Software System Test Plan

**Messenger**

**Client – Server Application**

**Simple Message Protocol (SMP)**

**Version 1**



# Table of Contents

[Table of Contents 2](#_Toc130981561)

[Revision History 4](#_Toc130981562)

[1 Introduction 5](#_Toc130981563)

[2 Documentation 5](#_Toc130981564)

[3 References 5](#_Toc130981565)

[3.1 Reference 1: *SMP\_Software\_Project\_Information\_File.docx* 5](#_Toc130981566)

[3.1.1 Description 5](#_Toc130981567)

[3.2 Reference 2: *SMP\_Software\_Project\_Plan.docx* 5](#_Toc130981568)

[3.2.1 Description 5](#_Toc130981569)

[3.3 Reference 3: *SMP\_Software\_Design\_Specification.docx* 5](#_Toc130981570)

[3.3.1 Description 5](#_Toc130981571)

[3.4 Reference 4: *SMP\_Software\_Requirements.docx* 5](#_Toc130981572)

[3.4.1 Description 5](#_Toc130981573)

[3.5 Reference 5: *SMP\_Software\_Technical\_Specification.docx* 6](#_Toc130981574)

[3.5.1 Description 6](#_Toc130981575)

[4 Test System Configurations 6](#_Toc130981576)

[4.1 Configuration 1: One *Local Area Network (LAN)* 6](#_Toc130981577)

[4.1.1 Description 6](#_Toc130981578)

[4.1.2 Diagram -- SMP Client Producer Sends SMP PUT Request 6](#_Toc130981579)

[4.1.3 Diagram -- SMP Client Consumer Sends SMP GET Request 7](#_Toc130981580)

[4.2 Configuration 2: Two *Local Area Networks* 7](#_Toc130981581)

[4.2.1 Description 7](#_Toc130981582)

[4.2.2 Diagram -- SMP Client Producer Sends SMP PUT Request 7](#_Toc130981583)

[4.2.3 Diagram -- SMP Client Consumer Sends SMP GET Request 8](#_Toc130981584)

[5 Configuration Files 9](#_Toc130981585)

[5.1 Configuration File 1 Name 9](#_Toc130981586)

[5.1.1 Description 9](#_Toc130981587)

[5.2 Configuration File 2 Name 9](#_Toc130981588)

[5.2.1 Description 9](#_Toc130981589)

[6 Test Automation Suite 9](#_Toc130981590)

[7 Test Cases – SMP Server Requirements 9](#_Toc130981591)

[7.1 Test Case – Application Start Feature (Not Automated) 9](#_Toc130981592)

[7.1.1 Target Requirement: *Requirement 1: Application Start Feature* 9](#_Toc130981593)

[7.1.2 Test Case Description 9](#_Toc130981594)

[7.1.3 Test Case Steps 9](#_Toc130981595)

[7.2 Test Case – Show Messages Feature (Not Automated) 10](#_Toc130981596)

[7.2.1 Target Requirement: *Requirement 2: Show Messages Feature* 10](#_Toc130981597)

[7.2.2 Test Case Description 10](#_Toc130981598)

[7.2.3 Test Case Steps 10](#_Toc130981599)

[8 Test Cases – SMP Client Producer Requirements 10](#_Toc130981600)

[8.1 Test Case – Send Message Feature (Not Automated) 10](#_Toc130981601)

[8.1.1 Target Requirement: *Requirement 1: Send Message Feature* 10](#_Toc130981602)

[8.1.2 Test Case Description 10](#_Toc130981603)

[8.1.3 Test Case Steps 10](#_Toc130981604)

[8.2 Test Case – SMP Server Offline (Not Automated) 11](#_Toc130981605)

[8.2.1 Target Requirement: *TBD* 11](#_Toc130981606)

[8.2.2 Test Case Description 11](#_Toc130981607)

[8.2.3 Test Case Steps 11](#_Toc130981608)

[9 Test Cases – SMP Client Consumer Requirements 11](#_Toc130981609)

