



## S@T Compliance Test Suite High Level Test Specification

### 1 History

Document history		
Release	Approved by	Comment
1.0.4	TDG WG3	1 <sup>st</sup> official version.

# TABLE OF CONTENTS

<b>1</b>	<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>2</b>	<b>TERMINOLOGY</b> .....	<b>8</b>
2.1.1	Notation .....	8
2.1.2	Abbreviations.....	8
<b>3</b>	<b>LIST OF REFERENCE DOCUMENTS</b> .....	<b>8</b>
<b>4</b>	<b>OVERVIEW</b> .....	<b>9</b>
<b>5</b>	<b>OVERALL ARCHITECTURE</b> .....	<b>9</b>
<b>6</b>	<b>TEST STRATEGY</b> .....	<b>11</b>
<b>7</b>	<b>BROWSER TESTS</b> .....	<b>11</b>
7.1	SSP/3.48/3.40 TESTS .....	11
7.2	SBC ACCEPTANCE TESTS .....	11
7.3	ADMINISTRATIVE COMMANDS TESTS .....	11
7.4	ENVIRONMENT VARIABLES TESTS .....	11
7.5	VARIABLE TEST.....	11
<b>8</b>	<b>SSP tests</b> .....	<b>12</b>
<b>8.1</b>	<b>Connection command</b> .....	<b>12</b>
8.1.1	Description .....	12
8.1.2	Specification references.....	12
8.1.3	Normal test .....	12
8.1.4	Error test .....	12
<b>8.2</b>	<b>Disconnection command</b> .....	<b>13</b>
8.2.1	Description .....	13
8.2.2	Specification references.....	13
8.2.3	Normal test .....	13
8.2.4	Error test .....	13
<b>8.3</b>	<b>Data command</b> .....	<b>13</b>
8.3.1	Description .....	13
8.3.2	Specification references.....	14
8.3.3	Normal test .....	14
8.3.4	Error test.....	14
<b>8.4</b>	<b>Get command</b> .....	<b>14</b>
8.4.1	Description .....	14
8.4.2	Specification references.....	14
8.4.3	Normal test .....	15
8.4.4	Error test .....	15
<b>8.5</b>	<b>Post command</b> .....	<b>15</b>
8.5.1	Description .....	15
8.5.2	Specification references.....	15
8.5.3	Normal test .....	15
8.5.4	Error test .....	16
<b>8.6</b>	<b>Express data command</b> .....	<b>16</b>
8.6.1	Description .....	16
8.6.2	Specification references.....	16
8.6.3	Normal test .....	16

8.6.4	Error test .....	16
<b>8.7</b>	<b>Pause command .....</b>	<b>16</b>
8.7.1	Description .....	16
8.7.2	Specification references.....	17
8.7.3	Normal test .....	17
8.7.4	Error test .....	17
<b>8.8</b>	<b>Resume command.....</b>	<b>17</b>
8.8.1	Description .....	17
8.8.2	Specification references.....	17
8.8.3	Normal test .....	17
8.8.4	Error test .....	17
<b>8.9</b>	<b>Unknown command .....</b>	<b>17</b>
8.9.1	Description .....	17
8.9.2	Specification references.....	18
8.9.3	Normal test .....	18
8.9.4	Error test .....	18
<b>8.10</b>	<b>Serial command.....</b>	<b>18</b>
8.10.1	Description .....	18
8.10.2	Specification references.....	18
8.10.3	Normal test .....	18
8.10.4	Error test .....	18
<b>8.11</b>	<b>Cancelling actions .....</b>	<b>19</b>
8.11.1	Description .....	19
8.11.2	Specification references.....	19
8.11.3	Normal test .....	19
8.11.4	Error test .....	19
<b>8.12</b>	<b>Simultaneous requests .....</b>	<b>19</b>
8.12.1	Description .....	19
8.12.2	Specification references.....	19
8.12.3	Normal test .....	19
8.12.4	Error test .....	20
<b>9</b>	<b>03.40 / 03.48.....</b>	<b>20</b>
<b>9.1</b>	<b>03.40 Browser stopped .....</b>	<b>20</b>
9.1.1	Description .....	20
9.1.2	Specification references.....	20
9.1.3	Normal test .....	20
9.1.4	Error test .....	20
<b>9.2</b>	<b>03.40 Browser launched .....</b>	<b>21</b>
9.2.1	Description .....	21
9.2.2	Specification references.....	21
9.2.3	Normal test .....	21
9.2.4	Error test .....	21
<b>9.3</b>	<b>03.48 TAR Tests.....</b>	<b>22</b>
9.3.1	Description .....	22
9.3.2	Specification references.....	22
9.3.3	Normal test .....	22
9.3.4	Error test .....	22
<b>9.4</b>	<b>03.48 : Test of the Security Parameters on a single SMS.....</b>	<b>22</b>
9.4.1	Description .....	22
9.4.2	Specification references.....	22
9.4.3	Normal test .....	22
9.4.4	Error test .....	23
<b>9.5</b>	<b>03.48 : Test of the Security Parameters on concatenated SMS.....</b>	<b>23</b>

9.5.1	Description .....	23
9.5.2	Specification references.....	23
9.5.3	Normal test .....	23
9.5.4	Error test .....	23
<b>10</b>	<b><i>Administrative commands tests .....</i></b>	<b>24</b>
<b>10.1</b>	<b>RequestId Management.....</b>	<b>24</b>
10.1.1	Description .....	24
10.1.2	Specification references.....	24
10.1.3	Normal test .....	24
10.1.4	Error test .....	24
<b>10.2</b>	<b>Install/Uninstall deck .....</b>	<b>25</b>
10.2.1	Description .....	25
10.2.2	Specification references.....	25
10.2.3	Normal test .....	25
10.2.4	Error test.....	25
<b>10.3</b>	<b>SPS.....</b>	<b>26</b>
10.3.1	Description .....	26
10.3.2	Specification references.....	26
10.3.3	Normal test .....	26
10.3.4	Error test .....	26
10.3.5	Other test .....	27
<b>10.4</b>	<b>Environment variables .....</b>	<b>28</b>
10.4.1	Description .....	28
10.4.2	Specification references.....	28
10.4.3	Normal test .....	28
10.4.4	Error test .....	28
<b>10.5</b>	<b>Contextual menus.....</b>	<b>29</b>
10.5.1	Description .....	29
10.5.2	Specification references.....	29
10.5.3	Normal test .....	29
10.5.4	Error test .....	29
<b>11</b>	<b><i>Operational commands tests.....</i></b>	<b>31</b>
<b>11.1</b>	<b>PULL.....</b>	<b>31</b>
11.1.1	Description .....	31
11.1.2	Specification references.....	31
11.1.3	Normal test .....	31
11.1.4	Error test .....	31
<b>11.2</b>	<b>Bookmarks : indexed implementation .....</b>	<b>31</b>
11.2.1	Description .....	31
11.2.2	Specification references.....	32
11.2.3	Normal test .....	32
11.2.4	Error test .....	32
<b>11.3</b>	<b>Bookmarks : URL implementation .....</b>	<b>32</b>
11.3.1	Description .....	32
11.3.2	Specification references.....	32
11.3.3	Normal test .....	32
11.3.4	Error test .....	32
<b>12</b>	<b><i>Miscellaneous tests.....</i></b>	<b>33</b>
<b>12.1</b>	<b>Start-up Procedure .....</b>	<b>33</b>
12.1.1	Description .....	33
12.1.2	Specification references.....	33
12.1.3	Normal tests.....	33

12.1.4	Error tests.....	33
<b>12.2</b>	<b>Basic local navigation.....</b>	<b>33</b>
12.2.1	Description .....	33
12.2.2	Specification references.....	33
12.2.3	Normal tests.....	34
12.2.4	Error tests.....	34
<b>12.3</b>	<b>Contextual menus.....</b>	<b>34</b>
12.3.1	Description .....	34
12.3.2	Specification references.....	34
12.3.3	Normal tests.....	34
12.3.4	Error tests.....	34
<b>12.4</b>	<b>Special coding : UCS2.....</b>	<b>35</b>
12.4.1	Description .....	35
12.4.2	Specification references.....	35
12.4.3	Normal tests.....	35
12.4.4	Error tests.....	35
<b>12.5</b>	<b>Telephony events.....</b>	<b>35</b>
12.5.1	Description .....	35
12.5.2	Specification references.....	35
12.5.3	Normal tests.....	35
12.5.4	Error tests.....	36
<b>12.6</b>	<b>General tests .....</b>	<b>36</b>
12.6.1	Description .....	36
12.6.2	Specification references.....	36
12.6.3	Normal tests.....	36
12.6.4	Error tests.....	36
<b>13</b>	<b>SBC Tests.....</b>	<b>36</b>
<b>13.1</b>	<b>Concatenate tag.....</b>	<b>36</b>
13.1.1	Description .....	36
13.1.2	Specification references.....	36
13.1.3	Normal test .....	37
13.1.4	Error test .....	37
<b>13.2</b>	<b>Extract tag .....</b>	<b>38</b>
13.2.1	Description .....	38
13.2.2	Specification references.....	38
13.2.3	Normal test .....	38
13.2.4	Error test.....	38
<b>13.3</b>	<b>SetHelp tag.....</b>	<b>38</b>
13.3.1	Description .....	38
13.3.2	Specification references.....	39
13.3.3	Normal test .....	39
13.3.4	Error test .....	39
<b>13.4</b>	<b>GetEnv tag.....</b>	<b>40</b>
13.4.1	Description .....	40
13.4.2	Specification references.....	40
13.4.3	Normal test .....	40
13.4.4	Error test .....	40
<b>13.5</b>	<b>InitVarSelected tag .....</b>	<b>41</b>
13.5.1	Description .....	41
13.5.2	Specification references.....	41
13.5.3	Normal test .....	41
13.5.4	Error test .....	41
<b>13.6</b>	<b>InitVar tag .....</b>	<b>42</b>

