

IOS - TEXT FIELD

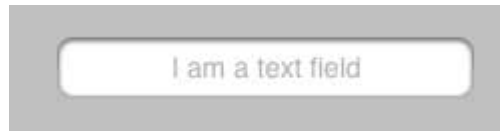
http://www.tutorialspoint.com/ios/ios_ui_elements_text_field.htm

Copyright © tutorialspoint.com

Use of Text Field

A text field is a UI element that enables the app to get user input.

A UITextField is shown below.

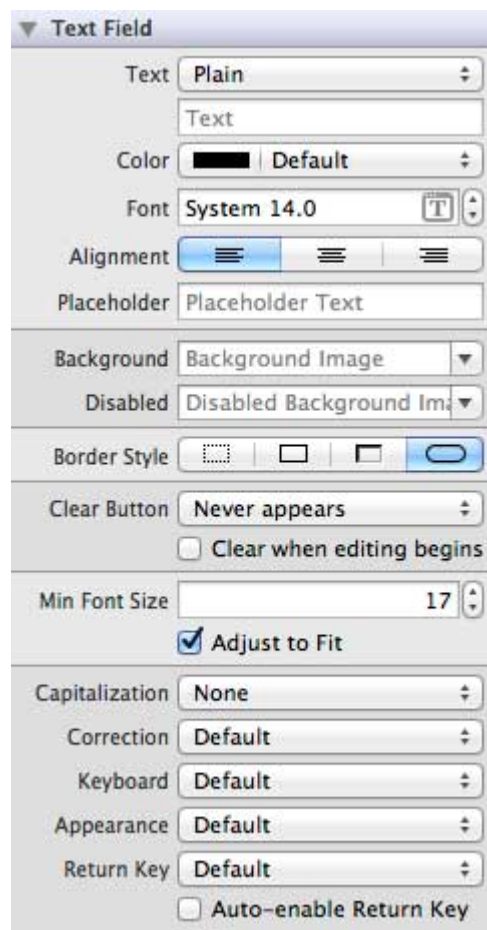


Important Properties of Text Field

- Placeholder text which is shown when there is no user input
- Normal text
- Auto correction type
- Key board type
- Return key type
- Clear button mode
- Alignment
- Delegate

