Component Definition Document (CDD) for the Basic Publish-Subscribe Example Component Assembly

Rev. -

December 16, 2009

Prepared By:
Northrop Grumman Corporation
Electronic Systems
Baltimore, MD

Table of Contents

1 Introduction

1.1 Scope

This document captures the specification and design for the Basic Publish-Subscribe Example software component assembly. This example assembly is targeted for deployment on the Scalable Node Architecture (SNA) real-time component framework. As such, it must be compliant with SNA Component Based Architecture (CBA) design guidelines.

This specification defines the component assembly's functional, interface and performance requirements, the context in which it must operate, and any design constraints it must adhere to. It provides criteria for verifying compliance, but it does not state methods for achieving results.

This is intended to be a relatively informal living document, to be included in same CM repository and package as the component source code. This CDD will initially be populated by a system engineer or software architect/lead to define component design constraints & guidelines. Over time, it will transition to enhance the "to be built" specification sections with "as built" design information documenting the final component product.

2 Applicable Documents

2.1 Applicable Government Documents

Document No.	Title

2.2 Other Applicable Documents

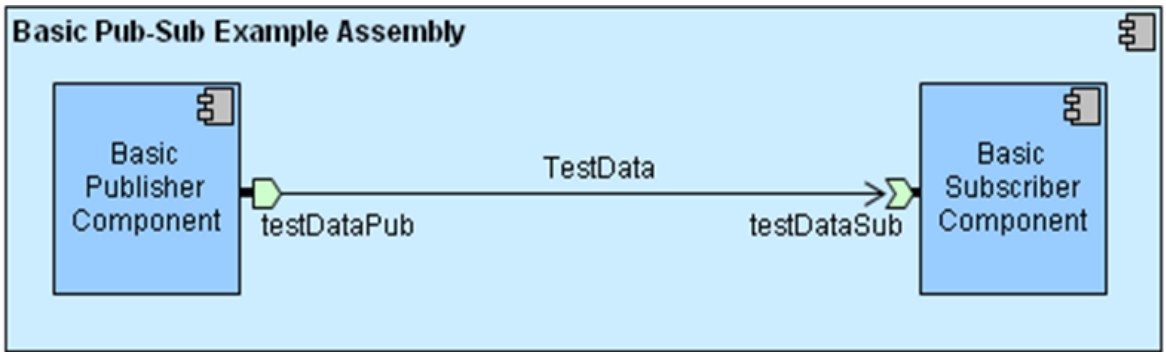
Document No.	Title

3 Component Description

3.1 Overview

The Basic Publish-Subscribe Example component assembly is one of the component source examples included in the SNA SDK for reference, testing and experimentation. It illustrates a simple component-based publish-subscribe information exchange pattern utilizing two single-port monolithic components. The Basic Publisher component periodically publishes one or more messages and the Basic Subscriber component subscribes to and receives them. The assembly containing these two components is shown in Figure 3-1 below.

Figure 3-1 – Basic Pub-Sub Example Component Assembly



3.2 Operational Context

This simple assembly is completely self contained and has no external connections. It is designed to operate solely within the constraints of the SNA SDK development environment to allow a new user/developer to step through the SNA component based development (CBD) process of loading a component assembly into the SNA IDE, building/compiling it, and then executing it.

The example is provided with an appropriate set of SNA configuration files and a deployment plan to support its execution within a single-host SNA SDK "localhost" Virtual Machine (VM). Alternative variations on the default supplied design and deployment are possible via experimentation by a software developer.

4 Component Interfaces

The Basic Publish-Subscribe Example assembly defines a single internal connection between a publisher port on the Basic Publisher monolithic component and a subscriber port on the Basic Subscriber monolithic component, as shown in Figure 3-1. The component assembly has no external interfaces.

4.1 Service Ports

There are no internal or external service ports defined for this component assembly.

4.2 Client Ports

There are no internal or external client ports defined for this component assembly.

4.3 Publisher Ports

4.3.1 TestData Publisher (internal)

The TestData publisher port on the Basic Publisher component internal to the assembly periodically publishes one or more TestData messages.

4.4 Subscriber Ports

4.4.1 TestData Subscriber (internal)

The TestData subscriber port on the Basic Subscriber component internal to the assembly receives message samples periodically published by the Basic Publisher component. The event handler "callback" registered for this port will output an SNA compliant log message for each received message.

5 Component Functionality

At startup, the Basic Publisher component sets an SNA timer (per the SNA Time Management API) to generate a timer event every 5 seconds. Upon timer expiration, the publish component's timer callback will publish a configurable number of test data sample messages, and then reset the timer. When a published sample message is received by the Basic Subscriber component, it will log a message (per the SNA Logging API) to note the event. Once started, this scenario will execute indefinitely, or until the deployment is halted manually.

6 Configurable Parameters

A BasicPublisher.cfg configuration file allows configuration of a "numSamplesToPublishEachTime" parameter, which is set to 1 by default. Changing this parameter allows more than one sample to be published each time the publication timer expires.

Since this component assembly is intended to support run-time experimentation, it is also packaged with a full set of SNA compliant run-time execution configuration and deployment files.

7 Design Constraints

1. A Basic Publisher component will periodically publish one or more (number is configurable) test data message samples.
2. A Basic Subscriber component will subscribe to the published test data messages and output a log message to indicate receipt of each published sample.
3. The Basic Publish-Subscribe Example component assembly will follow established naming conventions and code organization guidance defined in the Teton SNA SDK documentation, such that new SNA software developers can use it as a design reference.
4. A full set of SNA compliant configuration and deployment files will be provided with this example in order to support run-time execution in a default single-host target deployment.
5. This example will utilize the standard SNA APIs to perform all functions.

8 Component Test

This 2-component assembly must be executable on a single "localhost" development computer, to include the SNA SDK x86-64 VM at a minimum. Users can experiment with the deployment plan to redeploy to an alternative 2-host target environment if desired.

9 Component Dependencies

The Basic Publish-Subscribe Example assembly is self contained and has no dependencies on any other application components. Its only dependency is on the Teton SNA SDK's run time execution environment.