North State Framework in C#
Release Notes
V2.0

North State Software, LLC
www.northstatesoftware.com

Contents

1 Overview

Release 2.0 of the North State Framework represents the first co-release of a C# and C++ version of the framework. As we developed the new C++ framework for this release, one of our primary objectives was to keep the API as close as possible between the C# and C++ versions, so that developers could easily migrate between the two. This objective resulted in a few changes to the C# API, however, we feel the overall architecture is improved with these changes, and therefore, we were comfortable making them. These API changes are primarily in the areas of delegates, although there are also a few simplifications to constructors. Other changes help bring the framework up to UML V2.3, specifically a rename of NSFReaction to NSFInternalTransition. Finally, there are improvements to error handling, system shutdown, timer actions, and fork-join transitions, all of which make the framework easier and safer to use for mission critical applications.

2 Details

This section describes the V2.0 changes in detail.

2.1 Tagged Objects

2.1.1 NSFTaggedObject no longer supports INSFNumberedObject interface

A survey of applications revealed that NSFTaggedObject class was not using the INSFNumberedObject interface, so this support was removed. This change resulted in a significant simplification and reduction in the number of constructors supported for most framework classes.

2.2 Timer Scheduled Actions

2.2.1 NSFAction renamed to NSFScheduledAction

The class NSFAction was renamed to NSFScheduledAction to better reflect the association with the timer as a schedule action. Also, the action delegates are no longer executed on the timer thread. Rather, they are executed on a user specified event thread.

2.3 Delegates

The delegate signatures for state entry, state exit, and transitions actions have been changed to create a simplified and consistent interface between the C# and C++ versions with templatized delegates and delegate list classes. Improvements to exception handling during delegate invocation are encapsulated in the delegate list classes. These list classes handle delegate exceptions in such a way that exceptions can be reported out without interrupting the execution of other delegates in the list. Finally, transitions guard delegates have been changed to allow for multiple guards to be added to a transition.

2.3.1 NSFBoolGuard/NSFBoolGuards

The delegate type NSFBoolGuard< NSFStateMachineContext> is now used for transition guards instead of NSFGuard.

2.3.2 NSFVoidAction/NSFVoidActions

The delegate type NSFVoidAction< NSFStateMachineContext> is now used for state entry, state exit, and transition guards instead of NSFStateEventHandler.

2.4 Environment Termination

A new class, NSFEnvironment, has been added to manage termination of the environment for application shutdown. Calling the terminate() method in this class will terminate all the threads, state machines, and event handlers in a coordinated fashion.

2.5 Exception Handling

Exceptions are now handled in a consistent way across the framework. Exceptions are caught at a very low contextual level, generally at the individual delegate invocation, and are passed through a series of exception handlers through which the developer can receive notification. For example, an exception during a state entry action can provide notification through the top state machine's ExceptionActions delegate list, or through the global exception handler, NSFExecptionHandler.

2.6 Fork-Join Transition

The class NSFForkJoinTransition has been added to allow transitions between fork-joins. These transitions can optionally be associated with a region.

2.7 Internal Transitions

The class NSFReaction has been renamed to NSFInternalTransition to reflect UML 2.3 notation.

2.8 OS Abstraction Layer

Classes within the OS Abstraction layer are now prefixed with NSFOSxxx, specifically NSFOSSignal, NSFOSThread, NSFOSTimer.

3 Conclusion

We thank you for your interest in North State Software and the North State Framework. As you work with the framework, please share your experiences with us at: http://www.northstatesoftware.com.