S@T 01.00 v4.0.0 (Release 2009)

S@T Bytecode

Published by simalliance now Trusted Connectivity Alliance

TABLE OF CONTENTS

1 TERMINOLOGY

1.1 Notation

Prefix ‘0x’ indicates hexadecimal value. ‘bn’ indicates individual bit in a byte. Range from bit 0 (least significant), denoted b0, to bit 7 (most significant), denoted b7.

1.2 Abbreviations

LV	Length Value encoding
ME	Mobile Equipment
MSB	Most Significant Bit
S@T	SIM Alliance Toolbox
SBC	S@T Byte Code
SSP	S@T Session Protocol
S@TML	S@T Markup Language
STK	SIM Application Toolkit
TL[A]V	TLV with optional attribute bytes
TLV	Tag Length Value encoding
URL	Unified Resource Locator

1.3 Definitions

Address reference: is used to navigate to a card on the same deck or another deck and can contain deck Id or deck URL plus optionally card Id or only card Id.

Card: the smallest navigation unit, part of deck

Current card: card which is currently executed by the S@T browser

Current deck: deck which is currently executed by the S@T browser

Deck: the smallest unit that can be loaded into the S@T browser for execution.

Deck URL: is used to identify online deck on the S@T gateway, can be long or coded URL.

Dynamic deck: online deck which shall not be cached by the browser, Dynamic / Static attribute in Deck is set.

Online deck: deck received from the gateway for execution

Resident deck: deck which is stored in a SIM card

Static deck: online deck which can be cached by the browser, Dynamic / Static attribute in Deck is not set.

URL reference: address reference with optional parameters

2 LIST OF DOCUMENTS

/23.038/ 3GPP TS 23.038: "Alphabets and language-specific information".

/51.011/ 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface".

/51.014/ 3GPP TS 51.014: "Specification of the SIM Application Toolkit (SAT) for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface".

/23.048/ 3GPP TS 23.048: "Security mechanisms for the (U)SIM application toolkit; Stage 2".

/ISO/IEC 7816-6/ ISO/IEC 7816-6: 'Identification cards - Integrated circuit(s) cards with contacts, Part 6 Inter-industry data elements'.

/SSP/ S@T 01.20: 'S@T Session Protocol'.

/Admin/ S@T 01.21: 'S@T Administrative Commands'.

/Pull/ S@T 01.22: 'S@T Operational Commands'.

/Push/ S@T 01.23: 'S@T Push Commands'.

This document is part of a specification set, please refer to 'S@T Release Note' for a comprehensive document list, including document versions.

3 OVERVIEW

This document describes S@T bytecode that is recognized by S@T browser. S@T bytecode can be dynamically loaded from S@T gateway or allocated resident.

S@T bytecode primary purpose is to provide efficient programmatic access to the STK commands. S@T bytecode also contains commands used to perform additional local processing or branching.

The design objectives of the bytecode are:

- compact representation for efficient transmission over the air interface;
- to keep S@T browser complexity as low as possible to minimize SIM footprint.

S@T bytecode inherits some WML concepts like deck and card organization to match the WML mediation needs.

Some trade-offs have been made between S@T bytecode efficiency and its ease of parsing. It has been taken into account the evolution too, in aim to allow an easy extension or an incomplete implementation for non mandatory elements.

4 DECK EXECUTION MODEL

4.1 Introduction

The purpose of project S@T browser is to offer a generic browsing tool embedded in a SIM card. The sources of information to browse are either WML or S@TML pages. They are located either on operator's intranet or on the web.

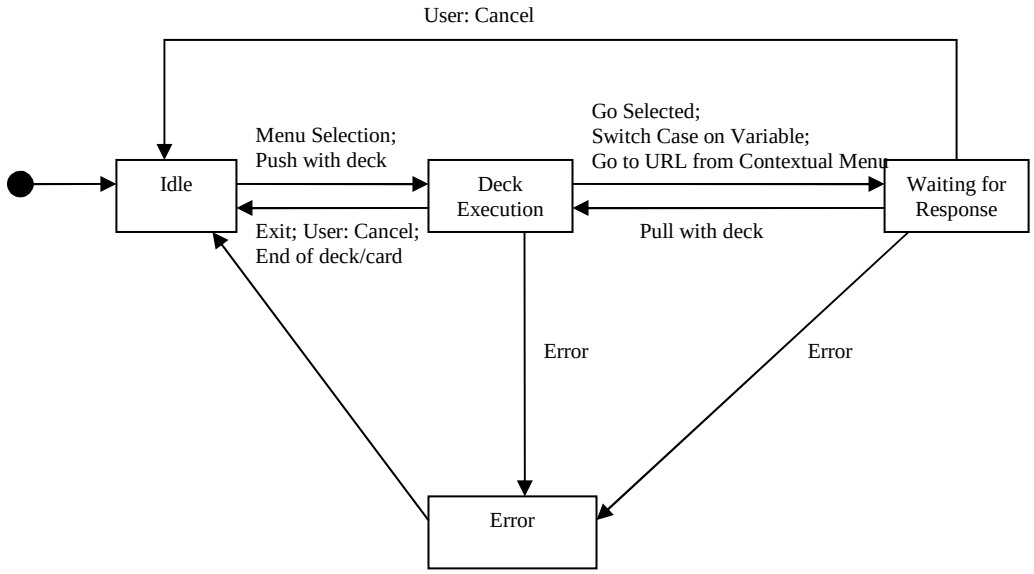
A service can be seen as a set of pages that are browsed successively with interactions to the user. A service is expected to be provided by a Service Provider.

The service pages are converted to decks. Deck is the smallest unit of transmission to S@T browser as well as the unit of S@T browser interpretation. Deck contains set of cards. The card element specifies one unit of navigation and contains S@T bytecode commands.

A card can contain list of resident decks (stored in a SIM card during pre-personalization phase or installed from S@T gateway via administrative commands as defined in /Admin/) and resident menu with items referencing resident decks. Online deck can be received from the gateway via pull (see /Pull/) or push (see /Push/).

The browser renders decks and provides a way to navigate within decks to cards belonging to the same deck or other decks.

4.2 High Level State Diagram



The S@T browser states correspond to the presence or the origin of proactive session. The base states are:

- Idle: no proactive session is running (i.e. no proactive command is pending);
- Deck Execution: proactive session issued from S@T bytecode command;
- Waiting for Response: browser keeps the proactive session alive by issuing DISPLAY TEXT to notify the user that the browser is waiting for or receiving the gateway response.

After SIM card reset the S@T browser goes to the idle state.

The S@T browser can leave the idle state and start deck execution in different ways:

- locally from the ME using menu selection: the browser starts executing resident deck referenced by the selected menu item;
- by an incoming online deck pushed by S@T gateway (see /Push/).

The S@T browser enters waiting for response state after requesting online deck originated in:

- Go Selected command;
- Switch Case on Variable command;
- contextual menu;
- bookmark list.

When S@T browser receives a response with the requested deck, it starts executing the deck.

S@T browser exits (i.e. goes to the idle state) after:

- the last command of the card has been executed and no branching has been done;
- Exit command has been executed;

- proactive session has been terminated by user;
- error handling.

4.3 Navigation

Deck is the smallest unit that can be executed by the S@T browser. A deck is partitioned into one or more cards each of which can be referenced using card Id.

The card is the elementary navigation target. The S@T browser can skip from one card to another, backwards and forwards based either on control flow commands like Go Selected, Switch Case on Variable and Go Back or user interaction. Cards can be also linked together by means of ChainNextCard flag in card attribute (as defined in 7.2). If a card contains no command to branch to a card within the current deck or another deck and is not linked to the next card, normally the S@T browser exits after processing the last command of the card.

To be able to create multiple-page services, address references within S@T bytecode commands are used to fetch new decks or to link decks together.

The behavior of the S@T browser in response on user interaction (e.g. backward move, proactive session terminated) is defined by the current Contextual Menu configuration. The contextual menu can be modified by Manage Contextual Menu Item command within the deck or card context.

In case of navigation action to a card of the current deck, the deck shall not be retrieved again from the gateway.

4.4 History Management

The S@T browser shall keep history (back and forward) of the last visited cards. A card is added on the top of the history when the browser leaves the card (i.e. when the S@T browser starts to render another card or goes to the waiting for response state). A card shall not be added to the history if DoNotHistorize flag is set in the card attribute.

The history navigation action ‘go back’ means that:

- the current card (if any) shall be added on the top of the forward history if DoNotHistorize flag is not set
- the card on the top of the back history shall be removed from the back history and rendered

The origin of this action can be:

- contextual menu system item ‘Back to previous card’;
- Go Back command with RestartCurrentCard flag not set;
- backward move on the waiting for response state or on the bookmark list.

When the history list is empty and a ‘go back’ history navigation action has to be performed, the S@T browser shall exit.

The history navigation action ‘go forward’ means that:

- the current card shall be added on the top of the back history if DoNotHistorize flag is not set;
- the card on the top of the forward history shall be removed from the forward history and rendered.

The origin of this action can be:

- contextual menu system item ‘Next (after back)’.