[9.1 Test Case – Get Message Feature (Not Automated) 11](#_Toc130981610)

[9.1.1 Target Requirement: *Requirement 1: Get Message Feature* 11](#_Toc130981611)

[9.1.2 Test Case Description 11](#_Toc130981612)

[9.1.3 Test Case Steps 12](#_Toc130981613)

[9.2 Test Case – No Priority One Messages (Not Automated) 12](#_Toc130981614)

[9.2.1 Target Requirement: *Requirement 1: Get Message Feature* 12](#_Toc130981615)

[9.2.2 Test Case Description 12](#_Toc130981616)

[9.2.3 Test Case Steps 12](#_Toc130981617)

[9.3 Test Case – SMP Server Offline (Not Automated) 13](#_Toc130981618)

[9.3.1 Target Requirement: *TBD* 13](#_Toc130981619)

[9.3.2 Test Case Description 13](#_Toc130981620)

[9.3.3 Test Case Steps 13](#_Toc130981621)

[10 Appendix 13](#_Toc130981622)

[10.1 Acronyms 13](#_Toc130981623)

[10.2 Engineering Terms 13](#_Toc130981624)

[10.2.1 Software Development 13](#_Toc130981625)

[11 Notes 13](#_Toc130981626)

[12 System Test Schedule 14](#_Toc130981627)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Date | Authors | Comments |
| 1 |  |  | 1. Initial release. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

This document is intended to be used for verifying the Messenger Simple Message Protocol (SMP) Client – Server software application prior to release. The primary goal is to verify that all expected software application features have been implemented.

# Documentation

For each version of released code, this document is to be saved in a location specified in the Software Project Information File (SPIF) with the following information:

Software Application Name: Messenger Client – Server Application

Software Application Version: Version 1

Software Application Repository: GitHub: https://github.com/NCSS/Messenger

Name of tester:

System Test Date:

# References

## Reference 1: *SMP\_Software\_Project\_Information\_File.docx*

### Description

The software project information file.

## Reference 2: *SMP\_Software\_Project\_Plan.docx*

### Description

The software project plan.

## Reference 4: *SMP\_Software\_Project\_Interface\_Prototype.docx*

### Description

The prototype of the UI.

## Reference 4: *SMP\_Software\_Requirements.docx*

### Description

The software requirement document.

## Reference 5: *SMP\_Software\_Technical\_Specification.docx*

### Description

The software technical specification.

## Reference 3: *SMP\_Software\_Design\_Specification.docx*

### Description

The software design specification.

# Test System Configurations

## Configuration 1: One *Local Area Network (LAN)*

### Description

This configuration requires that the SMP Server, SMP Client Producer, and the SMP Client Consumer are all connected to the same LAN.

