

Contents

- [1](#)
[Introduction](#)
- [2](#)
[Design](#)
- [3](#)
[Validation](#)
- [4](#)
[Download](#)

Introduction

There can be several situations in application UI development, where the user is expected to enter values for IP addresses or Dates or Time. It must be understood that these textfields are no ordinary ones as they have separators like Dot in the case of IP addresses or colon in Time. A general IP address has four numerals separated by three dots. And a Time entity has two numerals separated by a single Colon. In addition to this, one must also validate the entries in those fields. Like for example, IP address numerals can have values only between 0 and 255. And the time entity can have only between 0 - 23 in HOURS field and 0 ? 59 in MINUTES field.

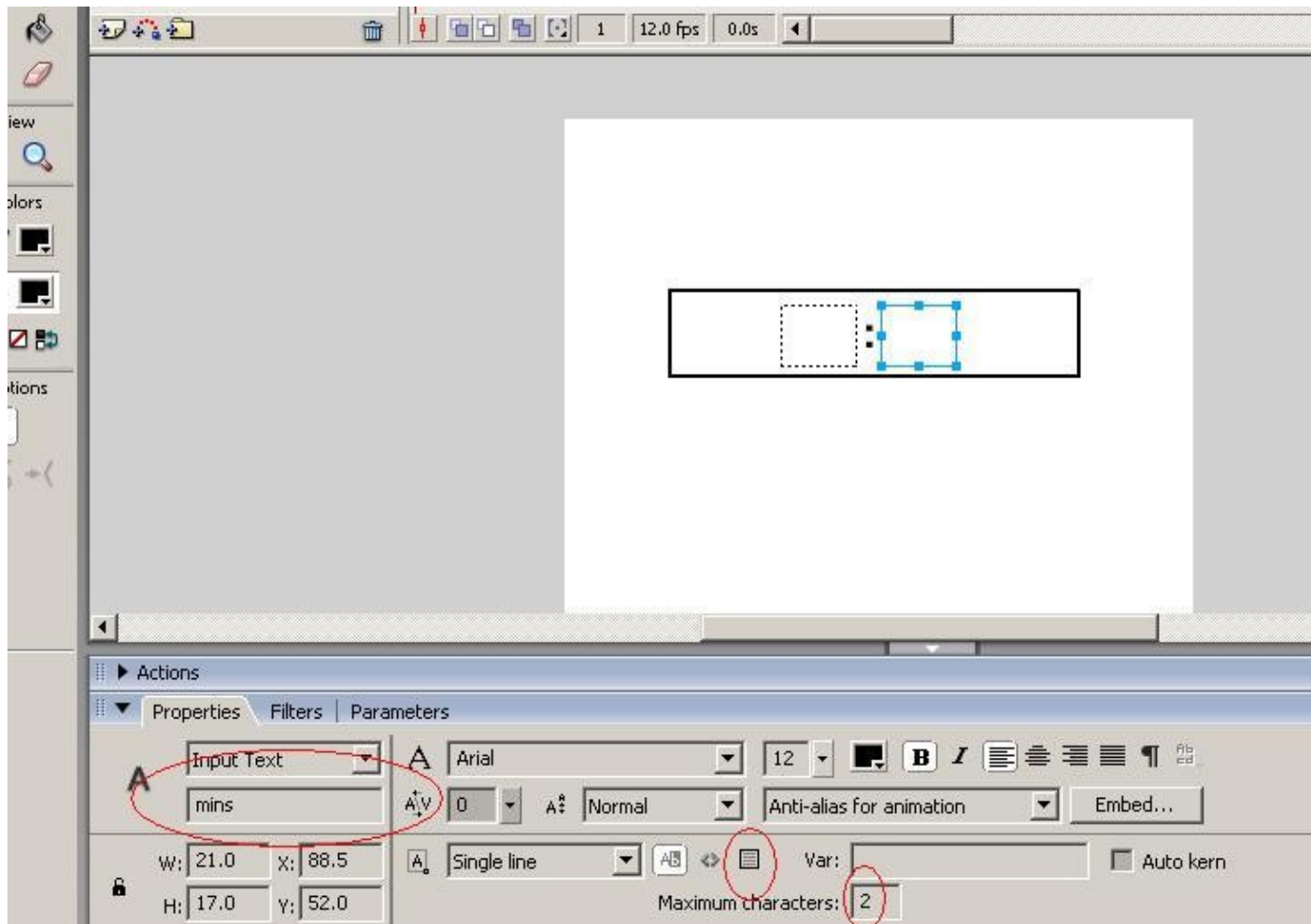
Design

To design these kinds of special UI components requires guile. Here, we demonstrate the creation of UI of the Time component. The Time component has only two fields, as you know.

1. First create two input textfields and give them instance names ? hrs and mins. In the properties panel of each of them, make the maximum no. of Characters field to 2.
2. Most important to disable the show border option of these textFields.
3. Place these two in proximity and separate them using a Colon symbol. A common method is to create a Static textfield with value ?:. Now place this Static TextField in between the two input textfields.
4. Now, create a bounding rectangle around all these components. Essentially, with no fill color.
5. We also need to restrict the values in these fields. Entry of Alphabets inside such textfields is prohibited. Restriction can be made using the SetInputTextType Fsccommand. For this feature, add the following code

```
mins.variable = "InputTextVar";
hrs.variable = "AnotherVar";
fscommand2("SetInputTextType", "InputTextVar", "Numeric");
fscommand2("SetInputTextType", "AnotherVar", "Numeric");
```

Creating_specialized_TextFields_in_Flash_Lite



Validation

This gives the look of a TextField (although we have two). Now this is not sufficient as we have still not validated the entries in the TextFields. The user (during input) can enter say 34 in HRS textField, which is unacceptable. Hence certain validations need to be performed. As per the UI in most Nokia phones, validation is done at the very moment of change of values. In order to retain focus after an invalid input, we add the same code to onKillFocus event too.

```
hrs.onKillFocus = hrs.onChanged = function(asFocus:Object) {
    textColor = 0x000000;
    if (parseInt(hrs.text)>23) {
        textColor = 0xFF0000;
        Selection.setFocus("hrs");
    }
};
mins.onKillFocus =mins.onChanged = function(asFocus:Object) {
    textColor = 0x000000;
    if (parseInt(mins.text)>59) {
        textColor = 0xFF0000;
        Selection.setFocus("mins");
    }
}
```

Creating_specialized_TextFields_in_Flash_Lite

```
};
```

On Flash Lite 2.1 handsets, its better to use the inline textentry feature. This means a separate dialog box will not open to receive input, instead it can be done inline. This too can be done using a fscommand functional feature, as shown below.

```
var focusListener:Object = new Object();  
focusListener.onSetFocus = function(oldFocus, newFocus) {  
fscommand("activateTextField", "");  
};  
hrs.addListener(focusListener);  
mins.addListener(focusListener);
```

Download

Sample FLA is available [Media:Customised_textFields.zip](#) for reference.

--[Manikantan](#) 14:36, 26 April 2009 (EEST)