13.6.1	Description .....	42
13.6.2	Specification references.....	42
13.6.3	Normal test .....	42
13.6.4	Error test .....	42
<b>13.7</b>	<b>Encryption .....</b>	<b>42</b>
13.7.1	Description .....	42
13.7.2	Specification references.....	42
13.7.3	Normal test .....	43
13.7.4	Error test .....	43
<b>13.8</b>	<b>Decryption .....</b>	<b>43</b>
13.8.1	Description .....	43
13.8.2	Specification references.....	43
13.8.3	Normal test .....	44
13.8.4	Error test .....	44
<b>13.9</b>	<b>Go Back.....</b>	<b>44</b>
13.9.1	Description .....	44
13.9.2	Specification references.....	44
13.9.3	Normal test .....	44
13.9.4	Error test .....	45
<b>13.10</b>	<b>Go Selected .....</b>	<b>45</b>
13.10.1	Description.....	45
13.10.2	Specification references .....	45
13.10.3	Normal test.....	45
13.10.4	Error test .....	46
<b>13.11</b>	<b>Switch case on variable.....</b>	<b>46</b>
13.11.1	Description.....	46
13.11.2	Specification references .....	46
13.11.3	Normal test.....	46
13.11.4	Error test .....	47
<b>13.12</b>	<b>Terminate Browser Session.....</b>	<b>47</b>
13.12.1	Description.....	47
13.12.2	Specification references .....	47
13.12.3	Normal test.....	47
<b>13.13</b>	<b>Manage Contextual Menu Item .....</b>	<b>47</b>
13.13.1	Description.....	47
13.13.2	Specification references .....	48
13.13.3	Normal test.....	48
13.13.4	Error test .....	48
<b>13.14</b>	<b>STK Generic Macro.....</b>	<b>48</b>
13.14.1	Description.....	48
13.14.2	Specification references .....	48
13.14.3	Normal test.....	49
13.14.4	Error test .....	49
<b>13.15</b>	<b>Execute.....</b>	<b>49</b>
13.15.1	Description.....	49
13.15.2	Specification references .....	50
13.15.3	Normal test.....	50
13.15.4	Error test .....	50
<b>14</b>	<b><i>GATEWAY TESTS [Not implemented]</i> .....</b>	<b>50</b>
<b>14.1</b>	<b>SSP/3.48/3.40 TESTS .....</b>	<b>50</b>
<b>14.2</b>	<b>SBC GENERATION TESTS .....</b>	<b>51</b>
<b>14.3</b>	<b>ADMINISTRATIVE COMMANDS TESTS.....</b>	<b>51</b>

14.4	<a href="#">S@TML</a> ACCEPTANCE TESTS .....	51
14.5	GATEWAY REQUESTS PROCESSING TESTS.....	51
15	<i>ANNEX : DETAILED TESTS DOCUMENTS</i> .....	51
15.1	Detail test sheets .....	51
15.2	Test decks descriptions .....	52
15.3	Test decks bytecode and dump files .....	52

# 1 TERMINOLOGY

## 1.1.1 Notation

Lexical and syntactical specifications are given in EBNF (extended Backus Naur Form), with literals enclosed in single quotes 'xyz' or given in a single character set like [ 0-9 ] for a digit, and using the operators (...) (precedence), ? (optional), \* (zero or more times), + (one or more times), | (alternative), and "... ::= ... ." for rules.

## 1.1.2 Abbreviations

<b>HTTP</b>	Hyper Text Transfer Protocol
<b>S@T</b>	SIM Alliance Toolbox
<b>SBC</b>	S@T Byte Code
<b>SSP</b>	S@T Session Protocol
<b>S@TML</b>	S@T Markup Language
<b>STK</b>	SIM Application Toolkit
<b>STLS</b>	S@T Transport Layer Security
<b>TLV</b>	Tag Length Value encoding
<b>URL</b>	Unified Resource Locator
<b>SSP</b>	Service Permanent Store

# 2 LIST OF REFERENCE DOCUMENTS

/SBC/	S@T 01.00 V1.0.3: SBC – S@TML Byte Code
/SSP/	S@T 01.20 V1.0.3: SSP –S@T Session Protocol
/ADM/	S@T 01.21 V1.0.3: ADM –S@T Administrative Commands
/OPE/	S@T 01.22 V1.0.3: SSP –S@T Operational Commands



### 3 OVERVIEW

This document intends to describe the high level system architecture of a [S@T](#) system and to propose a test strategy and required means in aim to test a [S@T](#) Gateway or a [S@T](#) browser against inter-operability.

The high level test description is included in that document.

A detailed description of the tests and its architecture is described in the last chapter.

### 4 OVERALL ARCHITECTURE

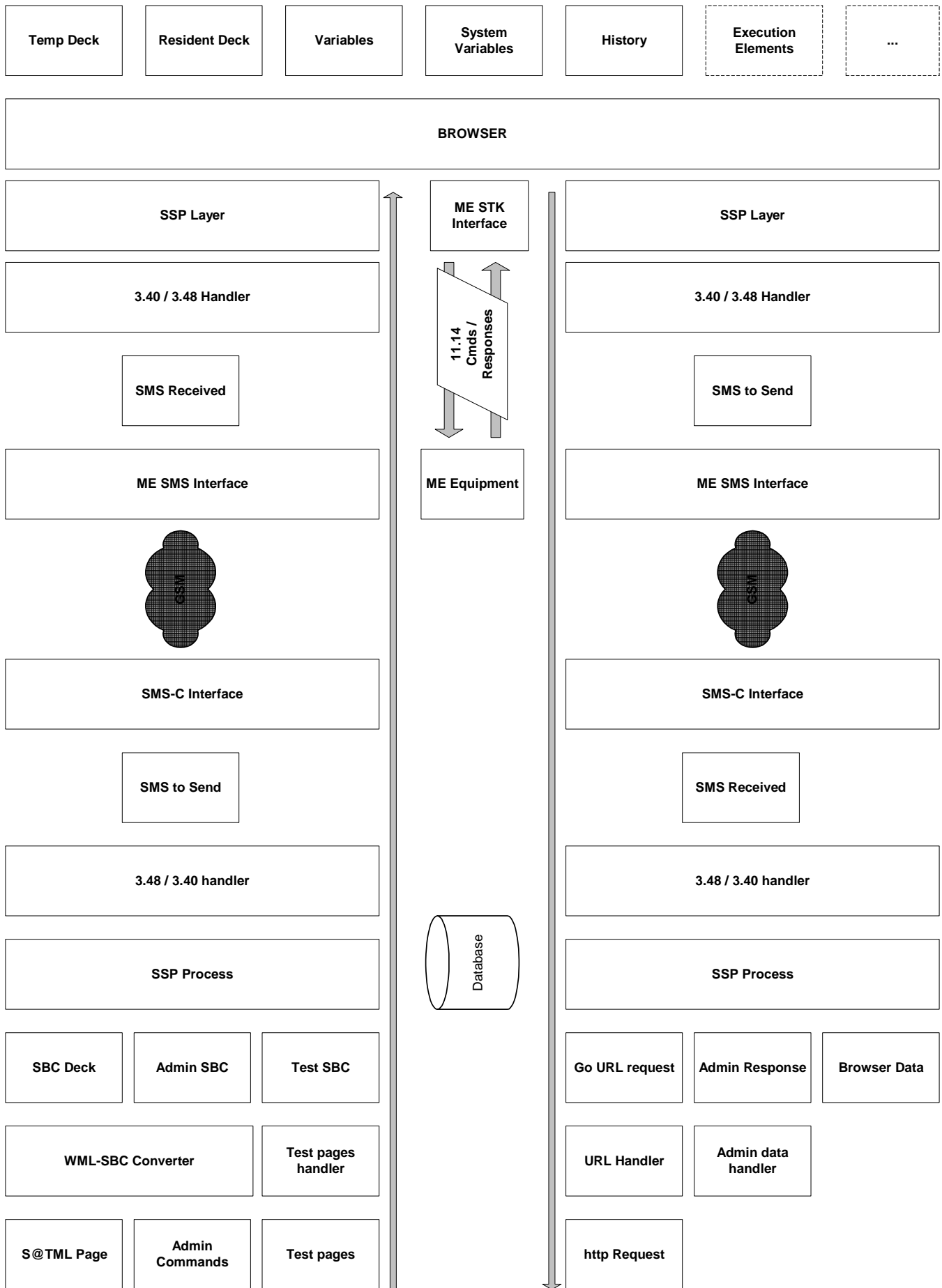
The [S@T](#) system is composed of several entities :

- The [S@TML](#) pages sources.
- The [S@T](#) Gateway generating SBC.
- The SMS transport mean including SMS-C
- The Mobile Equipment (ME)
- The SIM Card
- The [S@T](#) Microbrowser embedded in the SIM.

The specifications of [S@T](#) cover the [S@TML](#) pages format, the SMS transport layer called SSP, the format of the encoded pages called SBC and the Browser guideline.

As the [S@T](#) system can be provided by different vendors for the [S@T](#) gateway or the [S@T](#) microbrowser, inter-operability tests are defined.

Here is the detailed architecture of the overall system :



The interfaces that exist between subsystems and that need to be inter-operable are :

- SSP/3.48/3.40 SMS compatibility in both ways.
- SBC generated by the gateway(s) acceptable by the browser(s)
- Request handling of SSP well transformed in http: requests
- S@TML sources

## 5 TEST STRATEGY

The test suite aims to avoid the crash or abnormal behavior of the [S@T](#) browsers and gateways during the execution and administration of services.

In this first release, the focus has been put on the interoperability of Browsers implementation against one gateway.

The testing scope has been decided to be limited to the SIM+browser implementation. The test suite will check the browser itself against a defined test suite.

It manages the error codes, and exercises them to identify the browser or the gateway responses to that errors.

Objective is to have the majority of the test suite automated, leading to some interface inclusion in the tested gateway or browser.

The aim of that test suite is to be available widely across [Sim@lliance](#) members and that each company is able to run the tests on its own. Actual cross tests will be performed at later stage.

The body that will pronounce the final certification can be an operator during the progress of a project or an independent consultant assigned by SIM @lliance.