When branching to another card happens not because of ‘go back’ or ‘go forward’ history navigation, the current card shall be added on the top of the back history and forward history shall be cleared.

Go Back command with RestartCurrentCard flag set shall not modify the history list.

If the history is full, the bottom-most entry is removed from the list in order to free space for a new top-most entry.

The history is reset (is emptied) when

- S@T browser exits (i.e. goes to the Idle state);
- high priority push is received (see /Push/);
- an error happens.

For history navigation to an online deck, the browser shall keep URL reference (deck URL with parameters) that was used to retrieve the deck.

4.5 Deck Caching

A flag in deck TL[A]V attribute indicates whether to cache ("Static") the deck or not ("Dynamic"). On reception of a static deck from the gateway the browser shall cache the deck. There is no caching for dynamic decks.

For the cached decks the browser shall keep:

- URL Reference TL[A]V that was used to request the deck.
- Deck TL[A]V.

When an online deck is requested either by "go back" or "go forward" history navigation or by branching command, before requesting the deck from the gateway, the browser shall look it up in the cache. If the requested URL Reference matches URL Reference of a cached deck, the cached deck shall be executed by the browser. A URL Reference matches another URL reference if:

- deck URLs (i.e. values of Address Reference TL[A]Vs without Card Id) are equal;
- attributes are equal;
- URL parameters are equal.

If the requested URL Reference is not found in the cache, the browser shall retrieve this URL Reference from the gateway.

The cache shall be cleaned when browser exits (i.e. goes to the idle state).

If there is no memory to cache a deck, the oldest cached deck shall be overwritten. If the deck or URL reference is too large to be cached, the deck shall not be cached.

4.6 Contextual Menu

The S@T browser in response on user interaction may branch to the contextual menu. The support of Contextual Menu is optional. The S@T browser supported Contextual Menu shall handle the general results according to the following rules:

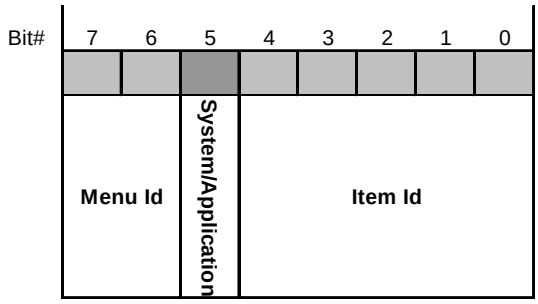
- For general result 'backward move requested by user' (0x11 as defined in /51.014/) the S@T browser shall branch to Back Menu.
- For general result 'proactive session terminated by user' (0x10 as defined in /51.014/) the S@T browser shall branch to Abort Menu.
- The S@T browser shall branch to the contextual menu only if a proactive command issued from a S@T bytecode command (defined in chapter 9), i.e. the browser is not in wait for response state, not in error handling state, not showing the contextual menu and etc.

4.6.1 Contextual Menu Item Coding

Contextual menu can contain system items with predefined action and application elements that allow retrieving any URL reference. Contextual menu can be temporary changed in deck or card context from Manage Contextual Menu Item command that allows to:

- hide or display (make visible) a system item;
- hide or add/update an application item.

Coding of contextual menu item in Manage Contextual Menu Item command is the following:



b7b6 – Menu Id:

00: Back Menu

10: Abort Menu

Other: RFU

b5 – System/Application:

0: Application item

1: System item

b4-b0 – Item Id

4.6.2 Predefined System Items

Menu Item Name	*Menu Item Id*	*Predefined Action*
Back to previous card	xx1 00001	History navigation “go back” (as defined in 4.4)
Next (after back) [1)]	xx1 00010	History navigation “go forward” (as defined in 4.4)
Go to home deck	xx1 00100	Go to the resident deck with deck Id “/s/home”
Go to bookmark list	xx1 00101	Go to the bookmark list (“/s/bookmarks” address reference)
Stop browser	xx1 00111	Exit S@T browser

[1)] NOTE: Menu item “Next (after back)”shall be automatically hidden if no history navigation is available (i.e. if forward history is empty).

4.6.3 Default Contextual Menu Configuration

The proposed default terminal response handler configuration may be modified at personalization stage by the card issuer.

Menu Item Name	*Menu Item Id*	*Back Menu xx=00*	*Abort Menu xx=10*
Back to previous card	xx1 00001	√	
Stop browser	xx1 00111		√

NOTE: There is no need to display a contextual menu with only one item; in this case the action shall be executed immediately when the menu is invoked. Anyway, for each of the system actions a text shall be assigned and shall to be used in the SELECT ITEM if more application items are added from the S@T bytecode.

NOTE: The S@T browser not supporting contextual menu shall behave as if default contextual menu is configured and the action is executed immediately.

5 VARIABLE MANAGEMENT

5.1 Introduction

Variables are name-value pairs. The name is called the variable identifier (Id) and the value is called the variable value. Operations are provided to refer to a variable value by using its variable Id and for setting and resetting the value associated with a variable.

Variables can be stored in the following usage areas:

- Permanent variable area;
- Temporary variable area;
- Deck text elements;
- Environment variable area.

Variables have one of the following variable DCS:

- SMS 7 bit unpacked: SMS default 7bit coded alphabet as specified in /23.038/ with bit 8 set to 0;
- UCS2;
- Binary.

5.2 Variable Usage

Variables are referred by using a unified one byte notation. The one byte variable reference is called the variable Id.

The coding of the variable Id is as follows:

Bit#	7	6	5	4	3	2	1	0	
	0	Index							Temporary variable reference
	1	0	Index						Permanent variable reference
	1	1	Index						Deck text element reference

Due to the used coding, the number of variables per area:

Temporary variables: 128 variables

Permanent variables: 64 variables

Deck text elements: 64 variables

5.3 Temporary Variables

Temporal variables are managed from the S@T bytecode (i.e. set, update, read or reset). Temporary variables are used during execution of the current deck and shared with the following decks (after branching).

If the S@T browser is not able to create a new temporary variable due to the limits of the temporary variable area memory space, the S@T browser shall generate a 'Problem in memory management' error.

Temporary variables shall be cleared (i.e. removed) when:

- the S@T browser starts a card with ResetVar flag set in the card attribute;

- the S@T browser goes to the idle state;
- high priority push is received (see /Push/).

5.4 Permanent Variables

The support of permanent variables is optional.

The S@T browser supporting permanent variables shall keep them in non-volatile memory (i.e. permanent variables can be accessed after SIM reset). Permanent variables can be read, created or updated from any deck.

If the S@T browser is not able to create/update a permanent variable due to the limits of the permanent variable area memory space, the S@T browser shall generate a 'Problem in memory management' error.

5.5 Deck Text Elements

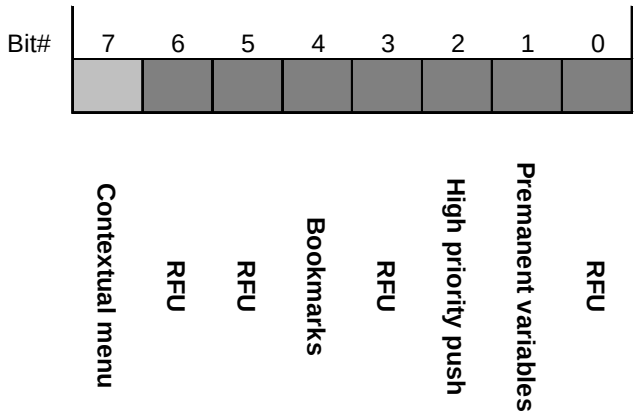
This area is provided optionally by the current deck (as defined in 7.1.2). It can be used to store e.g. strings that are used several times in the current deck. Text elements shall be considered as constants and cannot be modified by the S@T bytecode.

5.6 Environment Variables

Environment variables cannot be updated from the S@T bytecode. They can be updated only by the S@T browser internally. An environment variable value can be read by Get Environment Variable command.