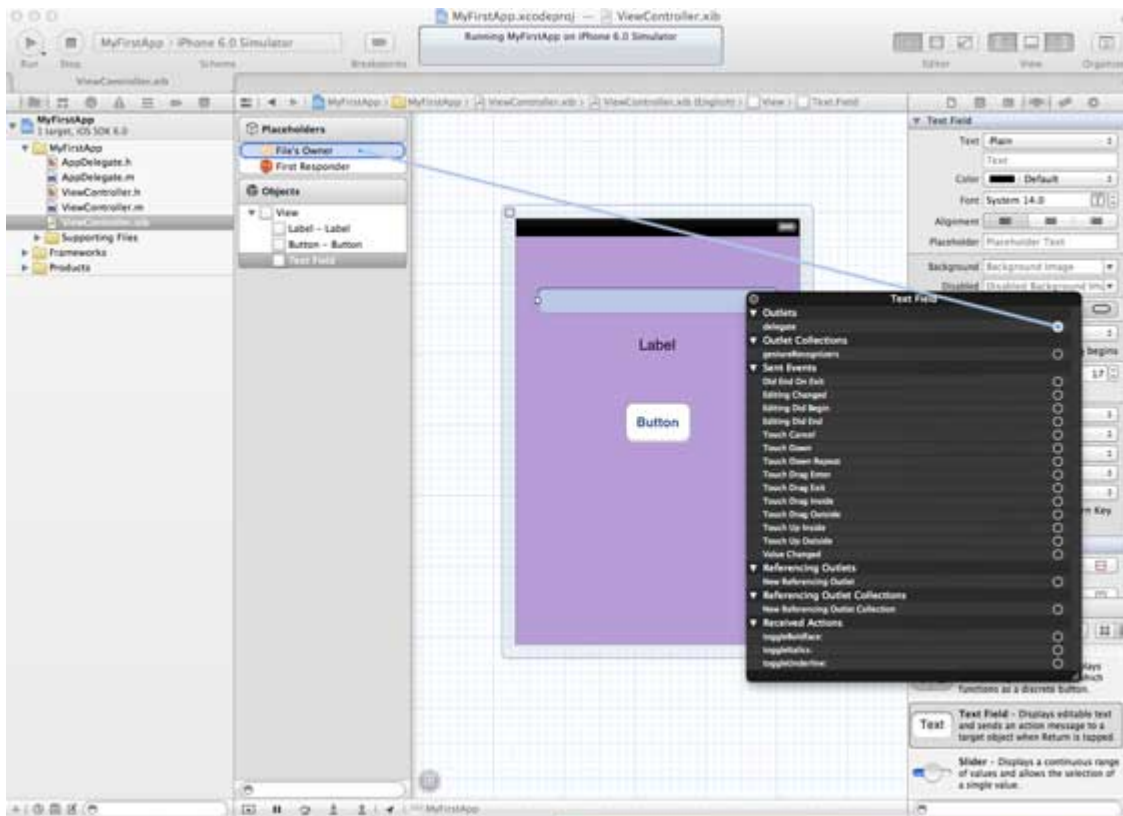
Updating Properties in xib

You can change the text field properties in xib in the attributes inspector in the utilities area *rightsideoftheWindow*.



Text Field Delegates

We can set delegate in interface builder by right-clicking on the UIElement and connect it to the file owner as shown below.



Steps in Using Delegates

Step 1. Set delegate as shown in the above figure.

Step 2. Add delegate to which your class responds to.

Step 3. Implement the text field Delegates, the important text field delegates are as follows –

```
(void)textFieldDidBeginEditing:(UITextField *)textField
```

```
(void)textFieldDidEndEditing:(UITextField *)textField
```

Step 4. As the name suggests, the above two delegates are called once we begin editing the text field and end editing respectively.

Step 5. For other delegates, please refer UITextFieldDelegate Protocol reference.

Sample Code and Steps

Step 1. We'll use the sample application created for UI elements.

Step 2. Our ViewController class will adopt **UITextFieldDelegate** and our **ViewController.h** file is updated as follows –

```
#import <UIKit/UIKit.h>

// You can notice the addition of UITextFieldDelegate below
@interface ViewController : UIViewController<UITextFieldDelegate>

@end
```

Step 3. Then we add a method **addTextField** to our ViewController.m file.

Step 4. Then we call this method in our viewDidLoad method.

Step 5. Update **viewDidLoad** in **ViewController.m** as follows –

```
#import "ViewController.h"
@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    //The custom method to create our textfield is called
    [self addTextField];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (void)addTextField{
    // This allocates a label
    UILabel *prefixLabel = [[UILabel alloc] initWithFrame:CGRectZero];
    //This sets the label text
    prefixLabel.text=@"## ";
    // This sets the font for the label
    [prefixLabel setFont:[UIFont boldSystemFontOfSize:14]];
    // This fits the frame to size of the text
    [prefixLabel sizeToFit];

    // This allocates the textfield and sets its frame
    UITextField *textField = [[UITextField alloc] initWithFrame:
    CGRectMake(20, 50, 280, 30)];

    // This sets the border style of the text field
    textField.borderStyle = UITextBorderStyleRoundedRect;
    textField.contentVerticalAlignment =
    UIControlContentVerticalAlignmentCenter;
    [textField setFont:[UIFont boldSystemFontOfSize:12]];

    //Placeholder text is displayed when no text is typed
    textField.placeholder = @"Simple Text field";

    //Prefix label is set as left view and the text starts after that
    textField.leftView = prefixLabel;

    //It set when the left prefixLabel to be displayed
    textField.leftViewMode = UITextFieldViewModeAlways;

    // Adds the textField to the view.
    [self.view addSubview:textField];

    // sets the delegate to the current class
    textField.delegate = self;
}

// pragma mark is used for easy access of code in Xcode
#pragma mark - TextField Delegates

// This method is called once we click inside the textField
- (void)textFieldDidBeginEditing:(UITextField *)textField{
    NSLog(@"Text field did begin editing");
}
```

```

}

// This method is called once we complete editing
-(void)textFieldDidEndEditing:(UITextField *)textField{
    NSLog(@"Text field ended editing");
}

// This method enables or disables the processing of return key
-(BOOL) textFieldShouldReturn:(UITextField *)textField{
    [textField resignFirstResponder];
    return YES;
}

- (void)viewDidUnload {
    label = nil;
    [super viewDidUnload];
}

@end

```

Step 6. When we run the application, we'll get the following output.



Step 7. The delegate methods are called based on user action. See the console output to know when the delegates are called

Loading [MathJax]/jax/output/HTML-CSS/jax.js