In a future version, the following scopes can be added :

- Check that the defined set of [S@TML](#) pages is well processed by the gateway.
- Output of the Gateway for compliance with SBC and protocol stack (SSP/3.48)

## 6 BROWSER TESTS

### 6.1 SSP/3.48/3.40 TESTS

Test of 3.48 layer for security level, check of fields...

For SSP, the check of the session management, time out, state machine analysis is done

### 6.2 SBC ACCEPTANCE TESTS

Each Byte code is tested in that test suite in all its configurations.

### 6.3 ADMINISTRATIVE COMMANDS TESTS

Check the execution or rejection of the administrative commands.

Analyse the result of that admin commands.

### 6.4 ENVIRONMENT VARIABLES TESTS

Set environment variables and check their management.

### 6.5 VARIABLE TEST

Typically check the usage of the variables, the substitution, the reset, the allocation.

## 7 SSP tests

### 7.1 Connection command

#### 7.1.1 Description

This chapter describes the tests used to validate the reception and the sending of connection command.

#### 7.1.2 Specification references

The specifications are written in:

- /SSP/
- /ADM/
- /OPE/

Tested points:

- Correct handling of connection command parameters : connect command tag, connection ID, server address, application address

#### 7.1.3 Normal test

Reference	Description	Component Name	Shared
1	Check the connection request tag during a connection request from the browser	Core	Y
2	Test the protocol ID during a connection request from the browser	Core	Y
3	Test the connection ID during a connection request from the browser	Core	Y
4	Test the server address during a connection request from the browser	Core	Y
5	Test the application address during a connection request from the browser	Core	Y
6	Test the protocol ID during a connection request from a gateway	Core	Y
7	Test the server address during a connection request from a gateway	Core	Y
8	Test the application address during a connection request from a gateway	Core	Y
9	Check the value of the connection response tag during a connection response from the browser	Core	Y
10	Check the connection ID during a connection response from the browser	Core	Y
11	Test the Session ID during a connection response from the browser	Core	Y
12	Test the Status/Cause during a connection response from the browser	Core	N
13	Test the Status/Cause during a connection response from the browser Multi-session not supported.		
14	Test the connection ID during a connection response from the gateway	Core	Y
15	Test the Session ID during a connection response from the gateway	Core	Y
16	Test the Status/Cause during a connection response from the gateway	Core	Y

#### 7.1.4 Error test

Reference	Description	Component Name	Shared
17	Test a bad protocol ID during a connection request from a gateway	Core	N
18	Test the browser when the gateway replies with a bad session ID	Core	Y
19	Test the browser when the gateway replies with a bad connection ID	Core	Y
20	Test the browser when the gateway replies with an unknown	Core	Y

	Status/Cause		
--	--------------	--	--

## 7.2 Disconnection command

### 7.2.1 Description

This chapter describes the tests used to validate the reception and the sending of disconnection command.

### 7.2.2 Specification references

The specifications are written in:

- /SSP/

Tested points:

- Correct handling of disconnection command parameters : disconnect command tag, session ID, cause/value

### 7.2.3 Normal test

Reference	Description	Component Name	Shared
1	Test the session ID during a disconnection request from the gateway in browsing mode	Core	Y
2	Test the Cause Value during a disconnection request from the gateway in browsing mode	Core	Y
3	Test the session ID during the reception of a disconnect request from the gateway in administrative mode	Core	Y
4	Test the Cause Value during the reception of a disconnect request from the gateway in administrative mode	Core	Y

### 7.2.4 Error test

Reference	Description	Component Name	Shared
5	Test the behaviour of the browser when the gateway sends a disconnect request without the current \$SessionID	Core	Y
6	Test the behaviour of the browser during a disconnection request from the gateway in browsing mode with an unknown cause value	Core	Y
7	Test the behaviour of the browser during the reception of a disconnect request from the gateway in administrative mode without the current session ID	Core	N
8	Test behaviour of the browser during the reception of a disconnect request from the gateway in administrative mode with an unknown cause value	Core	Y

## 7.3 Data command

### 7.3.1 Description

This chapter describes the tests used to validate the reception and the sending of data command.

### 7.3.2 Specification references

The specifications are written in:

- /SSP/
- /ADM/

Tested points:

- Correct handling of data command parameters : data command tag, connection ID, server address, application address

### 7.3.3 Normal test

Reference	Description	Component Name	Shared
1	Test the session ID during a data request from the gateway	Core	Y
2	Test the Transaction ID during a data request from the gateway	Core	Y
3	Test the TPS during a data request from the gateway	Core	Y
4	Check the data response tag during a data response from the browser	Core	Y
5	Test the session ID during a data response from the browser	Core	Y
6	Test the Transaction ID during a data response from the browser	Core	Y
7	Test the TPS during a data response from the browser	Core	Y
8	Check the data request tag during a data request from the browser	Bookmark	Y
9	Test the session ID during a data request from the browser	Bookmark	Y
10	Test the Transaction ID during a data request from the browser	Bookmark	Y
11	Test the TPS during a data request from the browser	Bookmark	Y
12	Test the session ID during a reply on a browser data request	Bookmark	Y
13	Test the transaction ID during a reply on a browser data request	Bookmark	Y
14	Test the TPS during a reply on a browser data request	Bookmark	Y

### 7.3.4 Error test

Reference	Description	Component Name	Shared
15	Test the behaviour of the browser during a data request from the gateway with a bad \$SessionID	Core	Y
16	Test the behaviour of the browser during a data request from the gateway with a bad \$TPS	Core	Y
17	Test the behaviour of the browser during a data response from the gateway with a bad \$SessionID	Bookmark	Y
18	Test the behaviour of the browser during a data response from the gateway with a bad \$TransactionID	Bookmark	Y
19	Test the behaviour of the browser during a data response from the gateway with a bad \$TPS	Bookmark	Y

## 7.4 Get command

### 7.4.1 Description

This chapter describes the tests used to validate the reception and the sending of get command.

### 7.4.2 Specification references

The specifications are written in:

- /SSP/
- /OPE/

Tested points:

- Correct handling of get command parameters : get command tag, session ID, transaction ID, TPS

### 7.4.3 Normal test

Reference	Description	Component Name	Shared
1	Check the get request tag during a get request from the browser	Core	Y
2	Test the session ID during a get request from the browser	Core	Y
3	Test the Transaction ID during a get request from the browser	Core	Y
4	Test the TPS during a get request from the browser	Core	Y
5	Test the session ID during a reply on a browser get request	Core	Y
6	Test the Transaction ID during a reply on a browser get request	Core	Y
7	Test the TPS during a reply on a browser get request	Core	Y

### 7.4.4 Error test

Reference	Description	Component Name	Shared
8	Test the behaviour of the browser during a reply on a browser get request with a bad session ID	Core	Y
9	Test the behaviour of the browser during a reply on a browser get request with a bad transaction ID	Core	Y
10	Test the behaviour of the browser during a reply on a browser get request with a bad TPS	Core	Y

## 7.5 Post command

### 7.5.1 Description

This chapter describes the tests used to validate the reception and the sending of post command.

### 7.5.2 Specification references

The specifications are written in:

- /SSP/
- /OPE/

Tested points:

- Correct handling of post command parameters : post command tag, session ID, transaction ID, TPS

### 7.5.3 Normal test

Reference	Description	Component Name	Shared
1	Check the post request tag during a post request from the browser	Core	Y
2	Test the session ID during a post request from the browser	Core	Y
3	Test the Transaction ID during a post request from the browser	Core	Y
4	Test the TPS during a post request from the browser	Core	Y
5	Test the session ID during the reception of a reply on a browser post	Core	Y

	request		
6	Test the Transaction ID during the reception of a reply on a browser post request	Core	Y
7	Test the TPS during the reception of a reply on a browser post request	Core	Y

#### 7.5.4 Error test

Reference	Description	Component Name	Shared
8	Test the behaviour of the browser during the reception of a reply on a browser post request with a bad session ID	Core	Y
9	Test the behaviour of the browser during the reception of a reply on a browser post request with a bad transaction ID	Core	Y

## 7.6 Express data command

### 7.6.1 Description

This chapter describes the tests used to validate the reception and the sending of express data command.

### 7.6.2 Specification references

The specifications are written in:

- /SSP/
- /ADM/
- /OPE/

Tested points:

- Correct handling of express data command parameters : express data command tag, session ID, TPS

### 7.6.3 Normal test

Reference	Description	Component Name	Shared
1	Check the express data request tag during an express data request from the browser	Core	Y
2	Test the session ID during an express data request from the browser	Core	Y
3	Test the TPS during an express data request from the browser	Core	Y
4	Test the session ID during the reception of an express data request from the gateway	Core	Y
5	Test the TPS during the reception of an express data request from the gateway	Core	Y

### 7.6.4 Error test

None.

## 7.7 Pause command

### 7.7.1 Description

This chapter describes the tests used to validate the sending of pause data command.



## 7.7.2 Specification references

The specifications are written in:

- /SSP/

Tested points:

- Correct handling of pause command parameters : pause command tag, session ID

## 7.7.3 Normal test

Reference	Description	Component Name	Shared
1	Check the pause request tag during a pause request from the browser	Resume / Pause	Y
2	Check the session ID during a pause request from the browser	Resume / Pause	Y
3	Test the state of the browser after a pause request	Resume / Pause	Y

## 7.7.4 Error test

None.

## 7.8 Resume command

### 7.8.1 Description

This chapter describes the tests used to validate the sending of resume data command.

### 7.8.2 Specification references

The specifications are written in:

- /SSP/

Tested points:

- Correct handling of resume command parameters : resume command tag, session ID

### 7.8.3 Normal test

Reference	Description	Component Name	Shared
1	Check the resume request tag during a resume request from the browser	Resume / Pause	Y
2	Check the session ID during a resume request from the browser	Resume / Pause	Y
3	Test the state of the browser after a resume request	Resume / Pause	Y

### 7.8.4 Error test

None.

## 7.9 Unknown command

### 7.9.1 Description

This chapter describes the tests used to validate the reception of unknown command.

## 7.9.2 Specification references

The specifications are written in:

- /BRW/

Tested points:

- Correct handling of unknown command tag

## 7.9.3 Normal test

None.

