

```
using PubSub.Implementation;

/// Defines the base configuration with actions and switches.
config Config
{
    action Publisher();
    action void Publisher.Publish(string data);
    action Subscriber(Publisher pub);
    action event void Subscriber.Received(string data);

    switch StepBound = 1024;
    switch StateBound = 1024;

    switch GeneratedTestPath = "..\TestSuite";
    switch TestClassBase = "vs";

    switch TestEnabled = false;
}

/// Defines the model program extracted from C#.
/// The exploration graph is infinite and will be pruned after a
/// step bound is hit.
machine ModelProgram() : Config where ForExploration = false
{
    construct model program from Config
}

machine JP1() : Config where ForExploration = true
{
    new Publisher; new Subscriber(_);
    (Publish("object1"); Publish("object2")) ; Received*
}

machine JP2() : Config
{
    JP1 || ModelProgram
}

/// Defines a scenario with two subscribers
machine TwoSubscribersScenario() : Config where ForExploration = true
{
    new Publisher; new Subscriber(_); new Subscriber(_);
    (Publish("object1"); Publish("object2")) ; Received*
}

/// Defines the slice with two subscribers. Parameters
/// for Publish are not determined so this not suitable for testing yet.
machine TwoSubscribersSlice() : Config
{
    TwoSubscribersScenario || ModelProgram
}

/// Defines a scenario with two subscribers
machine jp3() : Config where ForExploration = true
{
    new Publisher; new Publisher; new Subscriber(_); new Subscriber(_);
    (Publish("object1"); Publish("object2")) ; Received*
}

/// Defines the slice with two subscribers. Parameters
/// for Publish are not determined so this not suitable for testing yet.
machine jp4() : Config
{
    jp3 || ModelProgram
}
```

```
}

/// Defines a scenario where parameters are assigned
machine TwoSubscribersWithParametersScenario() : Config where ForExploration = true
{
    let string data1, string data2, string data3
        where {
            Condition.In(data1, "a1", "a2");
            Condition.In(data2, "b1", "b2");
            Condition.In(data3, "c1", "c2");
        }
    in
    new Publisher; new Subscriber(_); new Subscriber(_);
    Publish(data1); Publish(data2); Publish(data3); Received*
}

/// Defines a slice where parameters are assigned.
machine TwoSubscribersWithParametersSlice() : Config
{
    TwoSubscribersWithParametersScenario || ModelProgram
}

/// Derives a test suite. Explore this machine to see the translated and traversed test suite,
/// or generate tests from it which can be executed with VSTT. Alternatively, directly Test
/// from this machine. You can trigger a conformance failure by injecting an error in the
/// Implementation project setting PubSub.HasBug to true. Than the implementation will
/// have non-deterministic (random) failures.
machine TestSuite() : Config where TestEnabled = true
{
    construct test cases
        where strategy = "longtests", allowundeterminedcoverage = true
        for TwoSubscribersWithParametersSlice
}
}
```