### Diagram -- SMP Client Producer Sends SMP PUT Request



SMP Client Producer sends an SMP PUT Request packet to the SMP Server

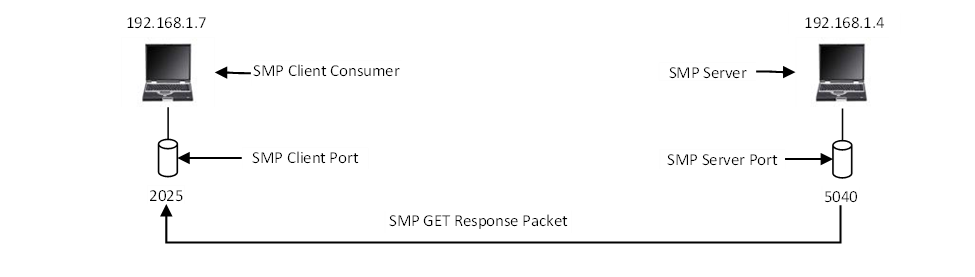


SMP Server sends back an SMP PUT Response Packet

### Diagram -- SMP Client Consumer Sends SMP GET Request



SMP Client Consumer sends an SMP GET Request packet to the SMP Server



SMP Server sends back an SMP GET Response Packet

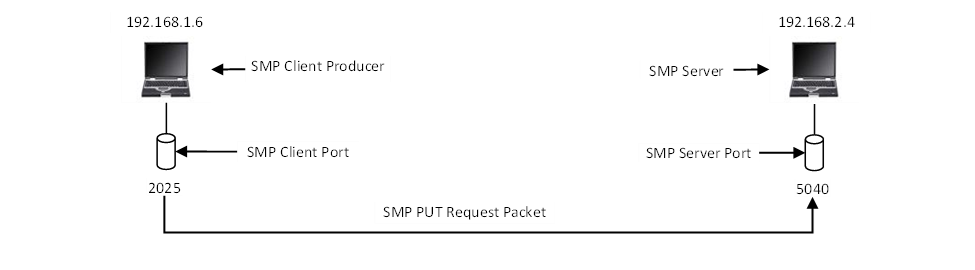
Figure 1: LAN System Test Configuration

## Configuration 2: Two *Local Area Networks*

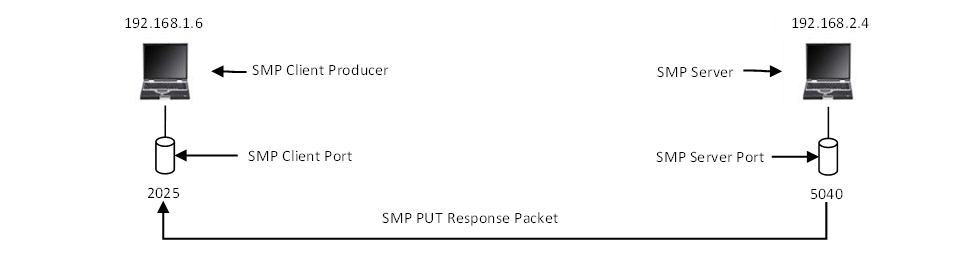
### Description

This configuration requires that the SMP Server is connected to a separate LAN, and the SMP Client Producer, and the SMP Client Consumer are connected to a different LAN.

### Diagram -- SMP Client Producer Sends SMP PUT Request

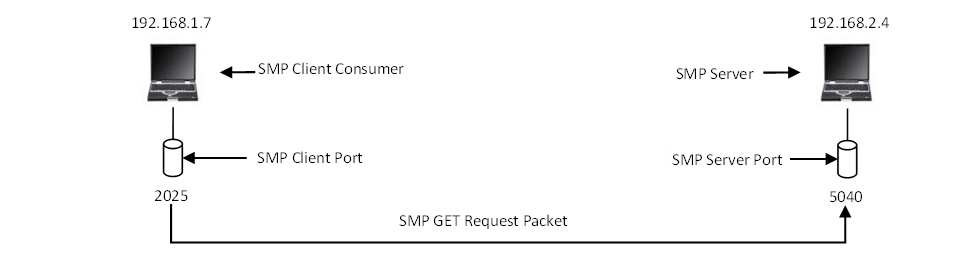


SMP Client Producer sends an SMP PUT Request packet to the SMP Server



SMP Server sends back an SMP PUT Response Packet

### Diagram -- SMP Client Consumer Sends SMP GET Request



SMP Client Consumer sends an SMP GET Request packet to the SMP Server



SMP Server sends back an SMP GET Response Packet

Figure 2: Two Separate LANs System Test Configuration

# Configuration Files

A test case may require a particular software system configuration file. The required configuration files are listed here.

## Configuration File 1 Name

### Description

TBD

## Configuration File 2 Name

### Description

TBD

# Test Automation Suite

Test cases that have been automated are marked as Automated. The test automation suite must run the automated test cases before each software application version release. The test automation suite and test case source code are saved in a location specified in the Software Project Information File (SPIF).

# Test Cases – SMP Server Requirements

## Test Case – Application Start Feature (Not Automated)

### Target Requirement: *Requirement 1:* *Application Start Feature*

Notes:

### Test Case Description

Description: This feature allows the SMP server to start processing SMP client requests. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button, in the case of a graphical user interface (GUI).