## 7.9.4 Error test

Reference	Description	Component Name	Shared
1	Test the browser tag in idle mode with an unknown tag	Core	Y
2	Test the browser when connected in browsing mode with an unknown tag	Core	Y
3	Test the browser when connected in administrative mode with an unknown tag	Core	Y

## 7.10 Serial command

### 7.10.1 Description

This chapter describes the tests used to validate the reception of several commands in one or different messages.

### 7.10.2 Specification references

The specifications are written in:

- /SSP/
- /ADM/
- /OPE/

Tested points:

- Correct handling of the express data request after a connection request;
- Correct handling of different commands in one or several messages;

### 7.10.3 Normal test

Reference	Description	Component Name	Shared
1	Check that the browser sends always a connection request followed by an express data request	Core	Y
2	Check that the browser can handle a connection request followed by an express data request	Core	Y
3	Check that the browser can serially handle different commands sent in different messages	Core	Y
4	Check that the browser can serially handle different commands sent in a single message	Core	Y

### 7.10.4 Error test

Reference	Description	Component Name	Shared
5	Test the browser when there are several commands and some of them are invalids	Core	Y

## 7.11 Cancelling actions

### 7.11.1 Description

This chapter describes the tests used to validate the behaviour of the browser when actions are cancelled by the user.

### 7.11.2 Specification references

The specifications are written in:

- /SSP/
- /BRW/

Tested points:

- Correct handling of the express data request after a connection request;
- Correct handling of a request after having cancelled the same request;

### 7.11.3 Normal test

Reference	Description	Component Name	Shared
1	Cancelling a get request from the browser. Then retrying it.	Core	N
2	Cancelling a post request from the browser. Then retrying it.	Core	N
3	Cancelling a message which contains a connection request and another request. Then sending again the last request;	Core	N
4	Cancelling a message which contains a connection request and another request. Without waiting the answer, retrying the same requests with;	Core	N

### 7.11.4 Error test

None.

## 7.12 Simultaneous requests

### 7.12.1 Description

This chapter describes the tests used to validate the behaviour of the browser when the gateway and the browser send simultaneously a request.

### 7.12.2 Specification references

The specifications are written in:

- /SSP/

Tested points:

- Correct handling of the simultaneous connect, post, get or pause commands;

### 7.12.3 Normal test

Reference	Description	Component Name	Shared
1	Connection request from the gateway when the browser is trying to open a browsing session;	Core	Y
2	Connection request from the browser when the gateway is trying to open an administrative session;	Core	Y
3	When the browser sends a get data request whereas the gateway sends a disconnect request, check that the browser handles the disconnect request and doesn't wait for its request response;	Core	Y
4	When the browser sends a post data request whereas the gateway sends a disconnect request, check that the browser handles the disconnect request and doesn't wait for its request response;	Core	Y
5	When the browser sends a pause request whereas the gateway sends a disconnect request, check that the browser doesn't handle the disconnect request;	Resume / Pause	Y

#### 7.12.4 Error test

None.

## 8 03.40 / 03.48

### 8.1 03.40 Browser stopped

#### 8.1.1 Description

This chapter describes the tests used to validate the reception of SMS according to 03.40 parameters when the browser is not active. To be compliant with all the SMS-C, a [S@T-Browser](#) should be able to handle SMS P-P envelope and SMS P-P update EF SMS.

#### 8.1.2 Specification references

The specifications are written in:

- [S@T](#) 01.20 v1.0.3 SSP.

Tested points:

- Correct handling of 03.40 parameters (TP-PID, concatenation)

#### 8.1.3 Normal test

Reference	Description	Component Name	Shared
1	Test reception of one SMS P-P envelope by the browser	Core	Y
2	Test reception of one SMS Update EF SMS	Core	Y
3	Test reception of concatenated SMS P-P envelope by the browser	Core	Y
4	Test reception of concatenated SMS P-P update EF SMS by the browser	Core	Y

#### 8.1.4 Error test

Reference	Description	Component Name	Shared
5	Test reception of concatenated SMS P-P envelope by the browser. SMS send twice.	Core	Y

6	Test reception of concatenated SMS P-P envelope by the browser. Bad ConcatRef.	Core	Y
7	Test reception of concatenated SMS P-P envelope by the browser. MaxNumber lower than the number of SMS send.	Core	Y
8	Test reception of concatenated SMS P-P update EF SMS by the browser. SMS send twice.	Core	Y
9	Test reception of concatenated SMS P-P update EF SMS by the browser. Bad ConcatRef.	Core	Y
10	Test reception of concatenated SMS P-P update EF SMS by the browser. MaxNumber lower than the number of SMS send.	Core	Y
11	Test reception of concatenated SMS P-P update EF SMS by the browser. Mix of two packet.	Core	Y

## 8.2 03.40 Browser launched

### 8.2.1 Description

This chapter describes the tests used to validate the reception of SMS according to 03.40 parameters when the browser is launched. To be compliant with all the SMS-C, a [S@T-Browser](#) should be able to handle SMS P-P envelope and SMS P-P update EF SMS.

### 8.2.2 Specification references

The specifications are written in:

- [S@T 01.20 v1.0.3 SSP](#).

Tested points:

- Correct handling of 03.40 parameters (TP-PID, concatenation)

### 8.2.3 Normal test

Reference	Description	Component Name	Shared
1	Test reception of one SMS P-P envelope by the browser	Core	Y
2	Test reception of one SMS Update EF SMS	Core	Y
3	Test reception of concatenated SMS P-P envelope by the browser	Core	Y
4	Test reception of concatenated SMS P-P update EF SMS by the browser	Core	Y

### 8.2.4 Error test

Reference	Description	Component Name	Shared
5	Test reception of concatenated SMS P-P envelope by the browser. SMS send twice.	Core	Y
6	Test reception of concatenated SMS P-P envelope by the browser. Bad ConcatRef.	Core	Y
7	Test reception of concatenated SMS P-P envelope by the browser. MaxNumber lower than the number of SMS send.	Core	Y
8	Test reception of concatenated SMS P-P envelope by the browser. Mix of two packets.	Core	Y
9	Test reception of concatenated SMS P-P update EF SMS by the browser. SMS send twice.	Core	Y
10	Test reception of concatenated SMS P-P update EF SMS by the browser. Bad ConcatRef.	Core	Y
11	Test reception of concatenated SMS P-P update EF SMS by the browser. MaxNumber lower than the number of SMS send.	Core	Y

12	Test reception of concatenated SMS P-P update EF SMS by the browser. Mix of two packet.	Core	Y
----	--	------	---

## 8.3 03.48 TAR Tests

### 8.3.1 Description

This chapter describes the tests used to validate the management of the TAR value in the 03.48 security header.

### 8.3.2 Specification references

The specifications are written in:

- [S@T](#) 01.20 v1.0.3 SSP.

Tested points:

- Correct handling of 03.48 parameters : TAR

### 8.3.3 Normal test

Reference	Description	Component Name	Shared
1	Reception of a single SMS with the right TAR	Core	Y
2	Reception of a packet with the right TAR	Core	Y

### 8.3.4 Error test

Reference	Description	Component Name	Shared
3	Reception of a single SMS with a wrong TAR	Core	Y
4	Reception of a packet with a wrong TAR	Core	Y

## 8.4 03.48 : Test of the Security Parameters on a single SMS

### 8.4.1 Description

This chapter describes the tests used to validate the management of the level of the security to use during SMS exchange for single SMS.

### 8.4.2 Specification references

The specifications are written in:

- [S@T](#) 01.20 v1.0.3 SSP.

Tested points:

- Correct handling of 03.48 parameters in a single SMS : SPI, Kic, Kid, Counter

### 8.4.3 Normal test

Reference	Description	Component Name	Shared
1	Sending of one SMS on operational mode, no security should be used.	Core	Y
2	Reception of one SMS on operational mode, no security should be used.	Core	Y
3	Sending of one SMS on administrative mode	Core	Y

4	Reception of one SMS on administrative mode using the minimum security required (DES in CBC mode)	Core	Y
5	Reception of one SMS on administrative mode using 3-DES in CBC mode	Core	Y

#### 8.4.4 Error test

Reference	Description	Component Name	Shared
6	Reception of one SMS on administrative mode with no security applied	Core	Y
7	Reception of one SMS on administrative mode with a bad DES CC	Core	Y
8	Reception of one SMS on administrative mode with a bad 3-DES CC	Core	Y
9	Reception of one SMS on administrative mode with a bad counter	Core	Y
10	Reception of one SMS on administrative mode with a SPI less than the administrative level	Core	Y

### 8.5 03.48 : Test of the Security Parameters on concatenated SMS

#### 8.5.1 Description

This chapter describes the tests used to validate the management of the level of the security to use during SMS exchange for concatenated SMS.

#### 8.5.2 Specification references

The specifications are written in:

- [S@T 01.20 v1.0.3 SSP](#).

Tested points:

- Correct handling of 03.48 parameters in concatenated SMS : SPI, Kic, Kid, Counter

#### 8.5.3 Normal test

Reference	Description	Component Name	Shared
1	Reception of concatenated SMS on administrative mode using the minimum security required (DES in CBC mode)	Core	Y
2	Reception of concatenated SMS on administrative mode using 3-DES in CBC mode	Core	Y

#### 8.5.4 Error test

Reference	Description	Component Name	Shared
3	Reception of concatenated SMS on administrative mode with no security applied	Core	Y
4	Reception of concatenated SMS on administrative mode with a bad DES CC	Core	Y
5	Reception of concatenated SMS on administrative mode with a bad 3-DES CC	Core	Y
6	Reception of concatenated SMS on administrative mode with a bad counter	Core	Y
7	Reception of concatenated SMS on administrative mode with a SPI less	Core	Y

	than administrative level		
--	---------------------------	--	--

## 9 Administrative commands tests

Mandatory tests: 52,

Optional tests: 24.

### 9.1 RequestId Management

#### 9.1.1 Description

This chapter describes the tests used to validate the requestId management in the administrative commands.

#### 9.1.2 Specification references

The specifications are written in:

- /ADM/.

