

# S@T 01.21 v2.0.0 (Release 2004)

S@T Administrative Commands

Published by  **simalliance** now Trusted Connectivity Alliance

Copyright © 2004 Trusted Connectivity Alliance Ltd



# 1 TABLE OF CONTENTS

<b>1</b>	<b>TABLE OF CONTENTS</b>	<b>2</b>
<b>2</b>	<b>TERMINOLOGY</b>	<b>4</b>
2.1.1	Notation	4
2.1.2	Abbreviations	4
<b>3</b>	<b>LIST OF DOCUMENTS</b>	<b>4</b>
<b>4</b>	<b>OVERVIEW</b>	<b>5</b>
<b>5</b>	<b>DEFINITION OF NEEDS</b>	<b>5</b>
<b>6</b>	<b>GENERIC SIMPLE-TL[A]V FORMAT</b>	<b>5</b>
<b>7</b>	<b>USE OF ADMINISTRATIVE COMMANDS</b>	<b>6</b>
7.1	GATEWAY COMMAND	6
7.2	BROWSER RESPONSE	6
7.3	SECURITY	7
<b>8</b>	<b>ADMINISTRATIVE COMMANDS</b>	<b>8</b>
<b>8.1</b>	<b>RESIDENT DECKS</b>	<b>8</b>
8.1.1	Introduction	8
8.1.2	Install Deck	8
8.1.3	Uninstall Deck	9
<b>8.2</b>	<b>SERVICE PERMANENT STORE</b>	<b>9</b>
8.2.1	Introduction	9
8.2.2	SPS Allocate	10
8.2.3	Allocate Permanent Variable	12
8.2.4	Set Permanent Variable	12
8.2.5	Get Permanent Variable	13
<b>8.3</b>	<b>ENVIRONMENT VARIABLES</b>	<b>14</b>
8.3.1	Introduction	14
8.3.2	Admin Set Env	14
8.3.3	Admin Get Env	15
<b>8.4</b>	<b>CONTEXTUAL MENUS</b>	<b>16</b>
8.4.1	Admin Manage Contextual Menu Item	16
<b>9</b>	<b>CONNECTION PROCEDURE FOR AN ADMINISTRATION SESSION</b>	<b>18</b>
<b>10</b>	<b>ERROR MANAGEMENT</b>	<b>20</b>
10.1	Error indication	20
<b>11</b>	<b>LIST OF COMMANDS TAGS</b>	<b>21</b>
<b>12</b>	<b>LIST OF VARIABLE IDENTIFIERS</b>	<b>22</b>
12.1	Resident Decks Buffer Size	22
12.2	Reception Buffer Size	22
12.3	List of Resident Decks	22
12.4	List of Executable Elements	23
12.5	Gateway Address	23



12.6 Variable identifier Coding.....24

13 Annex : *OPTIONAL FEATURES* ..... 24

14 *History* ..... 25

14.1 Annex: LIST OF CHANGE REQUESTS [informative].....26



## 2 TERMINOLOGY

### 2.1.1 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.

### 2.1.2 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>SPS</b>	Service Permanent Store

## 3 LIST OF DOCUMENTS

/SBC/ S@T 01.00 : SBC – S@TML Byte Code

/SSP/ S@T 01.20 : SSP –S@T Session Protocol

This document is part of a specification set, please refer to "S@T Release Note" for a comprehensive document list, including document versions.



## 4 OVERVIEW

The S@T Session Protocol (see /SSP/ specifies generic messages (Get\_Req, Post\_Req, Data\_Req, Express\_Data\_Req, Reply, Connect, Disconnect, etc) and related parameters to be used by a S@T Gateway and Browser. This document proposes a way to specialise these generic messages to be used by a S@T gateway-browser couple to perform administrative commands.

## 5 DEFINITION OF NEEDS

This document specifies the following operations concerning the communication between the browser and the gateway:

Information the Browser can send to the gateway:

- Send permanent or environment variable value

Information the gateway can send to the browser:

- Deck permanent storage
- Request to manage environment variable value (Get/Set) and SPS
- Request to manage contextual menus

## 6 GENERIC SIMPLE-TL[A]V FORMAT

Refer to paragraph 4 of /SBC/



## 7 USE OF ADMINISTRATIVE COMMANDS

### 7.1 GATEWAY COMMAND

Administrative commands from the gateway are sent using a DATA\_REQ message. They are stored in the value part of the DATA\_REQ message.