### Test Case Steps

* Start the SMP Server application.
* Click the Start button.
* Verify the server is listening to client requests on port 50400.
  + netstat -na | find 50400

## Test Case – Show Messages Feature (Not Automated)

### Target Requirement: *Requirement 2: Show Messages Feature*

Notes:

### Test Case Description

Description: Reads SMP message records from the SMP message file and displays the message date, message priority, and message content for each of the messages in the file and outputs the messages. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).

### Test Case Steps

* Start the SMP Server application.
* Click the Show Message button.
* Verify the client messages are displayed in the Messages window.

## Test Case – Show Messages Feature No Priority (Not Automated)

### Target Requirement: *Requirement 2: Show Messages Feature*

Notes:

### Test Case Description

Description: Reads SMP message records from the SMP message file and displays the message date, message priority, and message content for each of the messages in the file and outputs the messages. When the show messages button is clicked the SMP Server should retrieve the corresponding messages from the SMP message file. If no priority is checked the server should default to “all”.

### Test Case Steps

* Start the SMP Server application.
* Click the low priority radio button on the SMP Consumer application
* Write a test message on the SMP Consumer application
* Click the send message button
* Click the medium priority radio button on the SMP Consumer application
* Write a test message on the SMP Consumer application
* Click the send message button
* Do **Not** check any priority in the priority radio buttons on the SMP Server application
* Click the SMP Server show messages button.
* Verify that the SMP Server shows all messages from all priorities

# Test Cases – SMP Client Producer Requirements

## Test Case – Send Message Feature (Not Automated)

### Target Requirement: *Requirement 1:* *Send Message Feature*

Notes:

### Test Case Description

Description: The SMP Message Producer client program is designed to send an SMP PUT request to the server. This feature allows a user to send an SMP message to the server. The message sent to the server consists of the message date, message priority and the message content. The server adds the record to a file associated with the message priority. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).

### Test Case Steps

* Start the SMP Server application.
* Start the SMP Client Producer application.
* Enter the IP address of the SMP Server application.
* Enter the port number of the SMP Server application.
* From the Message Priority options, select Low.
* Enter Test Message Low in the Message Content text entry box.
* Click the Send Message button.
* Verify the SMP Server sent back a response.
  + Note the SMP Server response in the Server Response text box.
* Verify the SMP Server received the message.
  + Verify the Message Type text box displays PutMessage.
  + Verify the Message Priority text box displays Low.

## Test Case – SMP Server Offline (Not Automated)

### Target Requirement: *TBD*

Notes:

### Test Case Description

Description: The SMP Message Producer client program is designed to send an SMP PUT request to the server. If the SMP Server is offline, when a SMP PUT request is sent to the server, the SMP Message Producer should display a Message Box notifying the user.

### Test Case Steps

* Verify the SMP Server is OFFLINE.
* Start the SMP Client Producer application.
* Enter the IP address of the SMP Server application.
* Enter the port number of the SMP Server application.
* From the Message Priority options, select Low.
* Enter Test Message Low in the Message Content text entry box.
* Click the Send Message button.
* Verify the SMP Client Producer displays a Message Box notifying the user that the SMP PUT request failed.

## Test Case – Send Empty Message (Not Automated)

### Target Requirement: *TBD*

Notes:

### Test Case Description

Description: The SMP Message Producer client program is designed to always have a message when it sends. When the SMP Message Producer client’s send button is pressed if there is no message to send the SMP Server shouldn’t receive anything.

### Test Case Steps

* Verify the SMP Server is ONLINE
* Start the SMP Server
* Click the Low priority radio button on the SMP Producer
* Do **not** write any message in the message box
* Click send message button
* Click the Medium Priority radio button on the SMP Producer
* Do **not** write any message in the message box
* Click send message button
* Click the High Priority radio button on the SMP Producer
* Do **not** write any message in the message box
* Click send message button
* Click the “all” priority radio button on the SMP Server
* Click the get messages button on the SMP Server to verify no message has been received

# Test Cases – SMP Client Consumer Requirements

## Test Case – Get Message Feature (Not Automated)

### Target Requirement: *Requirement 1: Get Message Feature*

Notes:

### Test Case Description

Description: The SMP Message Consumer client program is designed to send an SMP GET request to the server to retrieve the next message. This feature allows a user to retrieve an SMP message from the server. Which message is sent back from the server depends on the priority selected. The message sent back to the client consists of the message date, message priority and the message content. The server deletes the record from the file after the client acknowledges that it has received the message. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).