Tested points:

- Correct management of RequestId field.

#### 9.1.3 Normal test

Reference	Description	Component Name	Shared
1	Check the RequestId management in administrative commands.	Core	Yes

#### 9.1.4 Error test

None.



## 9.2 Install/Uninstall deck

### 9.2.1 Description

This chapter describes the tests used to validate the install and uninstall deck administrative commands.

### 9.2.2 Specification references

The specifications are written in:

- /ADM/.

Tested points:

- The messages.

### 9.2.3 Normal test

Reference	Description	Component Name	Shared
1	Test the installation of a small deck ( $L \leq 128$ )	Core	Yes
2	Test the installation of a medium deck ( $128 < L < 256$ )	Core	Yes
3	Test the installation of a long deck ( $255 < L < 1024$ )	Core	Yes
4	Update a deck with Old deck size > new deck size.	Core	Yes
5	Update a deck with Old deck size < new deck size.	Core	Yes
6	Update a deck with Old deck size = new deck size.	Core	Yes
7	Test the installation of a deck depending on the value of the field ResidentDeckBufferSize, and of the field ReceptionBufferSize MS1 = memory size of deck1 (deck to be installed) MSH = memory size of the header 03.48, SSP, Admin for the install admin command for the deck1 $MS1 < \text{Min}(\text{ResidentDeckBufferSize}, (\text{ReceptionBufferSize} - \text{MSH}))$	Core	Yes
8	Test the installation of a deck depending on the value of the field ResidentDeckBufferSize, and of the field ReceptionBufferSize MS1 = memory size of deck1 (deck to be installed) MSH = memory size of the header 03.48, SSP, Admin for the install admin command for the deck1 $MS1 = \text{Min}(\text{ResidentDeckBufferSize}, (\text{ReceptionBufferSize} - \text{MSH}))$	Core	Yes
10	Uninstall a resident deck	Core	Yes
12	Uninstall the home Page	Core	Yes

### 9.2.4 Error test

Reference	Description	Component name	Shared
9	Test the installation of a deck depending on the value of the field ResidentDeckBufferSize, and of the field ReceptionBufferSize MS1 = memory size of deck1 (deck to be installed) MSH = memory size of the header 03.48, SSP, Admin for the install admin command for the deck1 $MS1 > \text{Min}(\text{ResidentDeckBufferSize}, (\text{ReceptionBufferSize} - \text{MSH}))$	Core	No
11	Uninstall a not present deck	Core	Yes

## 9.3 SPS

### 9.3.1 Description

This chapter describes the tests used to validate the SPS related administrative commands. SPS implementation is optional.

### 9.3.2 Specification references

The specifications are written in:

- /ADM/.

Tested points:

- SPS ALLOCATE message,
- ALLOCATE PERMANENT VARIABLE message,
- SET PERMANENT VARIABLE message,
- GET PERMANENT VARIABLE message.

### 9.3.3 Normal test

Reference	Description	Component Name	Shared
3	Reduce the memory allocated to a service which contains Permanent Variables with FreeServiceVariables attribute unset : the memory allocated is still sufficient for the Permanent variables of the service	SPS	No
5	Reduce the memory allocated to a service which contains Permanent Variables with FreeServiceVariables attribute set : the memory allocated is insufficient for the Permanent variables of the service	SPS	No
6	Free an allocated service which contains Permanent Variables with FreeServiceVariables attribute set	SPS	No
7	Extend the memory allocated to a service which contains Permanent Variables	SPS	No
9	Allocate a permanent variable in an existing service which possess enough space	SPS	No
10	Free a permanent variable which exist in an existing service	SPS	No
15	Set a permanent variable which exist with enough space allocated	SPS	No
19	Get a permanent variable which exist	SPS	No
23	Free an allocated service which does not contain Permanent Variables	SPS	No

### 9.3.4 Error test

Reference	Description	Component Name	Shared
1	Allocation of services in the SPS until we reach the maximum size of the SPS	SPS	No
2	Free an allocated service which contains Permanent Variables with FreeServiceVariables attribute unset	SPS	No
4	Reduce the memory allocated to a service which contains Permanent Variables with FreeServiceVariables attribute unset : the memory allocated is insufficient for the Permanent variables of the service	SPS	No
8	Free a not existing service	SPS	No

11	Allocate a permanent variable in an existing service which do not possess enough space	SPS	No
12	Free a permanent variable which do not exist in an existing service	SPS	No
13	Free a permanent variable in a not existing service	SPS	No
14	Allocate a permanent variable in a not existing service	SPS	No
16	Set a permanent variable which exist with not enough space allocated	SPS	No
17	Set a permanent variable which do not exist	SPS	No
18	Set a permanent variable in a service which do not exist	SPS	No
20	Get a permanent variable which do not exist	SPS	No
21	Get a permanent variable in a service which do not exist	SPS	No

#### 9.3.5 Other test

Reference	Description	Component Name	Shared
22	Extend and reduce an allocated and instanciated permanent variable	SPS	No

## 9.4 Environment variables

### 9.4.1 Description

This chapter describes the tests used to validate the environment variables management related administrative commands.

### 9.4.2 Specification references

The specifications are written in:

- /ADM/,
- /SBC/.

Tested points:

- Admin Set Env,
- Admin Get Env.

### 9.4.3 Normal test

Reference	Description	Component Name	Shared
2	Set the Gateway address which is read/write environment variable. Environment variable allowed value are listed in the parameter value section	Core	Yes
6	Get the XXX where XXX is an environment variable. Environment variable allowed value are listed in the parameter value section	Core	Yes

### 9.4.4 Error test

Reference	Description	Component Name	Shared
1	Set the XXX where XXX is an environment variable which is read only. Environment variable allowed value are listed in the parameter value section	Core	Yes
3	Set an undefined environment variable	Core	Yes
4	Set the Gateway address with a 4 bytes long value	Core	Yes
5	Set the Gateway address with a 2 bytes long value	Core	Yes
7	Get an undefined environment variable	Core	Yes

## 9.5 Contextual menus

### 9.5.1 Description

This chapter describes the tests used to validate the contextual menus management specifications.

### 9.5.2 Specification references

The specifications are written in:

- /ADM/,
- /SBC/.

Tested points:

- Default visibility set menu item,
- System menu item related commands,
- Application menu item related commands.

### 9.5.3 Normal test

Reference	Description	Component Name	Shared
1	Check that the default visibility specified menu item are present	Core	No
2	Show the system menu items of the "Back Menu"	Core	Yes
5	Hide the system menu items of the "Back Menu"	Core	No
6	Hide the system menu items of the "Help Menu"	Core	No
8	Add an application menu item in the "Back Menu"	Core	No
9	Add an application menu item in the "Help Menu"	Core	No
10	Add an application menu item in the "Abort Menu"	Core	No
11	Update an application menu item in the "Back Menu"	Core	No
12	Update an application menu item in the "Help Menu"	Core	No
13	Update an application menu item in the "Abort Menu"	Core	No
14	Remove an application menu item in the "Back Menu"	Core	No
15	Remove an application menu item in the "Help Menu"	Core	No
16	Remove an application menu item in the "Abort Menu"	Core	No

### 9.5.4 Error test

Reference	Description	Component Name	Shared
3	Send a command to show an undefined system menu	Core	No
4	Send a command to show a system menu Item on an undefined menu	Core	Yes
7	Hide the defined system menu items of the "Abort Menu"	Core	No
17	Add application menu item until a memory allocation problem occur or no id still available	Core	No
18	Send a command to add an application menu Item on an undefined menu	Core	Yes
19	Send an administrative command to add a menu application item, with the "operator bit" not set	Core	Yes
20	Send an administrative command to remove a menu application item, with the "operator bit" not set	Core	Yes
21	Send an administrative command to show a menu system item, with the "operator bit" not set	Core	Yes
22	Send an administrative command to hide a menu system item, with the "operator bit" not set	Core	Yes
23	Send an administrative command to add an application menu item, with the URL field not set	Core	Yes
24	Send an administrative command to add an application menu item, with the text field not set	Core	Yes

25	Send an administrative command to add an application menu item, without any field set	Core	Yes
26	Send an administrative command to remove an application menu item, with data fields set	Core	Yes
27	Send an administrative command to add a system menu item (kind of update command), with data fields set	Core	Yes

## 10 Operational commands tests

Mandatory tests: 15,

Optional tests: 12.

### 10.1 PULL

#### 10.1.1 Description

This chapter describes the tests used to validate the PULL operational commands.

#### 10.1.2 Specification references

The specifications are written in:

- /OPE/.

Tested points:

- When the browser requests a deck from the gateway, it uses a GET\_REQ or a POST\_REQ message depending on the HTTP request to perform.,
- "SendReferer" bit,
- A REPLY\_RSP message contains in its value part the deck TL[A]V described in /SBC/,
- ReceptionBufferSize,
- "GetBookmarkedDeck" bit.

#### 10.1.3 Normal test

Reference	Description	Component Name	Shared
1	Ask for a deck which URL reference has it "SendReferer" bit set	Core	Yes
2	Install a small temporary deck (L<=128)	Core	Yes
3	Install a medium temporary deck (128<L<256)	Core	Yes
4	Install a long temporary deck (255<L<1024)	Core	Yes
7	Test the correct use of URL reference : Adress is defined by a value	Core	Yes
8	Test the correct use of URL reference : Adress is defined by a reference	Core	Yes
9	Test the correct use of URL reference : Parameters are defined by a value	Core	Yes
10	Test the correct use of URL reference : Parameters are defined by reference	Core	Yes

#### 10.1.4 Error test

Reference	Description	Component Name	Shared
5	Send a REPLY_RSP with an undefined operational tag	Core	Yes
6	Send a REPLY_RSP with a bookmark operational tag	BKM	No
11	Test the correct use of URL reference : The adress is defined with a reference to a variable which is not initialized	Core	No
12	Test the correct use of URL reference : A parameter is defined with a reference to a variable which is not initialized	Core	Yes

## 10.2 Bookmarks : indexed implementation

### 10.2.1 Description

This chapter described the test used to validate the indexed implementation of bookmarks.

### 10.2.2 Specification references

The specifications are written in:

- /OPE/.

Tested points:

- Browser request,,
- Gateway response.

### 10.2.3 Normal test

Reference	Description	Component Name	Shared
1	Check the browser request management, and the gateway response management	BKM	No

### 10.2.4 Error test

Reference	Description	Component Name	Shared
2	Answer with the response normally used for URL implemented bookmark	BKM	No
3	Send a DATA_RSP with a wrong byte code	BKM	No
4	Send a DATA_RSP with a 0xFF index value	BKM	No

## 10.3 Bookmarks : URL implementation

### 10.3.1 Description

This chapter described the test used to validate the URL implementation of bookmarks.

### 10.3.2 Specification references

The specifications are written in:

- /OPE/.

Tested points:

- Browser request,,
- Gateway response.

### 10.3.3 Normal test

Reference	Description	Component Name	Shared
1	Check the browser request management, and the gateway response management	BKM	No

### 10.3.4 Error test

Reference	Description	Component Name	Shared
2	Answer with the response normally used for index implemented bookmark	BKM	No
3	Send a DATA_RSP with a wrong byte code	BKM	No



# 11 Miscellaneous tests

## 11.1 Start-up Procedure

### 11.1.1 Description

This chapter describes the tests used to validate the startup procedure of the browser.

### 11.1.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- browser launched directly or y an external application
- express data at first connection
- browser behavior if ucs2 is not supported

### 11.1.3 Normal tests

Reference	Description	Component Name	Shared
3	Terminal profile and UCS2 supported (input and output )	Core	Yes
4	History and temporary variable content verification at browser startup	Core	No
5	Launch broswer from an external application and return to this application (tag Exit) without Exit list element	Core	No
6	Launch broswer from an external application and return to this application (tag Exit) with Exit list element	Core	No
7	Express data command at first connection	Core	Yes

### 11.1.4 Error tests

Reference	Description	Component Name	Shared
1	Terminal profile with non supported proactive commands (select item)	Core	Yes
2	Terminal profile and UCS2 not supported (input / output )	Core	Yes

## 11.2 Basic local navigation

### 11.2.1 Description

This chapter describes the tests used to validate the management of variable in case of local navigation with Go selected macro.

### 11.2.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- local navigation with variable management
- card attribute ResetVar
- multiples cards with same Id

### 11.2.3 Normal tests

Reference	Description	Component Name	Shared
1	Intra-deck navigation with go selected and impact on temporary variable management	Core	Yes
2	Inter-deck navigation with go selected and impact on temporary variable management	Core	Yes
3	Navigation to a specific card with go selected element and impact on temporary variable management	Core	Yes
4	Intra-deck navigation with go direct and impact on temporary variable management	Core	Yes
5	Inter-deck navigation with go direct and impact on temporary variable management	Core	Yes
6	Navigation to a specific card with go direct and impact on temporary variable management	Core	Yes
7	Browse a deck with multiple cards with the same identifier	Core	Yes

### 11.2.4 Error tests

None

## 11.3 Contextual menus

### 11.3.1 Description

This chapter describes the tests used to validate historization and variable management in case of local navigation with Mcmi macro.

### 11.3.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Local navigation with Mcmi macro
- Card attribute ResetVar and ChainNextCard
- Empty card

### 11.3.3 Normal tests

Reference	Description	Component Name	Shared
1	Interactive inter-card navigation and historization (1 level)	Core	No
2	Historization and card attribute management	Core	No
3	Historization and impact on temporary variable management	Core	No
4	Inter-Card navigation and card attribute ChainNextCard Management	Core	No
5	Inter-Card navigation with last card with attribute ChainNextCard set	Core	No
6	Empty card with attribute ChainNextCard cleared	Core	No

### 11.3.4 Error tests

None

## 11.4 Special coding : UCS2

### 11.4.1 Description

This chapter describes the tests used to validate management of DCS.

### 11.4.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- ucs2 and variable management
- ucs2 in go selected and init variable selected
- ucs2 and concatenation
- ucs2 and extract macro

### 11.4.3 Normal tests

Reference	Description	Component Name	Shared
1	Initialization and display of variable with DCS	Core	Yes
2	Management of variable with multiple DCS	Core	Yes
4	Couple with inline value or variable ref in UCS2	Core	Yes
5	Go selected with item coded in UCS2	Core	Yes
6	Go selected with title coded in UCS2	Core	Yes
7	Extract a substing from a variable (temporary variable or text element) with a DCS UCS2	Core	Yes

### 11.4.4 Error tests

Reference	Description	Component Name	Shared
3	Concatenation of variable with different DCS	Core	Yes

## 11.5 Telephony events

### 11.5.1 Description

This chapter describes the tests used to validate browser in case of telephony events during browsing session.

### 11.5.2 Specification references

NA

### 11.5.3 Normal tests

Reference	Description	Component Name	Shared
1	Reception of a text SMS for another application during the browsing session	Core	Yes
2	Use of the setupcall command in a browsing session	Core	Yes
3	Sent of a text SMS during a browsing session	Core	Yes

## 11.5.4 Error tests

None

## 11.6 General tests

### 11.6.1 Description

This chapter describes the tests used to validate management of length and attributes

### 11.6.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- two attribute bytes
- length of macro lower than 127
- length of macro higher than 127 and lower than 256

### 11.6.3 Normal tests

Reference	Description	Component Name	Shared
1	Two bytes coded length at different levels (deck, card, card content and couple)	Core	Yes
2	Three bytes coded length at different levels (deck, card, card content and couple)	Core	Yes
3	Single Card attribute byte with all bits cleared	Core	Yes
4	One byte attribute with unspecified flag set	Core	Yes
5	Two bytes attribute with unspecified flag set	Core	Yes

### 11.6.4 Error tests

None

## 12 SBC Tests

### 12.1 Concatenate tag

#### 12.1.1 Description

This chapter describes the tests used to validate the correct processing of the concatenate tag by the browser.

#### 12.1.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.4

### 12.1.3 Normal test

Reference	Description	Component Name	Shared
1	Concatenate with several VarRefTags and several InValElements	Core	Y
2	Concatenate with TextElement and temporary variable	Core	Y
3	Concatenate with VarRef and empty Inline value	Core	Y
4	Concatenate with InlineValue and empty VarRefTag	Core	Y
5	Concatenate with SPS and temporary variable	SPS	Y
8	Concatenate with 2 VarRefTags with length 1. = 0x7E + 2.=0x02	Core	Y
9	Concatenate with 1 VarRefTags with length 1. = 0x7F to same Reference	Core	Y

### 12.1.4 Error test

Reference	Description	Component Name	Shared
10	Concatenate to a bad ref to DestVarID=0xc0-0xff	Core	Y
11	Concatenate with no memory space left	Core_memory	N
12	Concatenate to a bad ref to DestVarID=0x80-0xbf	SPS	Y

## 12.2 Extract tag

### 12.2.1 Description

This chapter describes the tests used to validate the correct processing of the extract tag by the browser.

### 12.2.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.5

### 12.2.3 Normal test

Reference	Description	Component Name	Shared
1	Extract from a temp variable first	Core	Y
2	Extract from a temp variable last	Core	Y
3	Extract from a temp variable with complete length	Core	Y
4	Extract from a temp variable with zero length	Core	Y
5	Extract from a TextElement first	Core	Y
6	Extract from a TextElement last	Core	Y
7	Extract from a TextElement with complete length	Core	Y
8	Extract from a TableElement with zero length	Core	Y
9	Extract from a Permanent Var Ref first	SPS	Y
10	Extract from a Permanent Var Ref last	SPS	Y
11	Extract from a Permanent Var Ref with complete length	SPS	Y
12	Extract from a Permanent Var Ref with zero length	SPS	Y
17	Extract with a Length > (Len(InputVarId)-StartIndex)	Core	Y

### 12.2.4 Error test

Reference	Description	Component Name	Shared
13	Extraction to a bad DestVarID=0xc0-0xff	Core	Y
14	Extract from an undefined variable	Core	Y
15	Extract without enough memory space	Core_memory	N
16	Extract with a StartIndex bigger than the length of InpVarId	Core	Y
18	Extraction to a bad DestVarID=0x80-0xBF	SPS	Y

## 12.3 SetHelp tag

### 12.3.1 Description

This chapter describes the tests used to validate the correct processing of the SetHelp tag by the browser.

### 12.3.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.3

### 12.3.3 Normal test

Reference	Description	Component Name	Shared
1	Set Help with 1 initialised Variable Reference followed with STK Command GET INPUT	Core	Y
2	Set Help with 3 initialised Variable Reference followed with STK Command Select ITEM	Core	Y
3	Set Help with 1 Inline Value Element followed with an empty VarRef and GET INPUT	Core	Y
4	Set Help with 3 inline values followed with STK Command Select ITEM	Core	Y
5	SetHelp with Attribute byte Bit 6 =1 (Reset Flag=1)	Core	Y

### 12.3.4 Error test

Reference	Description	Component Name	Shared
6	Set Help with reference to undefined variable Element	Core	Y
7	Set Help with no more memory space left	Core	N

## 12.4 GetEnv tag

### 12.4.1 Description

This chapter describes the tests used to validate the correct processing of the GetEnv tag by the browser.

### 12.4.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.2.1

### 12.4.3 Normal test

Reference	Description	Component Name	Shared
1	Getenv ICCID of the SIM	Core	Y
2	Getenv SIMBrowserSupplier	Core	Y
3	Getenv Browser Version	Core	Y
4	Getenv BrowserProfile	Core	Y
5	Getenv TerminalProfile	Core	Y
6	Getenv StatusWord	Core	Y
7	Getenv OperatorName	N	Y
8	Getenv LocationInformation	N	Y
9	Getenv UserType	N	Y
10	Getenv from a not defined ID	Core	Y

### 12.4.4 Error test

Reference	Description	Component Name	Shared
11	Getenv with no more memory space left	Core	N
12	Getenv with initialising a text element	Core	Y



## 12.5 InitVarSelected tag

### 12.5.1 Description

This chapter describes the tests used to validate the correct processing of the InitVarSelected tag by the browser.

### 12.5.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.1.2

### 12.5.3 Normal test

Reference	Description	Component Name	Shared
1	Init Variable Selected with select Item Titel Select item 1. Couple: VarRef,VarRef 2. Couple InVal, InVal	Core	Y
2	Init Variable Selected with select Item Titel Select item 1. Couple: VarRef, InVal, 2. Couple: InVal, VarRef	Core	Y
3	Init Variable Selected with select Item Titel Select item with 10 Couples	Core	
4	Init Variable Selected with select Item Titel Select item 1. Couple: item: VarRef="", value: VarRef=defined 2. Couple: item: VarRef=defined, value: InVal=""	Core	Y

### 12.5.4 Error test

Reference	Description	Component Name	Shared
5	Init Variable Selected with select Item Titel Select item 1. Couple: item: VarRef=undef, value: VarRef=defined	Core	Y
6	Init Variable Selected with select Item Titel Select item 1. Couple: item: VarRef=defined, value: VarRef=undef	Core	Y
7	Init Variable Selected with no more memory space left	Core	N
8	Init Variable Selected with initialise a wrong DestVarId=0xC0- 0xFF	Core	Y
9	Init Variable Selected STK command embedded failed (too long)	Core	Y

## 12.6 InitVar tag

### 12.6.1 Description

This chapter describes the tests used to validate the correct processing of the InitVar tag by the browser.

### 12.6.2 Specification references

The specifications are written in:

- S@T 01.00 v1.03
- SBC 6.2.1.1

### 12.6.3 Normal test

Reference	Description	Component Name	Shared
1	Init Variable with Inline Value Element	Core	Y
2	Init Variable with Variable Reference to temporary variable	Core	Y
3	Init Variable with Variable Reference to table element	Core	Y
4	Init Variable with Variable Reference to permanent variable	SPS	Y
5	Init Variable with empty Inline Value Element	Core	Y
6	Init Variable with Variable Reference to empty temporary variable	Core	Y
7	Init Variable with Variable Reference to empty table element	Core	Y

### 12.6.4 Error test

Reference	Description	Component Name	Shared
8	Init Variable with undefined reference	Core	Y
9	Init Variable with no more space left	Core	N
10	Init Variable with a wrong DestVarID=0xC0-0xFF	Core	Y

## 12.7 Encryption

### 12.7.1 Description

This chapter describes the tests used to validate the usage of the encrypt command.

### 12.7.2 Specification references

The specification is written in:

- [S@T](#) 01.00 v1.0.3 SBC
- Further information in GSM 3.48.

Tested points:

- Correct handling of encrypt command parameters

- Correct padding (multiple of 8)
- Correct handling of memory problems
- Correct handling of bad references

### 12.7.3 Normal test

Reference	Description	Component Name	Shared
1	Encryption (none selected); length of sum of LVs of variable list #1 is not a multiple of 8	CORE	YES
2	Encryption (CC); length of sum of LVs of variable list #1 is not a multiple of 8	CORE	YES
3	Encryption (cyphering); length of sum of LVs of variable list #1 is not a multiple of 8	CORE	YES
4	Encryption (CC and cyphering); length of sum of LVs of variable list #2 is a multiple of 8	CORE	YES
5	Encryption (CC and cyphering) length of sum of LVs of variable list #1 is not a multiple of 8	CORE	YES

### 12.7.4 Error test

Reference	Description	Component Name	Shared
6	Encryption (CC and cyphering) with memory problems; length of sum of LVs of variable list #1 is not a multiple of 8;	CORE	NO
7	Encryption (CC and cyphering) with bad key identifier length of sum of LVs of variable list #1 is not a multiple of 8	CORE	YES
8	Encryption (CC and cyphering) with bad destination var ID length of sum of LVs of variable list #1 is NOT a multiple of 8	CORE	YES
9	Encryption (CC and cyphering) with bad source var ID in reference list length of sum of LVs of variable list #1 is NOT a multiple of 8	CORE	YES
10	Encryption when a Kid is referencing a bad key.	CORE	YES
11	Encryption when a Kic is referencing a bad key.	CORE	YES

## 12.8 Decryption

### 12.8.1 Description

This chapter describes the tests used to validate the usage of the decrypt command.

### 12.8.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)
- Further information in GSM 3.48.

Tested points:

- Correct handling of decrypt command parameters

- Correct recognition of corrupted keys
- Correct handling of memory problems

### 12.8.3 Normal test

Reference	Description	Component Name	Shared
1	Decryption (no encryption selected)	CORE	YES
2	Decryption (encrypted done using CC) length of sum of LVs of variable list #1 was not a multiple of 8	CORE	YES
3	Decryption (encrypted done using cyphering) length of sum of LVs of variable list #1 was not a multiple of 8	CORE	YES
4	Decryption (encrypted done using CC and cyphering) length of sum of LVs of variable list #2 was a multiple of 8	CORE	YES
5	Decryption (encryption done using CC and cyphering) length of sum of LVs of variable list #1 was not a multiple of 8	CORE	YES

### 12.8.4 Error test

Reference	Description	Component Name	Shared
6	Decryption (encryption done using CC and no cyphering) with corrupted MAC	CORE	YES
7	Decryption (encryption done using CC and cyphering) with corrupted MAC	CORE	YES
8	Decryption (encryption done using CC and cyphering) with memory problems	CORE	NO
9	Decryption (encryption done using CC and cyphering) with bad reference to SecMsg	CORE	YES

## 12.9 Go Back

### 12.9.1 Description

This chapter describes the tests used to validate the reception and the sending of go back command.

### 12.9.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Correct handling of go back command parameters
- Correct handling of history decks
- Correct handling of bad references
- Correct handling of memory problems

### 12.9.3 Normal test

Reference	Description	Component Name	Shared
1	Go Back with "RestartCurrentCard" choose card1 of two	CORE	YES

2	Go Back with "RestartCurrentCard" choose card2 of two	CORE	YES
3	Go Back without "RestartCurrentCard"	CORE	YES
4	Go back on a card in a dynamic deck that is still resident	CORE	YES
5	Go back on a card in a static deck	CORE	YES
6	Go back one step without out putting this step into the history	CORE	YES

#### 12.9.4 Error test

Reference	Description	Component Name	Shared
7	Go Back when the history is empty	CORE	YES

## 12.10 Go Selected

### 12.10.1 Description

This chapter describes the tests used to validate the reception and the sending of go selected command.

### 12.10.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Correct handling of go selected command
- Correct handling of go direct command
- Correct handling of bad references
- Correct handling of memory problems

### 12.10.3 Normal test

Reference	Description	Component Name	Shared
1	Go Selected with a title as inline value (title format inherited from the deck)	CORE	YES
2	Go Selected with a title as inline value in 7-bit SMS format	CORE	YES
3	Go Selected without a title	CORE	YES
4	Go Selected with a single URL with title	CORE	YES
5	Test Go Selected with one couples	CORE	YES
6	Test Go Selected with multiple couples	CORE	YES
7	Go Selected (1 couple) from homepage to a card on a distant deck	CORE	YES
8	Go Selected (2 couples) from homepage to a card on a distant deck	CORE	YES
9	Go Direct from homepage to a distant deck	CORE	YES
10	Go Direct from homepage to to a card on a distant deck	CORE	YES
11	Go Selected, local navigation	CORE	YES
15	Inter-card navigation with url parameters in Go selected / Go direct	CORE	YES
16	Inter-deck navigation with url parameters in Go selected / Go direct	CORE	YES
17	Inter-card navigation with post method in Go selected / Go direct	CORE	YES
18	Inter-deck navigation with post method in Go selected / Go direct	CORE	YES

28	Go distant without parameters from a deck with a cleanup variable list element	CORE	YES
29	Go distant with parameters from a deck with a cleanup variable list element. The parameters are members of the cleanup variable list.	CORE	YES

#### 12.10.4 Error test

Reference	Description	Component Name	Shared
12	Test Go Selected with a memory allocation problem	CORE	NO
13	Test Go Selected with reference to an undefined variable	CORE	YES
14	Test Go Selected over the ME capacity	CORE	YES
19	Intra_deck navigation with go selected element to an unknown card	CORE	YES
20	Intra_deck navigation with go direct to an unknown card	CORE	YES
21	Local navigation with go selected element to an unknown deck	CORE	YES
22	Local navigation with go direct to an unknown deck	CORE	YES
23	Local navigation with go selected element to an unknown card in a local deck	CORE	YES
24	Local navigation with go direct to an unknown card in a local deck	CORE	YES
25	Local navigation with unknown variables in Go selected element	CORE	YES
26	Local navigation continuation of TN 26 after restarting browser	CORE	YES
27	Local navigation continuation of TN 27 after restarting browser	CORE	NO
30	Go selected with unknown url from a deck with a cleanup variable list element.	CORE	YES

## 12.11 Switch case on variable

### 12.11.1 Description

This chapter describes the tests used to validate the reception and the sending of switch case on variable command.

### 12.11.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Correct handling of switch case on variable command
- Correct handling of default case parameters
- Correct handling of bad cards and variables

### 12.11.3 Normal test

Reference	Description	Component Name	Shared
1	Switch case on variable case sensitive, compare lower to lower, whereby several strings match the compare string	CORE	YES
2	Switch case on variable case sensitive, compare lower to lower, whereby no substring matches the compare string	CORE	YES

3	Switch case on variable case sensitive, compare lower to lower, whereby the compare string is of maximum length (L=0xFE) and identical to one list element	CORE	YES
4	Switch case on variable case sensitive, compare upper to upper	CORE	YES
5	Switch case on variable not case sensitive, compare lower to upper	CORE	YES
6	Switch case on variable not case sensitive, compare upper to lower	CORE	YES
7	Switch case on variable with default value, variable found	CORE	YES
8	Switch case on variable with default value, variable not found	CORE	YES
9	Switch case on variable with default value and empty couple list	CORE	YES

#### 12.11.4 Error test

Reference	Description	Component Name	Shared
10	Switch case on variable case sensitive, compare lower to upper	CORE	YES
11	Switch case on variable case sensitive, compare upper to lower	CORE	YES
12	Switch case on variable with a reference to an undefined text variable	CORE	YES
13	Switch case on variable on a bad card reference	CORE	YES

## 12.12 Terminate Browser Session

### 12.12.1 Description

This chapter describes the tests used to validate the reception and the sending of exit command.

### 12.12.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Correct handling of exit command parameters
- Correct handling of execution buffer

### 12.12.3 Normal test

Reference	Description	Component Name	Shared
1	Exit without cleaning the execution buffer	CORE	YES
2	Exit cleaning the execution buffer	CORE	YES
3	Exit with an output variable list	CORE	NO

## 12.13 Manage Contextual Menu Item

### 12.13.1 Description

This chapter describes the tests used to validate the reception and the sending of manage contextual menu item command.

## 12.13.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)
- [S@T 01.20 v1.0.3 SSP](#)

Tested points:

- Correct handling of manage contextual menu item command parameters
- Correct handling of hide/display function
- Correct handling of update/insert/delete function
- Correct handling of administrative / operational session
- Correct handling of system / application menu items

## 12.13.3 Normal test

Reference	Description	Component Name	Shared
1	Hide a contextual menu system item in operational mode. Changes to be made will be stored temporarily.	CORE	YES
2	Display a contextual menu system item in operational mode. Changes to be made will be stored temporarily.	CORE	YES
3	Add a contextual menu application item in operational mode. Changes to be made will be stored temporarily.	CORE	YES
4	Update an contextual menu application item in operational mode. Changes to be made will be stored temporarily.	CORE	YES
5	Remove a contextual menu application item in operational mode. Changes to be made will be stored temporarily.	CORE	YES

## 12.13.4 Error test

Reference	Description	Component Name	Shared
6	Hide a contextual menu system item in operational mode.	CORE	YES
7	Display a contextual menu system item in operational mode.	CORE	YES
8	Add a contextual menu application item in operational mode.	CORE	YES
9	Update a contextual menu application item in operational mode.	CORE	YES
10	Remove a contextual menu application item in operational mode.	CORE	YES
11	Add a contextual menu application item . No space left.	CORE	NO
12	Manage contextual menu item with reference to undefined text element.	CORE	YES
13	Try to hide non existing menu item	CORE	YES

## 12.14 STK Generic Macro

### 12.14.1 Description

This chapter describes the tests about the translation of STK Byte Code in STK commands.

### 12.14.2 Specification references

The specification is written in:



- [S@T](#) 01.00 v1.0.3 SBC

Tested points:

- Correct handling of STK generic macro command parameters
- Correct handling of command type, command qualifier,
- Correct handling of destination device identity
- Correct handling of various TLV parameters of the command

### 12.14.3 Normal test

Reference	Description	Component Name	Shared
1	Perform STK Generic Macro based on "DISPLAY TEXT" without Output variable id	CORE	YES
2	Perform STK Generic Macro based on "Get Input" with response data object Value = TLV , LV Attribute: 0 with an Output variable id	CORE	YES
3	Perform STK Generic Macro based on "Provide Local Information" with response data object Value = TLV1... TLV N , LV Attribute:0	CORE	YES
4	Perform STK Generic Macro based on "Get Input" with response data object Value = TLV , LV Attribute: 1	CORE	YES
5	Perform STK Generic Macro based on "Provide Local Information" with response data object Value = TLV1...TLV N , LV Attribute: 1	CORE	YES
6	Perform STK Generic Macro with variable substitution applied on a variable with len=0	CORE	YES
7	Perform STK Generic Macro with variable substitution applied on a variable with len=64 with DCS	CORE	YES
8	Perform STK Generic Macro with variable substitution applied on a variable with len=64 without DCS	CORE	YES

### 12.14.4 Error test

Reference	Description	Component Name	Shared
9	Perform STK Generic Macro with variable substitution applied on a temporary variable with a bad reference Id (non existing ID)	CORE	YES
10	Perform STK Generic Macro with variable substitution applied on a variable with len=254	CORE	YES
11	Perform STK Generic Macro with a reference to an undefined destination variable ID (type: permanent)	CORE	YES
12	Perform STK Generic Macro with a reference to an undefined variable ID (type: permanent or text element) within variable substitution	CORE	YES
13	Perform STK Generic Macro with memory problems for STK command preparation	CORE	NO
14	Perform STK Generic Macro with a reference to an undefined destination variable ID (type: text element)	CORE	YES
15	Perform STK Generic Macro with a STK use failed applied on "DISPLAY TEXT"	CORE	YES

## 12.15 Execute

### 12.15.1 Description

This chapter describes the tests used to validate the reception and the sending of execute command.

## 12.15.2 Specification references

The specification is written in:

- [S@T 01.00 v1.0.3 SBC](#)

Tested points:

- Correct handling of execute command
- Correct handling of input and output parameters
- Correct handling of exit attribute

## 12.15.3 Normal test

Reference	Description	Component Name	Shared
1	Call execute element with 1 in- but without output parameters (exit=1)	CORE	YES
2	Call execute element with 1 in- and 1 output parameter (exit=0)	CORE	YES
3	Call execute element with 3 in- and 3 output parameters (exit=0)	CORE	YES
4	Call execute element with 1 in- but without output parameters (exit=0)	CORE	YES
5	Call execute element without in- but with 1 output parameter (exit=0)	CORE	YES

## 12.15.4 Error test

Reference	Description	Component Name	Shared
6	Call execute element with too many input parameters (exit=0)	CORE	YES
7	Call execute element with too few output parameters (exit=0)	CORE	YES
8	Call execute element with too few input parameters (exit=1)	CORE	YES
9	Call execute element with input parameter and output variable reference to text element	CORE	YES
10	Call execute element with input parameter and undefined (not existing) output variable reference to SPS	CORE	YES
11	Call execute element with input parameter and undefined (not existing) output variable reference to text element	CORE	YES
12	Call execute element with in- and output parameters and wrong format of the input parameter in order to provoke an execution failure	CORE	YES

# 13 GATEWAY TESTS [Not implemented]

## 13.1 SSP/3.48/3.40 TESTS

Description

In aim to check these layers independently, a test mean is introduced to bypass the proprietary [S@TML](#) to SBC conversion function. A set of test decks will be defined to be transferred via the SSP/3.48 layer directly. The proposal is to introduce an alternate protocol to *http://*, the *sattp://*

Requests received by the gateway(s) should be processed in such a way that the referred test deck file is fetched and sent over the underlying layers.

The file format of that files will be as simple as text files containing Hex bytes values.

In aim to avoid the need for short name conversions for decks, only long names will be used inside these test decks when required.

## 13.2 SBC GENERATION TESTS

### Description :

The objective here is to intercept all the SBC generated by a number of reference pages and to analyse the SBC for compliance with [S@T](#). This analysis is both syntactic and partially semantic. As the [S@TML](#) translation is not common, there is no 1-1 relation from a given [S@TML](#) page to a deck in SBC and this kind of analysis is required.

## 13.3 ADMINISTRATIVE COMMANDS TESTS

### Description

One need to examine the SBC generated by all the administrative commands to check for compliance with specification.

The question here is to know how one can initiate that administrative command on the server as no standard way to do it is specified in the [S@T](#).

Solution : only a specification is required and each gateway provider can develop this test subset.

## 13.4 [S@TML](#) ACCEPTANCE TESTS

### Description

Check the acceptance or rejection of [S@TML](#) pages with regard to DTDs, TAGs...

## 13.5 GATEWAY REQUESTS PROCESSING TESTS

### Description

Check that the SBC requests are correctly transformed to issue http requests. (management of parameters, post, get...)

## 14 ANNEX : DETAILED TESTS DOCUMENTS

The current document indicates the list of tests per type.

A complete architecture of documents is provided to SIM @lliance members to improve efficiency in test development. It is composed of Detail tests sheets, Test Decks description, and decks bytecode and interpretation files.

The detailed description of the previously listed tests are found in one excel sheet.

The tests are performed on a set of Decks. The decks are referred in the test description sheets and found in a set of word documents. For each deck, the bytecode is given with its high level interpretation. The bytecode files are also provided separated (.sbc or .hsbc) to be directly usable by test suites. An HTML version of the decks is also provided to ease the reading of the bytecode.

### 14.1 Detail test sheets

Each test described in this document is described in detail in an Excel sheet.

Each test is composed from 1 to N phases, each corresponding to an action and a verification clause. (Column *Phase* used to index the phases)

The *Component* column is specifying the component that is exercised by the test.

The *Test Type* column is specifying if the test is a Normal, Error or Other case. **Normal case means that the bytecode is well formed and semantically correct, Error cases are made to trigger exeptions as specified in [S@T](#) specifications.**

The *Init state* is specified to indicate what preliminary action is performed prior to test execution.

The tests put decks in action. For the purpose of the test, one or several decks are required. Their names are indicated in the *Test Decks Reference* column

For each phase, the column *Continue on Error* is specifying if the test is continued to next phase or not when a verification is negative.

The column *Parameter names and values* is used when the same test has to be performed for a predefined set of parameters values. (this to avoid a duplication of the test description)

The deck descriptions are provided in separate documents

## 14.2 Test decks descriptions

Several word documents are provided to describe the decks. The decks are referred by the test cases, and some decks are used by several test cases.

The *deck names* are used as cross reference between the documents.

The column *Home/Temp* is used to specify if the deck has to be installed as a home deck or used as a temporary deck.

The *Deck* description is an informal description of the deck function.

The *Bytecode* column contains the bytecode of the deck in Hex string.

The *Code dump* column contains an interpreted version of the bytecode.

## 14.3 Test decks bytecode and dump files

The content of deck bytecode and deck code dump is provided in another set of files.

The name of the corresponding files are the name of the test followed by *.sbc*(binary) or *.hsbc*(Hexa string of sbc)

The CodeDump files are stored in *.HTML* format.

The files *<TestName>\_frames.HTML* contain bytecode and interpretation in separate frames.

The files *<TestName.HTML>* contain bytecode only.

These files are useful for test development purposes and available on request to SimAlliance.