The following environment variables are defined for the S@T browser:

ID	*NAME*	*LENGTH*	*CODING*	*M /O*
0x00	ICCID	10	Coding according to /51.011/ as in EF ICCID	M
0x01	Browser Supplier	1	0x01: Giesecke&Devrient 0x02: Gemplus / Gemalto 0x03: Sagem Orga 0x04: Axalto / Gemalto 0x05: Oberthur Technologies 0x06: XPonCard 0x07: Prism 0x08: Incard 0xFF: Other	M
0x02	Browser Version	1	High nibble: S@T Version number (current version is 4) Low nibble: Manufacturer Release number	M

0x03	Browser Profile	1	Coding:  **bn – facility:** 0: not supported 1 : supported	M
0x05	Terminal Profile	X	Coding as defined in /51.014/ for TERMINAL PROFILE	M
0x06	Error Code	2	Error Code of the previous executed S@T bytecode command as defined in section 10.	M
0x07	Location information [1)]	7	Coding as defined in /51.014/ for Location Information	O
0x09	IMEI [1)]	8	Coding as defined in /51.014/ for IMEI	O
0x0A	NMR [1)]	16	Coding as defined in /51.014/ for Network Measurement Results	O
0x40	Resident Decks Buffer Size	>= 2	Resident decks buffer size, i.e. space allocated for residents deck. Installing or uninstalling decks will not change this value.	M
0x42	Resident Deck List	X	List of the all resident decks. Coding : list of LVs containing Deck Ids	M
0x43	Plug-in List	2 * N	List of the all activated S@T plug-ins. Coding : list of plug-in names (2 bytes per each name)	M
0x45	Reception Buffer Size	2	This size includes all possibly needed space for SSP /SSP/ transport headers, security, concatenation information and so on.	M

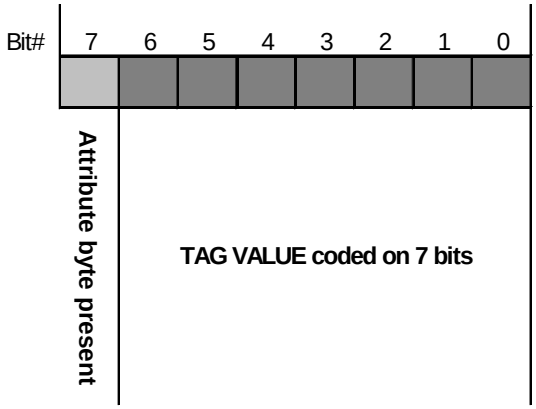
[1)] NOTE: PROVIDE LOCAL INFO with the appropriate command qualifier shall be generated each time when Location information, IMEI or NMR are requested. In case of error in general result, empty value shall be written to the environment variable.

DEPRECATED IDs
0x04
0x08
0x41
0x44

0x60 – 0x7F are reserved for proprietary use. Other variable Ids are RFU. Deprecated Ids must not be re-used in future.

6 GENERIC TL[A]V FORMAT

The TL[A]V (standing for Tag Length [Attributes] Value) is the basic data structure element. The tag byte contains a seven-bit tag value and an attribute byte-present bit in the MSB. If the attribute byte-present bit is set then the leading byte(s) in the value field contain attribute information for the element identified by the tag. The value part of a TL[A]V can itself contain TL[A]Vs.

<table>
<tr><th>LENGTH</th><th>VALUE</th><th>DESCRIPTION</th><th>M/O</th></tr>
<tr><td>1</td><td>V</td><td>S@T bytecode tag (+ attributes presence flag)
 

b7 - Attribute byte present:
0: Attribute byte not present as first byte of value
1: Attribute byte present as first byte of value</td><td>M</td></tr>
<tr><td>1-3</td><td>A + B</td><td>Length of subsequent data including the attribute bytes in BER-TLV format /ISO/IEC 7816-6/:
<table>
<tr><th>Length</th><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th></tr>
<tr><td>0-127</td><td>length</td><td>not present</td><td>not present</td></tr>
<tr><td>128-255</td><td>0x81</td><td>length</td><td>not present</td></tr>
<tr><td>256-65535</td><td>0x82</td><td colspan="2">length</td></tr>
</table></td><td>M</td></tr>
<tr><td>A</td><td>V</td><td>Attribute bytes (see 6.1)</td><td>O</td></tr>
<tr><td>B</td><td>V</td><td>Value</td><td>O</td></tr>
</table>

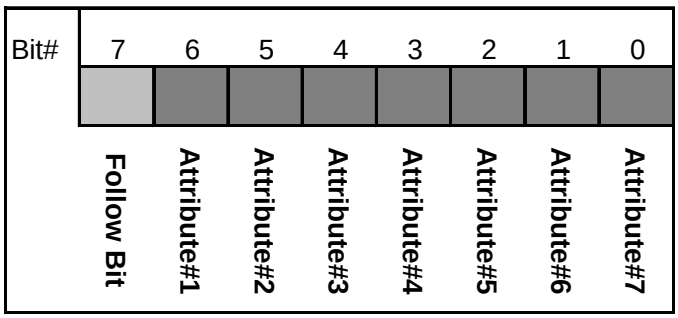
6.1 Coding of the attribute bytes

Every TL[A]V can have one or more attributes bytes if indicated by the attribute byte present bit of the tag byte. The coding of an attribute byte is shown below. Attributes provided in the attribute byte shall be related to the belonging TL[A]V. The meaning of the attributes of a TL[A]V is TL[A]V specific and specified in the TL[A]V descriptions.

An attribute given in an attribute byte can consist of a single bit or a combination of consecutive bits forming an attribute value.

The default value of an attribute value or an attribute bit within an attribute byte is always 0x00. The 0x00 value of an attribute shall be used by the S@T browser, if the attribute is not available in the TL[A]V.

Whenever the attributes for a tag require more than 7 bits within an attribute byte, the number of attribute bytes will be extended. The extension of the attribute byte shall be indicated by the MSB of the attribute byte, which is called the follow bit.

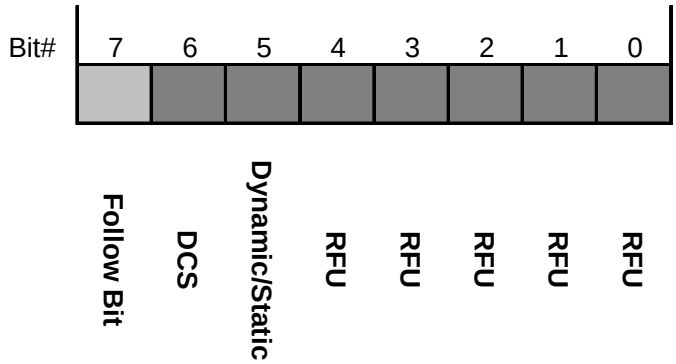


NOTE: RFU bits in an attribute shall be ignored by the S@T browser.

7 BYTECODE STRUCTURE

7.1 Deck

Deck is the smallest unit that can be loaded into the S@T browser for execution. Deck is referenced by deck Id and contains one or more cards.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x01 | 0x81	Deck tag	M
1-3	A + B + C + D + E1 +…+ En	Length of subsequent data (length coding in BER-TLV format)	M
A (0 -1)	V	Attribute:  **b6 – DCS:** 0: SMS 7bit unpacked 1: UCS2 **b5 - Dynamic /Static:** 0: Static, deck shall be cached by the S@T browser 1: Dynamic, deck shall not be cached by the S@T browser	O
B	Deck Id TL[A]V	Deck Id	M
C	Text Element Table TL[A]V	Text element table	O
D	Card Template TL[A]V	Card Template	O
E1	Card TL[A]V	Card 1	M
…	..	….	O
En	Card TL[A]V	Card N	O

NOTE: Unknown TL[A]Vs inside Deck TL[A]V shall be ignored and skipped by the browser.

7.1.1 Deck Id

The content of this TL[A]V is a sequence of bytes to uniquely identify the deck.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x02	Deck Id tag	M
1	A	Deck Id Length (< 32 bytes)	M
A	V	Deck Id	M

7.1.2 Text Element Table

The text element table contains a list of deck text elements that can be referenced during the deck execution.

The text element structure is LV (length coded in 1 byte). The DCS of these elements is inherited from the deck level.

The first text element in the Text element table shall be identified by the variable reference 0xC0, the next with 0xC1 and so on. Text element table can contain up to 64 elements.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x04	Text Element Table tag	M
1-3	n +A1 + … + An	Length of subsequent data (length coded in BER-TLV)	M
1	A1 (0x00 to 0xFF)	Text element 1 length	O
A1	V	Text element 1 value	O
…	…	….	O
1	An (0x00 to 0xFF)	Text element N length	O
An	V	Text element N value	O

7.1.3 Card Template

Card Template contains command sequence that shall be executed before the first command on each card of a deck if DoNotUseTemplate flag is not set in card attribute.

For example Card Template can be used to change contextual menu temporary within the deck context so that each card of the deck inherits Manage Contextual Menu Item commands from this template.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x07	Card Template tag	M
1-3	A1 + … + An	Length of subsequent data (length coded in BER-TLV)	M
A1	TL[A]V	S@T bytecode command 1	M
…	…	….	O
An	TL[A]V	S@T bytecode command N	O

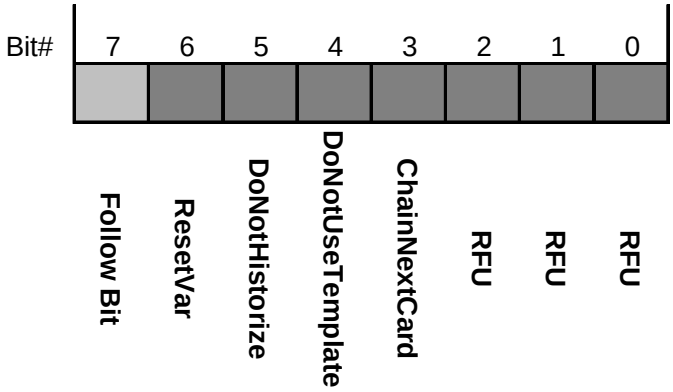
NOTE: The following commands can be included to Card Template:

- Manage Contextual Menu Item;
- Init Variables;
- Concatenate;
- Extract.

Other commands or unknown TL[A]Vs inside Card Template TL[A]V shall be ignored and skipped by the browser.

7.2 Card

Card is the smallest navigation unit that can be referenced using card Id. A card contains S@T bytecode commands. Navigation to a card means to start executing its first command.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x05 | 0x85	Card tag	M
1-3	A + B + C1 + … + Cn	Length of subsequent data (length coded in BER-TLV)	M
A (0 -1)	V	Attribute:  **b6 – ResetVar:** 0: keep temporal variables 1 : reset (i.e. remove) all the temporary variables before executing the first command in the card **b5 – DoNotHistorize:** 0: historize the card when jump to another card 1: do not historize the card **b4 – DoNotUseTemplate** : 0: apply the card template before executing the first command in the card 1: do not use card template **b3 – ChainNextCard :** 0: the S@T browser exits (goes to the idle state) when the last command in the card was executed and no branching was done. 1: automatically start the next card of the deck when the last command in the card was executed and no branching was done.	O
B	Card Id TL[A]V	Card Id	O
C1	TL[A]V	S@T bytecode command 1	O
…	…	…	O
Cn	TL[A]V	S@T bytecode command N	O

NOTE: If ChainNextCard flag is set in the attribute of the last card, the S@T browser exits when the last command in the card was executed and no branching was done.

NOTE: Unknown TL[A]Vs inside Card TL[A]V shall be ignored and skipped by the browser.

7.2.1 Card Id

The content of this TL[A]V is a sequence of bytes to identify the card.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x06	Card Id tag	M
1	A	Card Id length (< 32 bytes)	M
A	V	Card Id	M

7.3 Referencing Decks and Cards

Address reference is used to refer to a card in the current deck or in another deck.

Address reference coding is the following:

<Address Reference> := [(<Deck URL> | <Deck Id>)] [#<Card Id >]

Where:

- Deck URL: used to identify online deck and generally sent to the S@T gateway in a pull request (see /Pull/).
- Deck Id: value part of Deck Id TL[A]V; used to reference a deck;
- Card Id: value part of Card Id TL[A]V; used to reference a card within the deck.

NOTE: If Card Id is only present, this denotes a card in the current deck.

NOTE: If Card Id is not present then deck execution shall be started from the first card. If present then deck execution shall be started from the card with this Card Id.

NOTE: '#' is forbidden in Deck URL and Deck Id because it is used as a separator before Card Id.

8 GENERAL TL[A]Vs

8.1 Variable Reference

This TL[A]V is used to refer to a variable.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x08	Variable Reference tag	M
1	1	Variable Id length	M
1	VarId	Variable Id	M

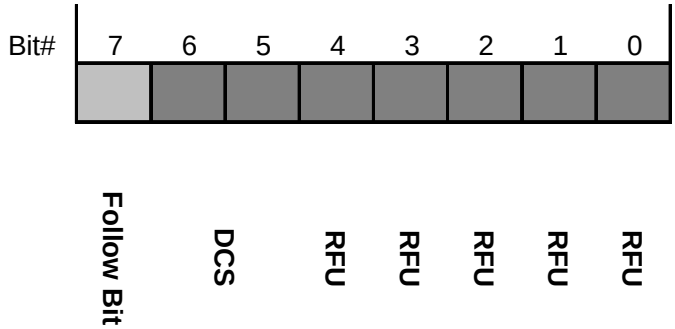
8.2 Variable Reference List

This TL[A]V is used to list a sequence of variables.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x09	Variable Reference List tag	M
1	N	Length (= number of variables); up to 127 variables	M
1	VarId	Variable Id 1	M
…	…	…	O
1	VarId	Variable Id N	O

8.3 Inline Value

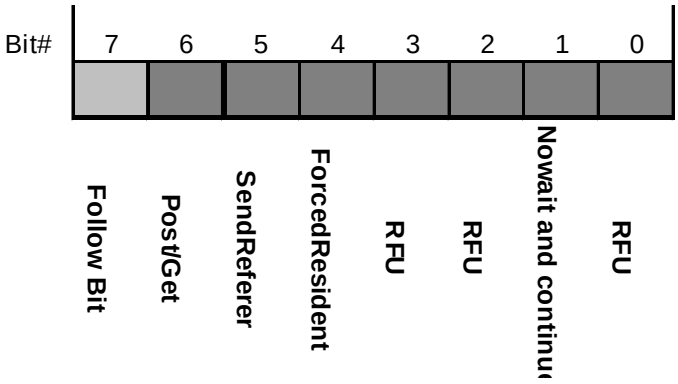
This TL[A]V contains a byte array that can be used as a text or binary data.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0A | 0x8A	Inline Value tag	M
1-2	A + B	Inline value length (length including attribute byte < 255)	M
A (0-1)	V	Attribute:  **b6b5 - DCS:** 00: DCS inherited from the current Deck DCS [1)] 01: SMS 7bit unpacked 10: UCS2 11: Binary	O
B	V	Inline Value content	M

[1)] NOTE: When no current deck is available (e.g. when Inline Value TL[A]V is received by the gateway), DCS '00' means 'Binary'.

8.4 URL Reference

This TL[A]V contains address reference and optional parameters.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0D | 0x8D	URL reference tag	M
1-2	A + B + C1 +…+ Cn	URL reference length (length including attribute byte < 255)	M
A (0-1)	V	Attribute:  **b6 - Post/Get:** 0: GET_REQ is used in pull request 1: POST_REQ is used in pull request **b5 - SendReferer:** 0: do not send SendRerefer 1: send SendReferer **b4 - ForcedResident:** 0: Address reference presents online deck 1: Address reference presents resident deck **b1 – Nowait and continue:** 0: Enter waiting for response state after sending pull request 1: Do not enter waiting for response state and process next bytecode commnad after sending pull request (gateway response on the pull request will not be accepted by the browser)	O
B	Address Reference TL[A]V | Variable Reference TL[A]V	Address reference	M
C1	Parameter TL[A]V | Constant Parameter TL[A]V	Parameter 1	O
…	…	…	O
Cn	Parameter TL[A]V | Constant Parameter TL[A]V	Parameter N	O

NOTE: Address Reference with only card Id results in local navigation inside the current deck. Attribute value and parameters shall be ignored in this case.

NOTE: If ForcedResident flag is set then other attribute bits and parameters shall be ignored.

Before retrieving an online deck, URL reference TL[A]V shall be transformed as described below:

- Variable Reference TL[A]V shall be replaced with Address Reference TL[A]V containing the variable value.
- If Constant Parameter value DCS has to be inherited from the deck DCS, then Constant Parameter value DCS (i.e. attribute in Inline Value TL[A]V) shall be set to the deck DCS.
- All Parameter TL[A]Vs shall be converted to Constant Parameter TL[A]Vs according to the following rules:
  - Constant Parameter value shall contain variable value and inherit its DCS;
  - Parameter name (if any) shall be treated as binary data (i.e. no attribute in Inline Value TL[A]V).

If an online deck has to be retrieved from the gateway, the S@T browser shall generate a pull request (see /Pull/) as described below:

- URL Reference TL[A]V (transformed as described above) shall be added to the browser request.
- If SendReferer flag is set, another URL Reference TL[A]V with the current deck Id shall be added to the browser request.
- Depending on Post/Get flag, browser request shall be wrapped in POST_REQ or GET_REQ (see /SSP/).

8.4.1 Address Reference

This TL[A]V contains address reference.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0E	Address Reference tag	M
1-2	A	Length (< 255)	M
A	V	Address reference (as defined in 7.3) coded in SMS 7bit unpacked	M

8.4.2 Parameter

This TL[A]V contains URL reference parameter.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0C	Parameter tag	M
1-2	1 + A	Length (< 255)	M
1	VarId	Variable containing parameter value	M
A	V	Parameter name	O

8.4.3 Constant Parameter

This TL[A]V contains URL reference constant parameter.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0F	Constant Parameter tag	M
1-2	A + B	Length (< 255)	M
A	Inline Value TL[A]V	Parameter value	M
B	Inline Value TL[A]V	Parameter name	O

8.5 Value Couple

This TL[A]V represents value – value couple.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x11	Couple tag	M
1-3	A + B	Length	M
A	Variable Reference TL[A]V | Inline Value TL[A]V	Value	M
B	Variable Reference TL[A]V | Inline Value TL[A]V	Value	M

8.6 URL Reference Couple

This TL[A]V represents value – URL reference couple.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x11	Couple tag	M
1-3	A + B	Length	M
A	Variable Reference TL[A]V | Inline Value TL[A]V	Value	M
B	URL Reference TL[A]V	URL Reference	M

9 BYTECODE COMMANDS

9.1 Init Variables

Initialize variables with values.

The output variable contains the input value and inherits its DCS.

LENGTH	***VALUE***	***DESCRIPTION***	***M/O***
1	0x20	Init Variable tag	M
1-3	n + A1 + … + An	Length of subsequent data (length coded in BER-TLV)	M
1	VarId	Output variable 1	M
A1	Variable Reference TL[A]V | Inline Value TL[A]V	Input value for variable 1	M
…	…	…	O
1	VarId	Output variable N	O
An	Variable Reference TL[A]V | Inline Value TL[A]V	Input value for variable N	O
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format	
Problem in memory management		Not enough memory to allocate output variable and store the result	
Reference to undefined		Variable not found	

9.2 Init Variable Selected

Generate proactive command SELECT ITEM.

The output variable contains the value of the selected item and inherits its DCS.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x21	Init Variable Selected tag	M
1-3	1 + A + B1 + … + Bn	Length of subsequent data (length coded in BER-TLV)	M
1	VarId	Output variable Id	M
A	Variable Reference TL[A]V | Inline Value TL[A]V	Select Item title	O
B1	Value Couple TL[A]V	Text for item #1 – value for item #1	M
…	…	…	O
Bn	Value Couple TL[A]V	Text for item #n – value for item #n	O
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format or proactive command length more that 255 bytes	
Problem in memory management		Not enough memory to allocate output variable and store the result	
Reference to undefined		Variable not found	
STK use failed		ME returns general result with error on proactive command	

9.3 Get Environment Variable

Get the value of an environment variable.

The output variable contains the value of the environment variable and has binary DCS.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x22	Get Environment Variable tag	M
1	2	Length of subsequent data (length coded in BER-TLV)	M
1	VarId	Output variable Id	M
1	V	Environment variable Id (as defined in 5.6)	M
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format	
Problem in memory management		Not enough memory to allocate output variable and store the result	

NOTE: If an environment variable does not exist, then empty string shall be written to the output variable.

9.4 Concatenate

Concatenate two or more values.

The output variable contains concatenated values and inherits their DCS

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x24	Concatenate Tag	M
1-3	1 + A1 + A2 + … + An	Length of subsequent data (length coded in BER-TLV)	M
1	VarId	Output variable Id	M
A1	Variable Reference TL[A]V | Inline Value TL[A]V	Value 1	M
A2	Variable Reference TL[A]V | Inline Value TL[A]V	Value 2	M
…	…	…	O
An	Variable Reference TL[A]V | Inline Value TL[A]V	Value N	O
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format	
Problem in memory management		Not enough memory to allocate output variable and store the result	
Reference to undefined		Variable not found	
Type mismatch		Values to be concatenated have different DCS [1)]	

[1)] NOTE: Binary DCS matches any DCS. If all variables are binary, the result is binary. If some variables are binary and other variables are SMS or UCS2, the result is SMS or UCS2 accordingly.

9.5 Extract

Extract a substring from the input variable.

The output variable contains a substring of the input variable and inherits its DCS.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x25	Extract Tag	M
1	4	Length of subsequent data	M
1	VarId	Output variable Id	M
1	VarId	Input variable Id	M
1	0 - (Input Value Length – 1)	Start index of the substring in the input variable (position of the character)	M
1	0-255	Substring length (in characters) If start index + substring length exceeds length of the input variable, the length shall be internally reduced to length of input variable length - start Index and no error shall be generated.	M

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format
Problem in memory management	Not enough memory to allocate output variable and store the result
Reference to undefined	Input Variable not found
Out of range	Start index exceeds input variable length – 1

NOTE: If the input variable has DCS UCS2, then the S@T browser shall multiplied start index and substring length by 2.

9.6 Encrypt

Encrypt set of values.

The output variable contains Secure Message TL[A]V (as defined in 9.6.1) and has binary DCS.

Secure message data block contains LV (length coded in 1 byte) sequence with input variable values:

<Length of value 1><Value 1>…<Length of value N><Value N>

LENGTH	***VALUE***	***DESCRIPTION***	***M/O***
1	0x26	Encrypt tag	M
1-3	4 + A	Length of subsequent data (length coded in BER-TLV)	M
1	According to /03.48/	SPI First Octet	M
1	According to /03.48/	KIc	M
1	According to /03.48/	KID	M
1	VarId	Output variable Id	M
A	Variable Reference List TL[A]V	Input values	M
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format	
Problem in memory management		Not enough memory to allocate output variable and store the result	
Reference to undefined		Variable not found	
Security problem		Algorithm not supported or key not found	

9.6.1 Secure Message

This structure is used in Encrypt/Decrypt command.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x10	Secure Message tag	M
1-2	5 + B or 6 + A + B	Length of subsequent data (length < 255)	M
1	According to /03.48/	SPI First Octet[3)]	M
1	According to /03.48/	KIc[3)]	M
1	According to /03.48/	KID[3)]	M
1	0 - 7	Padding Counter	M
1	A	Length of MAC	O
A	V [2)]	MAC (Cryptographic Checksum) (encrypted or plain) MAC is calculated over Plain Data Block padded with 0x00. If no ciphering is applied, padding shall be removed and padding counter shall be set to 0x00 after MAC calculation.	O
1	B	Length of the following data (length coded in 1 byte) NOTE: length shall be multiple of 8 if ciphering is applied	M
B	V [1) 2)]	Data block (encrypted or plain) Plain Data Block contains LV (length coded in 1 byte) sequence optionally padded with 0x00 (if ciphering is applied)	M

[1)] NOTE: These fields shall be included to MAC (Cryptographic Checksum) calculation.

[2)] NOTE: These fields shall be ciphered.

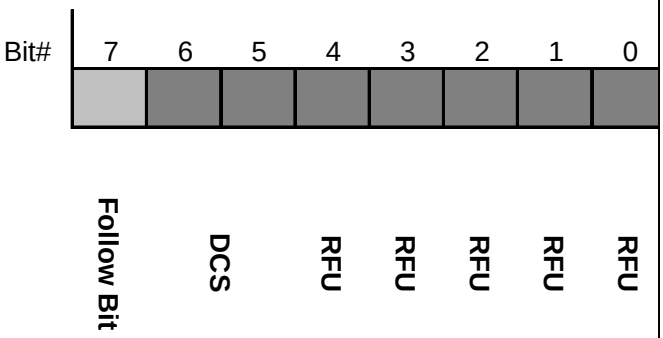
[3)] NOTE: The following algorithms shall be supported by the S@T browser:

SPI First Octet	*KIc*	*KID*	*DESCRIPTION*
0x00	0x00	0x00	No security applied
0x02	0x00	0xX5	Triple DES Cryptographic Checksum (8-byted MAC)
0x04	0xX5	0x00	Triple DES Ciphering
0x06	0xX5	0xY5	Triple DES Cryptographic Checksum (8-byted MAC) + Triple DES Ciphering

9.7 Decrypt

Decrypt data block.

The output variables contain values from LV sequence and have DCS according to DCS flag.

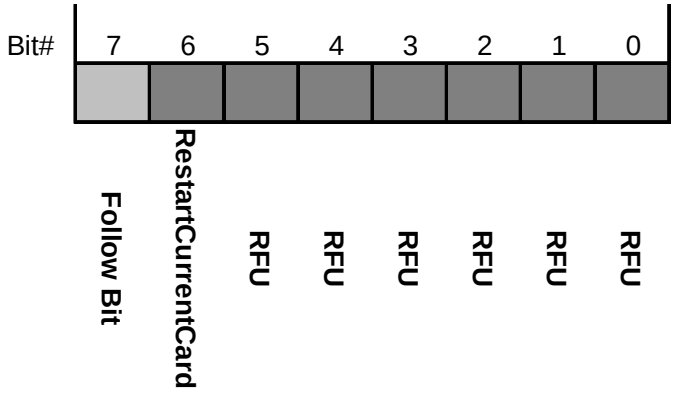
LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x27 | 0xA7	Decrypt tag	M
1-3	A + B + C	Length of subsequent data (length coded in BER-TLV)	M
A	V	Attribute:  **b6b5 - DCS:** 01: SMS 7bit unpacked 10: UCS2 00, 11: Binary	
B	Variable Reference TL[A]V | Inline Value TL[A]V	Secure message (as defined in 9.6.1)	M
C	Variable Reference List TL[A]V	Output variables	M

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format or bad secure message format (e.g. ciphering is applied but data block length is not multiple of 8)
Problem in memory management	Not enough memory to allocate output variables and store the result
Reference to undefined	Variable not found
Security problem	Algorithm not supported or key not found or MAC verification failed

NOTE: If number of output variables is more than number of LVs in data block then remain variables shall be initialized with empty values. If number of LVs in Data Block is more than number of output variables then remain LV shall be skipped silently.

9.8 Go Back

Perform history navigation action ‘go back’ or restart the current card (no history impact).

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x28 | 0xA8	Go Back tag	M
1	0-1	Length of subsequent data (length coded in BER-TLV)	M
1	V	Attribute:  **b6 - RestartCurrentCard**: 0: history navigation action ‘go back’(as defined in 4.4) 1: restart the current card (no history impact)	O

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format

NOTE: When RestartCurrentCard flag is not set and history is empty, the S@T browser shall exit silently.