### Test Case Steps

* Start the SMP Server application.
* Start the SMP Client Consumer application.
* Enter the IP address of the SMP Server application.
* Enter the port number of the SMP Server application.
* From the Message Priority options, select Low.
* Click the Get Message button.
* Verify the SMP Server sent back a response.
  + The message content should be displayed in the Message Content text box.
* Verify the SMP Server
  + Received the acknowledgement message from the SMP Client Consumer.
  + Verify the Message Type text box displays GetMessage.
  + Verify the Message Priority text box displays Low.
  + Verify the message is deleted from the Low Priority Messages file.

## Test Case – No Priority One Messages (Not Automated)

### Target Requirement: *Requirement 1: Get Message Feature*

Notes:

### Test Case Description

Description: The SMP Message Consumer client program is designed to send an SMP GET request to the server to retrieve the next message.

This feature allows a user to retrieve an SMP message from the server. Which message is sent back from the server depends on the priority selected. The message sent back to the client consists of the message date, message priority and the message content. The server deletes the record from the file after the client acknowledges that it has received the message. The UI for this feature can be a command-line option in the case of a command-line UI (CLUI), or a clickable button in the case of a graphical user interface (GUI).

### Test Case Steps

* Start the SMP Server application.
* Start the SMP Client Consumer application.
* Enter the IP address of the SMP Server application.
* Enter the port number of the SMP Server application.
* From the Message Priority options, select Low.
* Click the Get Message button.
* Verify the SMP Server sends back the following response: No Priority Low Messages Available
* Verify the SMP Client Consumer displays a Message Box notifying the user that No Priority Low Messages Available.

## Test Case – SMP Server Offline (Not Automated)

### Target Requirement: *TBD*

Notes:

### Test Case Description

Description: The SMP Message Consumer client program is designed to send an SMP GET request to the server to retrieve the next message. If the SMP Server is offline, when a SMP GET request is sent to the server, the SMP Message Consumer should display a Message Box notifying the user.

### Test Case Steps

* Verify the SMP Server is OFFLINE.
* Start the SMP Client Consumer application.
* Enter the IP address of the SMP Server application.
* Enter the port number of the SMP Server application.
* From the Message Priority options, select Low.
* Click the Get Message button.
* Verify the SMP Client Consumer displays a Message Box notifying the user that the SMP GET request failed.

## Test Case – SMP Server Offline (Not Automated)

### Target Requirement: *TBD*

Notes:

### Test Case Description

Description: The SMP Message Consumer client program is designed to send an SMP GET request to the server to retrieve the next message. If there are multiple messages of the same priority the SMP Consumer application should retrieve them in chronological order.

### Test Case Steps

* Verify the SMP Server is ONLINE.
* Click the low priority radio button on the SMP Consumer application
* Write "test message 1” on the SMP Consumer application
* Click the send message button
* Click the low priority radio button on the SMP Consumer application
* Write "test message 2" on the SMP Consumer application
* Click the send message button
* Click the low priority radio button on the SMP Client application
* Click the get message button
* Verify message text is “test message 1”
* Click the low priority radio button on the SMP Client application
* Click the get message button
* Verify message text is “test message 2”

# Appendix

## Acronyms

SMP – Simple Message Protocol

## Engineering Terms

### Software Development

Client – A software application that sends and receives messages to and from a server application.

Server – A software application that processes messages from client applications.

Producer – A software application that produces messages.

Consumer – A software application that consumes messages.

# Notes

# System Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| System Test Date | Hours | Team Member | Comments |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |