

## CocoaCast Express 4 notes

### Demo - Build a Cocoa app from scratch

- Download F-Script from the Web
- Create a new XCode Cocoa Application Project
- Drag&Drop FScript.framework to the project folder
- Add a new Copy Phase in the application target
- Drag&Drop FScript.framework to the copy phase
- Create an ApplicationController class

### In ApplicationController.h:

- Add FScript.h Import
- Add “scriptCommand”, “scriptResult” properties, FSInterpreter instance variables and execute method to ApplicationController.h:

```
#import <Cocoa/Cocoa.h>
#import <FScript/FScript.h>

@interface ApplicationController : NSObject {
    NSString* scriptCommand;
    NSString* scriptResult;
    FSInterpreter *interpreter;
}

@property (readwrite, copy) NSString *scriptCommand;
@property (readwrite, copy) NSString *scriptResult;

- (IBAction)execute:(id)sender;

@end
```

### In ApplicationController.m:

- Add “scriptCommand” and “scriptResult” synthesizes, dealloc and “execute” method:

```
#import "AppController.h"

@implementation ApplicationController

@synthesize scriptCommand;
@synthesize scriptResult;

- (void)dealloc
{
```

```

        if (interpreter)
            [interpreter release];

        [super dealloc];
    }

- (IBAction)execute:(id)sender
{
    if (!interpreter)
        interpreter = [[FSInterpreter alloc] init];

    // execute the F-Script command
    FSInterpreterResult *execResult =
        [interpreter execute:self.scriptCommand];

    // test status of the result
    if ([execResult isOk])
    {
        id resultResult = [execResult result];

        // print the result
        if (resultResult != nil)
        {
            self.scriptResult = [resultResult printString];
        }
        else
        {
            self.scriptResult =
                @"Error executing the F-Script command";
        }
    }
    else
    {
        // print an error message
        self.scriptResult =
            [NSString stringWithFormat:@"%@", character %d",
             [execResult errorMessage],
             [execResult errorRange].location];
    }
}

@end

```

That's it for the code!

- Open the MainMenu.nib file

- Add an object controller and set its class to “AppController”
- Add a SplitView
- Add a NSTextField to the splitview top pane and bind its value to scriptCommand. Uncheck the “Rich text” checkbox for the text field.
- Add another NSTextField to the splitview bottom pane and bind its value to scriptResult.
- Add a NSButton and name it “Execute”
- Link the NSButton to the AppController execute: method.
- Close Interface Builder
- Build, run and enjoy!