9.9 Go Selected

Go Selected can have one of the following formats:

9.9.1 Go Selected (Select Item)

Generate proactive command SELECT ITEM and retrieve URL reference depending on the selected item.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x29	Go Selected tag	M
1-3	A + B1 + … + Bn	Length of subsequent data (length coded in BER-TLV)	M
A	Variable Reference TL[A]V | Inline Value TL[A]V	Select Item Title	O
B1	URL Reference Couple TL[A]V	Text for item #1 – URL reference for item #1	M
…	…	…	O
Bn	URL Reference Couple TL[A]V	Text for item #n – URL reference for item #n	O
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format or proactive command length more that 255 bytes	
Reference to undefined		Variable not found	
STK use failed		ME returns general result with error on proactive command	
Jump to undefined		Address reference not found	

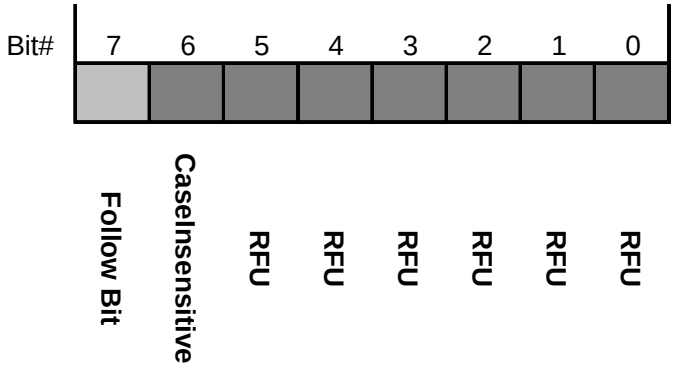
9.9.2 Go Selected (Jump to URL)

Retrieve URL reference

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x29	Go Selected tag	M
1-3	A	Length of subsequent data (length coded in BER-TLV)	M
A	URL Reference TL[A]V	URL reference	M
ERROR CODES		***DESCRIPTION***	
Syntax error		Bad command format	
Reference to undefined		Variable not found	
Jump to undefined		Address reference not found	

9.10 Switch Case on Variable

Compare the input variable with the list of values and retrieve URL reference depending on match.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x2A | 0xAA	Tag	M
1-3	1 + A+ B1 +…+ Bn + C	Length of subsequent data (length coded in BER-TLV)	M
A (0 - 1)	V	Attribute:  **b6 - CaseInsensitive:** 0: case sensitive comparison 1: case insensitive comparison	O
1	VarId	Input variable Id	M
B1	URL Reference Couple TL[A]V	Value #1 – URL reference if the input variable matches value #1	M
…	…	…	M
Bn	URL Reference Couple TL[A]V	Value #n – URL reference if the input variable matches value #n	M
C	URL Reference TL[A]V	Default URL reference if no match is found	O

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format
Reference to undefined	Input variable or variable with given Id does not exist
Jump to undefined	Address reference is not found

NOTE: the CaseInsensitive flag is taken into account only for DCS SMS 7bit unpacked and only for range 0x41-0x5E/0x61-0x7E.

NOTE: When no match is found, and no default URL reference is present, the S@T browser shall continue deck execution.

NOTE: Binary DCS matches any DCS

9.11 Exit

Exit the S@T browser.

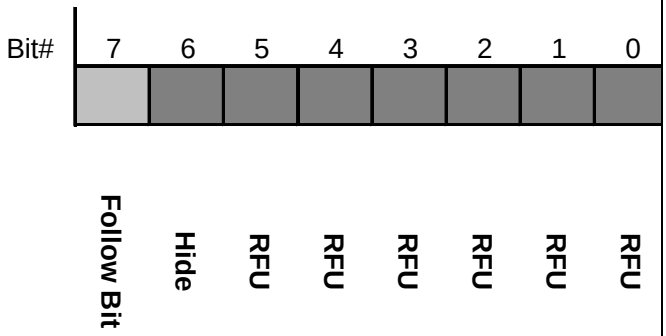
LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x2B	Exit Tag	M
1	0	Length of subsequent data (length coded in BER-TLV)	M

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format

9.12 Manage Contextual Menu Item

Display/hide system contextual menu item or add/update/hide application specific items.

This command can be used inside Card Template TL[A]V or inside Card TL[A]V to modify temporary contextual menu in the deck or card context.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x2C | 0xAC	Manage Contextual Menu Item tag	M
1-3	A + 1 + B	Length of subsequent data (length coded in BER-TLV)	M
A(0-1)	V	Attribute:  **b6 - Hide:** 0 : display system contextual menu item or add/update application specific item 1: hide contextual menu item	O
1	V	Contextual menu item Id (as defined in 4.6.1)	M
B	URL Reference Couple TL[A]V	Title and URL reference for application specific item	O

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format
Reference to undefined	Contextual menu item not found: system item for operation “display/hide”; application specific item for operation “hide”.
Problem in memory management	Too many items allocated

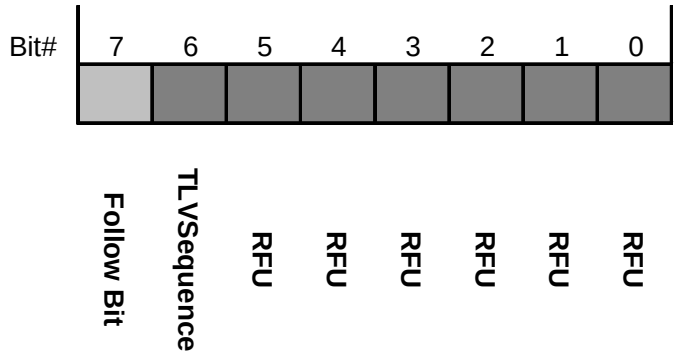
NOTE: URL Reference Couple TL[A]V shall be ignored when manage system contextual menu item.

NOTE: If the S@T browser does not support contextual menu it shall skip this TL[A]V.

9.13 Execute STK Command

Generate a proactive command.

After ME returns terminal response on the generated proactive command, data from the terminal response shall be written to the output variable (if present) depending on the attribute value. Only proactive command specific TLVs (TLVs following Result TLV) shall be used for the output generation.

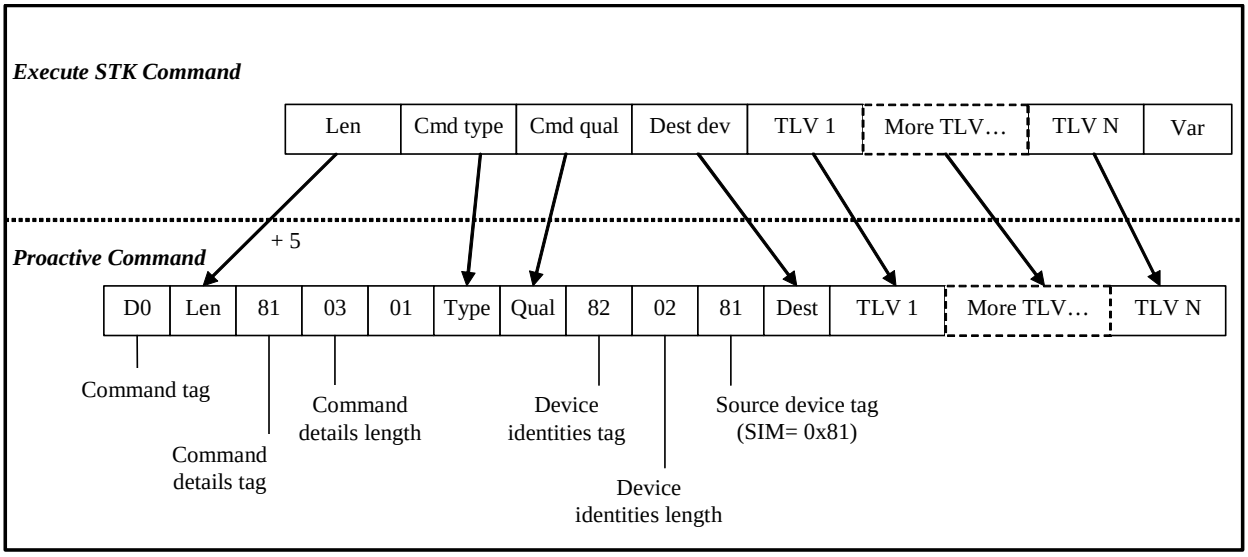
LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x2D | 0xAD	Execute STK Command tag	M
1-3	A + 3 + B + C	Length of subsequent data (length coded in BER-TLV)	M
A (0 - 1)	V	Attribute:  **b6 – TLVSequence:** 0: the Output variable contains value of the 1st specific TLV from the terminal response. [1)] 1: the Output variable contains sequence of specific TLVs from the Terminal Response (copied as is) and has binary DCS.	O
1	According to /51.014/	Command type	M
1	According to /51.014/	Command qualifier	M
1	According to /51.014/	Destination device	M
B	According to /51.014/	STK TLVs for proactive command excluding Command Details and Device Identities TLVs. Variable Substitution (as defined in 9.13.1) can be used in STK TLV coding.	O
C (0 - 1)	VarId	Output variable Id	O

ERROR CODES	*DESCRIPTION*
Syntax error	Bad command format or proactive command length more that 255 bytes
Problem in memory management	Not enough memory to allocate output variable and store the result
Reference to undefined	Variable with given Id (inside STK TLV) does not exist
STK use failed	ME returns general result with error on proactive command

[1)] NOTE: If TLVSequence flag is not set and TLV value has DCS byte (e.g. Text String TLV) then DCS byte shall be removed from the output variable value and used for the output variable DCS otherwise the output variable DCS is binary. If DCS is SMS 7bit packed then the browser shall unpack the data and store it in SMS 7bit unpacked format.

NOTE: If there is no TLV following Result TLV in terminal response then the output variable shall have empty value.

Mapping into Proactive Command:



9.13.1 Variable Substitution

Variable substitution can take place in the STK TLVs. The variable substitution indicator is used to indicate that the next byte is a variable Id.

STK TLV with value:

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	According to /51.014/	Tag	M
1-3	A	Length of subsequent data (length coded in BER-TLV)	M
A	According to /51.014/	Value	O

STK TLV with variable Id:

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	According to /51.014/	Tag	M
1	0xFF	Variable substitution indicator	M
1	VarId	Variable Id	M

The following procedure shall be applied to STK TLV with the variable substitution indicator:

- The variable Id is replaced by the current content of the variable.
- If STK TLV implies DCS byte (Text String; Default Text; USSD String) then DCS byte is added to the current value according to the variable DCS. For USSD string the browser shall pack SMS 7bit data and set DCS byte to 0x0F. For alpha fields (Alpha Identifier TLV) the browser shall set the first byte for UCS2 text accordingly. Binary variables shall be treated as SMS 7bit unpacked data in this case.
- The variable substitution indicator is replaced by the length of the current value. The length is coded in BER-TLV (1-3 bytes).

9.14 Execute Plug-in

This command is used to invoke S@T browser plug-in from S@T bytecode. Plug-in is unequally identified by plug-in name. For each plug-in usage protocol is defined: coding of input and output parameters.

At the end a plug-in writes its output to the output variables and set their DCS.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x2E	Execute Plug-in tag	M
1-3	2 + A + B	Length of subsequent data (length coded in BER-TLV)	M
2	V	Plug-in name: 1st byte : Vendor Id or 0xFF (denotes S@T standard plug-in) 2nd byte : Plugin Id	M
A	Input List TL[A]V	Input parameters	O
B	Variable Reference List TL[A]V	Output parameters	O
ERROR CODES		***DESCRIPTION***	
Reference to undefined		Variable not found	
Jump to undefined		Plug-in not found	
Problem in memory management		Not enough memory to allocate output variable and store the result	
Syntax Error		Bad command format or bad input parameters number and coding	
Execution Failure		Plug-in internal error	

NOTE: Plug-in input parameters number and coding shall be verified by the plug-in.

9.14.1 Input List

This TL[A]V contains input parameters for a plug-in.

LENGTH	*VALUE*	*DESCRIPTION*	*M/O*
1	0x0B	Input List tag	M
1-3	A1+…+An	Length of subsequent data (length coded in BER-TLV)	M
A1	Variable Reference TL[A]V | Inline Value TL[A]V	Input parameter 1	M
…	…	…	O
An	Variable Reference TL[A]V | Inline Value TL[A]V	Input parameter N	O

10 ERROR MANAGEMENT

For the indication of errors occurring during bytecode processing error codes listed in the following table are defined. This information can be accessed using the Error Code environment variable.

Table of predefined Error Codes:

Type of error	***Coding***
Communication problem	0x6F01
Syntax error	0x6F02
STK use failed	0x6F03
Jump to undefined	0x6F04
Problem in memory management	0x6F05
Security problem	0x6F06
Reference to undefined	0x6F07
Type mismatch	0x6F09
Out of range	0x6F0A
Unknown tag	0x6F0D
URL not found	0x6F0E
Execute failed	0x6F0F
General unspecific error	0x6FFF

In the case no error occurs the value 0x0000 is assigned to the Error Code.

11 LIST OF TAGS

Since bit 7 is reserved for attribute indication the tag values can be in range [0x00..0x7F]

TAG	*VALUE*
Bytecode structure and general TL[A]Vs	
Deck tag	0x01
Deck Id tag	0x02
Text Element Table tag	0x04
Card tag	0x05
Card Id tag	0x06
Card Template tag	0x07
Variable Reference tag	0x08
Variable Reference List tag	0x09
Inline Value tag	0x0A
Input List tag	0x0B
Parameter tag	0x0C
URL Reference tag	0x0D
Address Reference tag	0x0E
Constant Parameter tag	0x0F
Secure Message tag	0x10
Couple tag	0x11
Bytecode commands	
Init Variable tag	0x20
Init Variable Selected tag	0x21
Get Environment Variable tag	0x22
Concatenate tag	0x24
Extract tag	0x25
Encrypt tag	0x26
Decrypt tag	0x27
Go Back tag	0x28
Go Selected tag	0x29
Switch Case on Variable tag	0x2A
Exit tag	0x2B

Manage Contextual Menu Item tag	0x2C
Execute STK Command tag	0x2D
Execute Plug-in tag	0x2E

DEPRECATED TAGs
0x03
0x23

Other tags are RFU. Deprecated tags must not be re-used in future.

12 Annex A [normative]: OPTIONAL FEATURES

12.1 Contextual Menu support

The S@T browser not supporting contextual menu shall behave as if default contextual menu is configured (as defined in 4.6.3) and skip silently Manage Contextual Menu Item commands.

12.2 Permanent Variable support

The S@T browser not supporting permanent variables shall behave as if no buffer is configured for the permanent variables, i.e. generate ‘Problem in memory management’ error when a permanent variable has to be created or updated and “Reference to undefined” when a permanent variable has to be read.

13 HISTORY

Document history		
Release	Approved by	Comment
1.0.0	SIM Alliance TDG	First internal release
1.0.1	SIM Alliance TDG	Changes after meeting 16 (new and unified error codes, CRs: G+#1, SLB#1)
1.0.2	SIM Alliance TDG	Changes after meeting 17, 18 CRs: 10001, 10002, 10003, 10005, 10008, 10009, 10011, 10012, 10013, 10021, 10022, 10023, 10024, 10026, 10027, 10028, 10029, 10019, 10020, 10031, 10032, 10033, 10034, 10035, 10039, 10041, 10019, 10040 integrated
1.0.3	SIM Alliance TDG	after meeting 18a and 19 CRs: 10042, 10030, 10038, 10044, 10047, 10049, 10050, 10054, 10055, 10056 integrated
1.0.4	SIM Alliance TDG	after meeting 25 CRs: 10084, 10090, 10091, 10096, 10099 integrated
1.0.5	SIM Alliance TDG	after meeting 27 CRs: 10083, 10098, 10106, 10107, 10108, 10112 integrated
1.0.6	SIM Alliance TDG	after meeting 29 CRs: 10121, 10122 integrated
1.0.7	SIM Alliance TDG	Editorial changes for publication
2.0.0	SIM Alliance TDG	Clarifications according to change request forum in section 5.3.7, 6.2.10, 6.2.6, June 2004
3.0.0	SIM Alliance TDG	CRs: 2006-12, 2006-32, 2006-33, 2006-34 integrated
3.0.1	SIM Alliance TDG	CRs: PRISM-May-2007 #002 and PRISM-Dec-2007 #004 integrated.

4.0.0	SIM Alliance TDG	S@T Task Force: Review of Spec: Major editorial modifications for clarification Updated features: - STK generic macro (pack/unpack USSD string; modify output data encapsulation) - Decrypt macro (add DCS management) - Contextual menu management (removed card/operator item. Remove help menu. Remove “Restart current deck”. Remove “Pause/Resume”. Remove “Push activation/deactivation”. Define default configuration.) - URL reference (nowait attribute removed) - Execute Generic macro (remove exit attribute and variable reference list) - Exit macro (remove CleanBuffer attribute and variable reference list) - Reception Buffer Size, Resident Decks Buffer Size, Resident Deck List, Plug-in List (Environment variables) - Service Permanent Store is replaced by permanent variables New features: - Binary DCS (may impact all commands managing variables) Removed features: - Set help command - Pause/Resume session - Clean up variable list - Operator name, User type (Environment variables)

13.1 Annex B [informative]: List of CRs

CR Number	CR Identifier	Subject	Document Reference	Status / Meeting No.
10001	SLB-WG1-APRIL-2000#1	REMOVE THE START CARD REFERENCE	S@T 1.00 V1.0.1	Accepted #17
10002	SLB-WG1-APRIL-2000#2	CHANGE URL FORCED LOCAL TO FORCED RESIDENT	S@T 1.00 V1.0.1	Accepted #17
10003	SLB-WG1-APRIL-2000#3	ADD A TAG FOR SECMSG	S@T 1.00 V1.0.1	Accepted #17
10005	G&D-1-04-2000#1	"#"-CHARACTER IN LONG DECK NAMES	S@T 1.00 V1.0.1	Accepted #17
10008	G&D-1-04-2000#4	Reset Help String	S@T 1.00 V1.0.1	Accepted #17
10009	G&D-1-04-2000#5	Extract with Length	S@T 1.00 V1.0.1	Accepted #17
10011	ORGA-WG1-APRIL-2000#1	Additional Error – Unknown tag	S@T 1.00 V1.0.1	Accepted #17
10012	GEMPLUS-WG1-MAY-2000#1	ADD AN ITEM FOR PAUSE IN CONTEXTUAL MENU	S@T 1.00 V1.0.1	Accepted #17
10013	GEMPLUS-WG1-MAY-2000#2	REMOVE A SENTENCE	S@T 1.00 V1.0.1	Accepted #17
10019	GEMPLUS-WG1-MAY-2000#17.6	ENCAPSULATION OF COUPLE STRUCTURES	S@T 1.00 V1.0.1	Accepted #18
10020	GEMPLUS-WG1-MAY-2000#17.7	CARD ONLY ATTRIBUTE IN URL REFERENCE	S@T 1.00 V1.0.1	Accepted #18
10021	G&D-WG1-MAY-2000#1	Editorial change for SET Help	S@T 1.00 V1.0.1	Accepted #18
10022	G&D-WG1-MAY-2000#2	Editorial change for Deck	S@T 1.00 V1.0.1	Accepted #18
10023	G&D-WG1-MAY-2000#3	Editorial change for init_variables_Selected and go_selected	S@T 1.00 V1.0.1	Accepted #18
10024	G&D-WG1-MAY-2000#4	Editorial change for variable ref list	S@T 1.00 V1.0.1	Accepted #18
10026	G&D-WG1-MAY-2000#6	ICCID Mandatory	S@T 1.00 V1.0.1	Accepted #18
10027	G&D-WG1-MAY-2000#7	Clarify coding of Location Information	S@T 1.00 V1.0.1	Accepted #18
10028	G&D-WG1-	Remove Substitution, Output, security	S@T 1.00	Accepted #18

	MAY-2000#8		V1.0.1	
10029	GEMPLUS-WG1-MAY-2000#18.1	Contents of the Length byte in TL[A]V struct	S@T 1.00 V1.0.1	Accepted #18
10030	GEMPLUS-WG1-MAY-2000#18.2	Editorial change for init_variables_Selected and go_selected	S@T 1.00 V1.0.1	Pending
10031	GEMPLUS-WG1-MAY-2000#18.3	RFU indication after Follow bit in Attribute field	S@T 1.00 V1.0.1	Accepted #18
10032	ORGA-WG1-MAY-2000#18.1	Term “result string” in STK Generic Macro	S@T 1.00 V1.0.1	Accepted #18
10033	ORGA-WG1-MAY-2000#18.2	Browser behaviour on empty variables	S@T 1.00 V1.0.1	Accepted #18
10034	SLB-WG1-MAY-2000#2	ORDER OF SIGNATURE VS ENCRYPTION IN ENCRYPT/DECRYPT MACROS	S@T 1.00 V1.0.1	Accepted #18
10035	SLB-WG1-MAY-2000#1	PRECISE USE OF FORCE RESIDENT FLAG IN URL REFERENCE	S@T 1.00 V1.0.1	Accepted #18
10038	ORGA-WG1-MAY-2000#18.4	ADDRESS REFERENCE IN CARD ID ELEMENT OF A CARD ELEMENT	S@T 1.00 V1.0.1	Accepted #18a
10039	GEMPLUS-WG1-MAY-2000#18.8	OUTPUT VARIABLE AS DEFAULT VARIABLE	S@T 1.00 V1.0.1	Accepted #18
10040	GEMPLUS-WG1-MAY-2000#18.5	COUPLE USAGE IN CONTEXTUAL MENUS	S@T 1.00 V1.0.1	Accepted #18
10041	GEMPLUS-WG1-MAY-2000#18.6	GET ENV MACRO PARAMETERS ORDER	S@T 1.00 V1.0.1	Accepted #18
10042	GEMPLUS-WG1-MAY-2000#18.7	OPTIONAL / MANDATORY FEATURES	S@T 1.00 V1.0.1	Accepted #18
10044	GEMPLUS-WG1-MAY-2000#18.10	EXTRACT MACRO PARAMETERS ORDER	S@T 1.00 V1.0.1	Accepted #18a
10047	ORGA-WG1-JUNE-2000#18a.1	ADD AN OPTIONAL TYPE INFORMATION IN THE INLINE VALUE STRUCTURE	S@T 1.00 V1.0.2	Accepted #18a
10049	GEMPLUS-Thread1-June-2000#18a.1	Standard Plug-In Definition (Phone Number Converter)	S@T 1.00 V1.0.2	Accepted #18a
10050	GEMPLUS-Thread1-June-2000#18a.2	Change the Default Visibility of the ‘Go To Home Page’ Item in Contextual Menu to True	S@T 1.00 V1.0.2	Accepted #18a
10054	GIESECKE & DEVRIENT-WG1-JUNE28-2000#1	Editorial change for dynamic / static	S@T 1.00 V1.0.3	Accepted #19
10055	SCHLUMBERGER-WG1-JUNE-2000#19.1	ERROR: Change of tag value for byte code	S@T 1.00 V1.0.3	Accepted #19

10056	GIESECKE & DEVRIENT-WG1-JUNE28-2000#2	Clarification and changes for crypto macros	S@T 1.00 V1.0.3	Accepted #19
10084	SCHLUMBERGER-WG1-OCTOBER-2000#1	Clarification of 'go back' macro	S@T 1.00 V1.0.4	Accepted #22
10090	ORGA-WG1-OCTOBER-2000#22.1	Change Range for Extract Length	S@T 1.00 V1.0.4	Accepted #25
10091	ORGA-WG1-OCTOBER-2000#22.2	Consideration of DCS for Extract Length and Startindex	S@T 1.00 V1.0.4	Accepted #25
10096	SCHLUMBERGER-WG1-JANUARY-2001#1	DCS for 'SMS Alphabet'	S@T 1.00 V1.0.4	Accepted #25
10099	GEMPLUS-WG1-JAN-2001#1	SecMsg Format	S@T 1.00 V1.0.4	Accepted #25
10083	SCHLUMBERGER-WG1-SEPTEMBER 2000#5	PAUSE ITEM IN CONTEXTUAL MENU OPTIONAL	S@T 01.00 V1.0.5	Accepted #26
10098	GEMPLUS-Feb-2001#26-0	Standard Plug-in definition (COMPUTE VALUE LENGTH)	S@T 1.00 V1.0.5	Accepted by eMail 05.03.01
10106	SCHLUMBERGER-WG1-MAR-2001#2	clarification of "SET_HELP" macro	S@T 01.00 V1.0.5	Accepted by eMail 05.03.01
10107	10107 Gemplus-WG1-mar-2001#1	VARIABLE/TEXT ELEMENT length	S@T 01.00 V1.0.5	Accepted by eMail 05.03.01
10108	Gemplus-WG1-mar-2001#2	Enciphered or ClearText DataBlock field in Sec msg Tag	S@T 01.00 V1.0.5	Accepted by eMail 05.03.01
10112	Gemplus-WG1-mar-2001#3	Optional MAC Element in SecMsg Tag	S@T 01.00 V1.0.5	Accepted #27
10121	Gemplus-WG1-MAY-2001#29	Push Support Indication in Browser Profile	S@T 01.00 V1.0.6	Accepted #29
10122	Gemplus-WG1-JUNE-2001#1	System Item in Contextual Menu for Push Activation/Deactivation	S@T 01.00 V1.0.6	Accepted by eMail 21.06.01
2004-009	CR_Axalto_S@T_Number009	6.2.6 EDITORIAL CHANGE IN ENCRYPT DECRYPT MACRO	S@T 01.00 V2.0.0	Accepted by email voting Feb 2004
2004-011	CR_Axalto_S@T_Number011	Replace Schlumberger by Axalto	S@T 01.00 V2.0.0	Accepted by email voting Feb 2004
2004-023	CRG&D01_01.00_SBC_5.3.7-S@T_Specs2003	5.3.7 ALLOWED MACROS IN TEMPLTES	S@T 01.00 V2.0.0	Accepted by email voting Feb 2004

2004-026	CRG&D04_01.00_SBC_6.2.10-S@T_Specs2003	6.2.10 DEFINITION OF FUNCTIONALITY "EXIT"	S@T 01.00 V2.0.0	Accepted by email voting Feb 2004
2006-032	CR_GEMPLUS_004_IMEI_v002	10.6 IMEI environment variable	S@T 01.00 V3.0.0	Accepted by email voting Feb 2006
2006-033	CR_GEMPLUS_005_NOWAIT_v002	5.5.7 Extend the URL reference macro with nowait attributes	S@T 01.00 V3.0.0	Accepted by email voting Feb 2006
2006-034	CR_GEMPLUS_006_Plug-insvartype_v002	6.2.10 Plug-ins Variable Types	S@T 01.00 V3.0.0	Accepted by email voting Feb 2006
2006-012	CR_AXALTO December 2004 #015	Administrative Plug-in Management cmd	S@T 01.00 V3.0.0	Accepted by email voting Feb 2006
	PRISM-May-2007 #002	Clarifications for certification	S@T 1.00 V3.0.1	Accepted at WG meeting Jan 2008
	PRISM-Dec-2007 #004	Clarifications for certification	S@T 1.00 V3.0.1	Accepted at WG meeting Jan 2008