check boxes. In the first version of the component, there is just one event to be caught (onChange).

### Creating and setting up a button

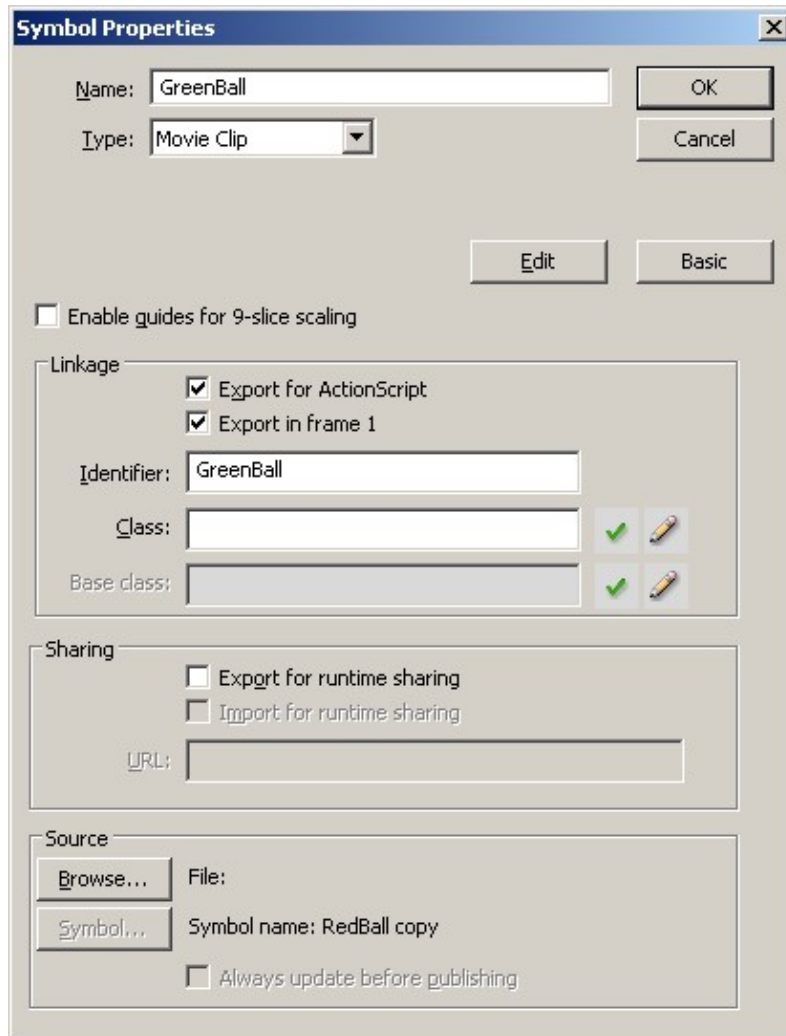
The component comes with a default button, but the developer can create your own button. The button is just a MovieClip with few restrictions. It has to have two frames named 'ON' and 'OFF', one to start the animation when the user check a check box and another to start the animation when the user uncheck a check box. An example is shown in the figure below.



## CheckBox\_Component\_for\_Flash\_Lite

After the creation of the MovieClip (don't forget the 'ON' and 'OFF' frames) you have to set a 'Identifier' to the MovieClip. This identifier is used in buttonMovieClip property of the check box.

To create an identifier to the MovieClip, go to (Window -> Library), select your new MovieClip, right-click on it and select 'Properties...' option. You will see a screen like the shown below. Check both options 'Export for ActionScript' and 'Export in frame 1' and type an 'Identifier'



## Formatting the label of the CheckBox

The developer can format the label of the check box with an instance of the TextFormat class. The code below shows how to do that.

```
var my_fmt:TextFormat = new TextFormat();
my_fmt.bold = true;
my_fmt.size = 24;
my_fmt.color = 0xFF0000;

//The instance name of the CheckBox component
check_1.setTextFormat(my_fmt);
```

## Downloads

- The source code of the component, download [here](#)
- The MXP file, ready to install, download [here](#)
- A simple example using this component, download [here](#)

## Conclusion

This is the second component i've created. It's very similar to the RadioButton component posted [here](#). If you want to see more, go to [here](#)