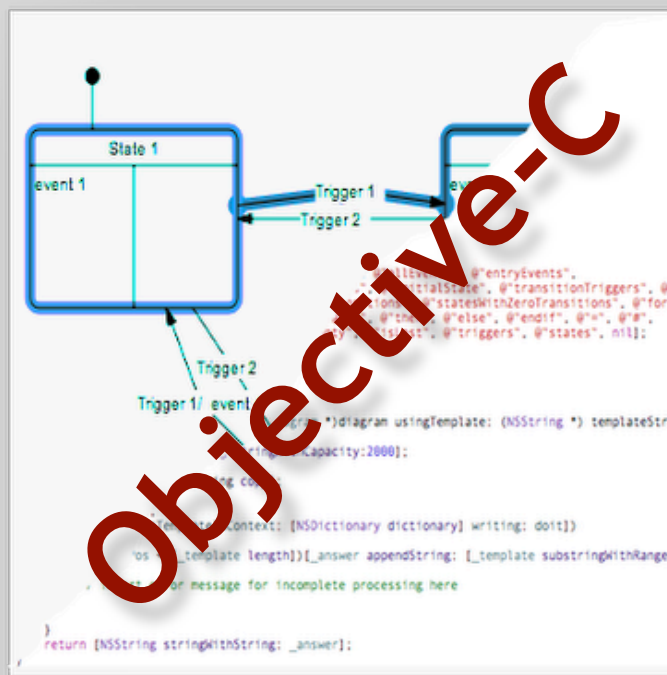
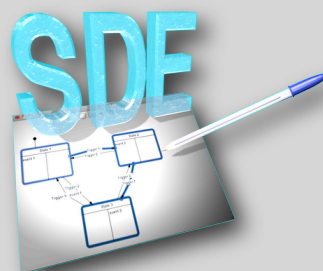


A Template for Objective-C Code Generation



Code Templates for the State Diagram Editor



License

This product, a Template for the State Diagram Editor is copyrighted shareware, not public-domain software. You may use the unregistered version with certain restrictions for personal, non-commercial purposes. Upon request, you will receive an unlimited license for a period of thirty days. To use the software after this period for commercial purposes, you have to register it. The commercial license is bound to one specific machine, but may be transferred at no cost to another one.

Stefan Sachs does not warrant that State Diagram Editor will meet your specific needs or requirements, nor that the operation of State Diagram Editor will in all instances be uninterrupted or error-free. Because software is inherently complex and may not be completely free of errors, you must verify all results before relying on them.

Stefan Sachs shall not be liable for any direct, indirect, consequential, special, incidental, or other damages, including but not limited to damages for lost profits, interruption of business, lost or corrupted data or programs, system crashes, or diversion of system or other resources, arising out of or relating to this agreement or the performance, quality, results, use of, or inability to use the State Diagram Editor or any support, advice, products, or services which are or should have been provided to you, even if Stefan Sachs has been advised of the possibility of such damages.

State Diagram Editor may contain technical protections that restrict use to certain equipment, to a single personal computer or workstation, or to a certain time period, and may provide other restrictions, all of which shall be consistent with the License. Stefan Sachs shall not be liable for any damages incurred as a result of such technical protections or the failure of State Diagram Editor to operate outside the scope of the license. Stefan Sachs shall not be liable for any claim asserted against you by any other party, including but not limited to claims for infringement of copyright or other intellectual property rights.

Some jurisdictions do not permit the exclusion or limitation of implied warranties or consequential damages, so some of the above exclusions or limitations may not apply to you.

Stefan Sachs may terminate the License with or without cause at any time immediately and without notice if you materially breach any term of this Agreement or infringe any of Stefan Sachs's rights in State Diagram Editor.

Stefan Sachs may terminate the License upon ten days written notice if Stefan Sachs believes that your use of State Diagram Editor may infringe any third party rights. If such termination occurs within 24 months of your first installation of State Diagram Editor, Stefan Sachs will provide you a modified, non-infringing version of State Diagram Editor and reinstate the License.

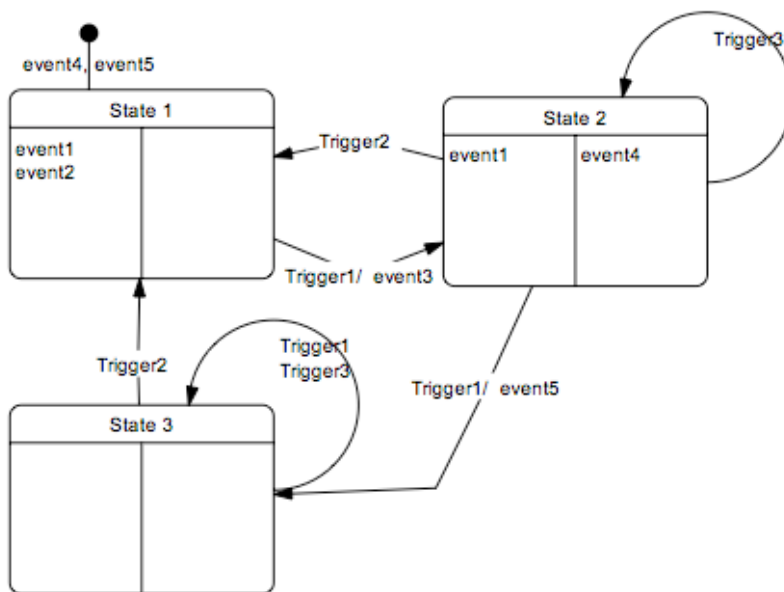
Immediately upon termination of the License, you must stop using State Diagram Editor and destroy all copies of State Diagram Editor within your possession, custody, or control.

You may not assign, sublicense, or otherwise transfer the License or any other rights arising under this Agreement. Any purported transfer in violation of this paragraph shall be void

A Template for Objective-C Code Generation

The template Objective-C.codetemplate is used to generate an Objective-C class which can be used in the responder chain to process actions from UI elements or from nil targeted actions sent to NSApp. Events defined in the diagram are processed as nil-targeted actions too, so interconnecting state diagrams is done simply by using identical names for triggers and events.

In this code template, zero-transitions are treated differently: While a zero-transition causes neither a state change, nor does it trigger an event, it causes the object to answer YES to the respondsToSelector: message. So the trigger will be processed by the diagram and will not be transferred to the next object in the responder chain. Therefore defining zero transitions in a state will block the triggers of these zero transitions from all responders following, while triggers not processed in the actual state will be sent down the responder chain.



The example diagram

States	Trigger1	Trigger2	Trigger3
State 1	State 2	State 1	State 1
State 2	State 3	State 1	State 2
State 3	State 3	State 1	State 3

The table representation of the example diagram. Zero transitions can be identified by the yellow cell background color

Interface Definition generated from the Example Diagram:

```

// Code generated by StateDiagramEditor (c) Copyright 2010 Stefan Sachs www.ssachs.de
// (c) Copyright for template 2009 Stefan Sachs
// generated from diagram demo at 2010-03-12 17:18:00 +0100
#import <Cocoa/Cocoa.h>
@interface demo: NSObject {
}
  
```

```

    int _actualState;
    NSResponder * precedingResponder;
}

-(void)initDiagram;
-(void)becomeResponderAfter: (NSResponder *) anotherResponder;
-(void)resignResponder;

-(IBAction) Trigger1: (id)sender;
-(IBAction) Trigger2: (id)sender;
-(IBAction) Trigger3: (id)sender;
@end

enum {
    State1,
    State2,
    State3};

```

Objective-C Code generated from the Example Diagram:

```

// Code generated by StateDiagramEditor (c) Copyright 2010 Stefan Sachs www.ssachs.de
// Free version, not for commercial use
// commercial use of this generated code is illegal
// (c) Copyright for template 2009 Stefan Sachs
// generated from diagram demo at 2010-03-15 08:55:13 +0100
#import "demo.h"

```

```
@implementation demo
```

```

-(void)initDiagram
{
    _actualState = State1;
    [NSApp addAction: NSSelectorFromString(@"event4:") to: nil from: self];
    [NSApp addAction: NSSelectorFromString(@"event5:") to: nil from: self];
    [NSApp addAction: NSSelectorFromString(@"event1:") to: nil from: self];
    [NSApp addAction: NSSelectorFromString(@"event2:") to: nil from: self];
}

-(void)becomeResponderAfter: (NSResponder *) anotherResponder
{
    NSResponder * nRe = [anotherResponder nextResponder];
    [anotherResponder setNextResponder: self];
    [self setNextResponder: nRe];
    precedingResponder = anotherResponder;
}

-(void)resignResponder
{
    [precedingResponder setNextResponder: [self nextResponder]];
}

-(IBAction) Trigger1: (id)sender
{
    switch(_actualState)
    {
        case State1:
            [NSApp addAction: NSSelectorFromString(@"event3:") to: nil from: self];
            [NSApp addAction: NSSelectorFromString(@"event1:") to: nil from: self];
            _actualState = State2;
            break;
        case State2:
            [NSApp addAction: NSSelectorFromString(@"event4:") to: nil from: self];
            [NSApp addAction: NSSelectorFromString(@"event5:") to: nil from: self];
            _actualState = State3;
            break;
        default:
            break;
    }
}

-(IBAction) Trigger2: (id)sender
{
    switch(_actualState)

```

```

{
    case State2:
        [NSApp sendAction: NSSelectorFromString(@"event4:") to: nil from: self];
        [NSApp sendAction: NSSelectorFromString(@"event1:") to: nil from: self];
        [NSApp sendAction: NSSelectorFromString(@"event2:") to: nil from: self];
        _actualState = State1;
        break;
    case State3:
        [NSApp sendAction: NSSelectorFromString(@"event1:") to: nil from: self];
        [NSApp sendAction: NSSelectorFromString(@"event2:") to: nil from: self];
        _actualState = State1;
        break;
    default:
        break;
}
}
-(IBAction) Trigger3: (id)sender
{
    switch(_actualState)
    {
        default:
            break;
    }
}

- (BOOL)respondToSelector:(SEL)aSelector
{
    NSString * sel = NSStringFromSelector(aSelector);
    switch(_actualState)
    {
        case State1:
            return [[NSArray arrayWithObjects: @"Trigger1:", nil] containsObject: sel];
        case State2:
            return [[NSArray arrayWithObjects: @"Trigger1:", @"Trigger2:", @"Trigger3:",
nil] containsObject: sel];
        case State3:
            return [[NSArray arrayWithObjects: @"Trigger2:", @"Trigger1:", @"Trigger3:",
nil] containsObject: sel];
        default:
            return NO;
    }
}

@end

```


Stefan Sachs
Dr. Ing.
Beratender Ingenieur

Ringreiterweg 20
23558 Lübeck
Germany

Tel. +49 (451) 8993444
Fax +49 (451) 8993445
E-Mail: ssachs@acm.org
web: www.ssachs.de