To keep the browser complexity as low as possible and to minimize the code size in the SIM, administrative commands that the browser will have to execute are encoded using the S@TML bytecode format defined in [SBC]. A Request Id is added to identify each command.

<i>LENGTH</i>	<i>VALUE</i>	<i>DESCRIPTION</i>	<i>M/O</i>
1	0x00 to FF	Bytecode Tag (+ Attribute presence flag)	M
1-3	L	Length of Subsequent data	M
		Optional attributes	O
X		Attributes byte	
		Bytecode parameters	M
2	0-65535	Request Id	
L – X – 2		Bytecode parameters List	

Usable administration bytecodes are described in section 8.

NOTE : This specification assumes that the parameters are always in the specified order, in aim to simplify the browser implementation.

### 7.2 BROWSER RESPONSE

When the administrative command has been executed by the S@T browser, it has to respond to the gateway by sending a DATA\_RSP message.

This DATA\_RSP message contains in its value part an administration TL[A]V which encapsulates the error code of the executed administrative command and optionally the output value (for Admin Get Env bytecode for example).

The request id element identifies which request the response corresponds to.

<i>LENGTH</i>	<i>VALUE</i>	<i>DESCRIPTION</i>	<i>M/O</i>
1	Browser Response Tag	Administration tag (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Error Code Element	M
2	0-65535	Error code.	



		Output Element	O(0-n)
1	InlineValueTag	InlineValue Tag	
1-2	X	Length (< 255)	
X	Value	Output of the execution of the administrative command.	

### 7.3 SECURITY

The administration messages must be sent using the Administrative Security Level at the STLS.



## 8 ADMINISTRATIVE COMMANDS

This section describes the browser administrative commands used by the operator to customize the S@T browser.

Customization is possible at several levels:

- Resident decks. As some decks can be preinstalled in the SIM at personalization time, deck update or removal over the air, as well as addition of new decks, must be possible.
- Service Permanent Store. Decks grouped into a service can share permanent variables. The operator must be able to allocate/free space for these variables and to set/get their values.
- Environment variables. They contain information about the browser, the mobile and the subscriber which are necessary for the gateway to generate useable bytecode and/or optimize communication.
- Contextual menus. System menu items can be shown/hidden, and applicative menu items can be added/updated/removed.

### 8.1 RESIDENT DECKS

#### 8.1.1 Introduction

As the browser engine is available in the card, it can be used to execute online applications received over the air or start the execution of decks located permanently in the SIM memory (i.e. : decks put in the card at the personalization time).

These resident decks are managed by the gateway. They are callable by decks received on the fly.

The resident decks can be installed or updated by the INSTALL DECK administrative command, and removed by the UNINSTALL DECK command.

#### 8.1.2 Install Deck

<i>INSTALL DECK</i>			
<i>Description</i>	Install a resident deck on the SIM. If a deck with the same name is already stored on the SIM, it is replaced by the new one (deck update).		
<i>LENGTH</i>	<i>VALUE</i>	<i>DESCRIPTION</i>	<i>M/O</i>
1	InstDeckTag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Deck Element (see /SBC/)	M
1	DeckTag	Deck Tag (+ Attribute Byte presence flag)	
1-3	L	Length of subsequent data (length coding in BER-TLV format)	
L	Value	Deck content	





<i>OUTPUT</i>		<i>DESCRIPTION</i>	
		None.	
<i>ERROR CODE</i>		<i>DESCRIPTION</i>	<i>Action</i>
NO_ERROR		OK	
MEMORY_MGT		Memory allocation problem	Stop

### 8.1.3 Uninstall Deck

<i>UNINSTALL DECK</i>			
<i>Description</i>	Delete the specified resident deck on the SIM.		
<i>LENGTH</i>	<i>VALUE</i>	<i>DESCRIPTION</i>	<i>M/O</i>
1	UninstDeck Tag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Deck Identification Element	M
1	DeckIdTag	Deck Identification Tag	
1-3	L	Deck Identification Value Length	
L	Value	Deck Identification Value (Unique identification of the Deck) (Same coding as the AddressReference: long name)	
<i>OUTPUT</i>		<i>DESCRIPTION</i>	
		None.	
<i>ERROR CODE</i>		<i>DESCRIPTION</i>	<i>Action</i>
NO_ERROR		OK	
UNDEFINED_REF		Reference to undefined (the specified deck cannot be found)	Stop

## 8.2 SERVICE PERMANENT STORE

Note: As the Service Permanent Store is optional; the support of the following commands is also optional.

### 8.2.1 Introduction

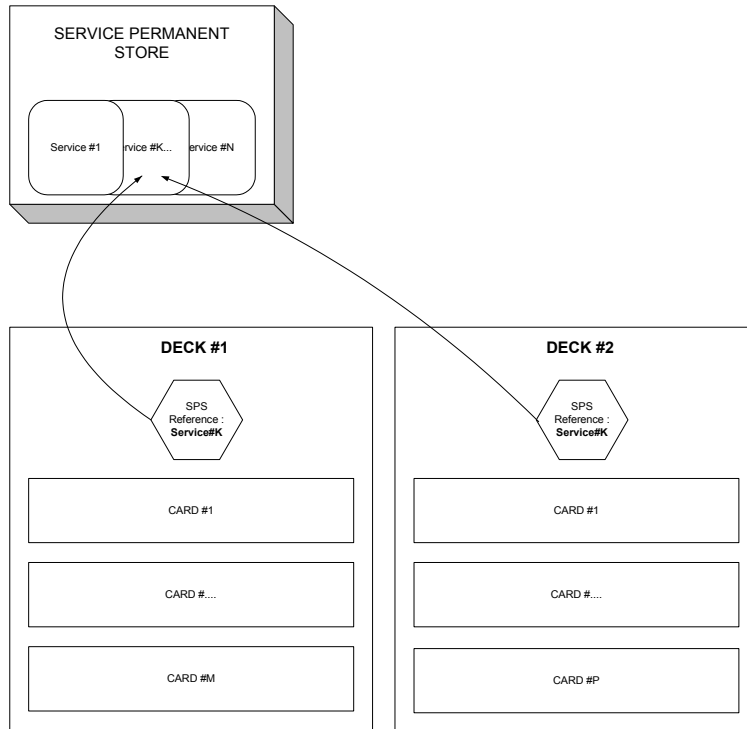
A service is represented by a set of Decks (WML jargon). The decks are able to interact with the User via STK commands, store values in variables, send requests to the gateway.

A persistent storage area for variables, is provided in the card, and shared by all decks of a given service. It is called the Service Permanent Store (SPS).



The SPS store size and Id are allocated by the SPS Allocate administrative command. Then permanent variables are created in this SPS using the Allocate Permanent Variable administrative command.

Any Deck shall refer to a SPS reference prior to be able to use its variables.



### 8.2.2 SPS Allocate

<i><b>SPS ALLOCATE</b></i>			
<i><b>Description</b></i>	Reserves/frees a certain amount of space for the service’s permanent variable memory. This operation is done under the operator’s control.		
<i><b>LENGTH</b></i>	<i><b>VALUE</b></i>	<i><b>DESCRIPTION</b></i>	<i><b>M/O</b></i>
1	SPSAllocTag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Optional Bytecode attributes	O



1	0xXX	<p>Attribute byte :</p> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px;">Bit#</td> <td style="padding: 2px;">7</td> <td style="padding: 2px;">6</td> <td style="padding: 2px;">5</td> <td style="padding: 2px;">4</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;">2</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> </table> <p style="margin: 10px 0 0 0;"> <span style="display: inline-block; transform: rotate(-90deg); transform-origin: left top; white-space: nowrap;">FreeServiceVariables</span> <span style="display: inline-block; transform: rotate(-90deg); transform-origin: left top; white-space: nowrap;">Follow Bit</span> </p> </div> <p><b>FreeServiceVariables Attribute :</b></p> <p>Default Value : 0 : Permanent Variables of the specified service are unchanged.</p> <p>If Set : 1 : Permanent Variables of the specified service are removed.</p>	Bit#	7	6	5	4	3	2	1	0										
Bit#	7	6	5	4	3	2	1	0													
		Request Id Element	M																		
2	0-65535	Request Id																			
		Service Id Element	M																		
1	SPSTag	Tag																			
1	L	Length (max 8)																			
L	Value	Value																			
		Memory Amount Element	M																		
1	InlineValueTag	Tag																			
1	L	Length																			
L	Value	New amount of memory for the specified service. If the value is greater than 0: if the service does not exist, the space is allocated, else the space is reduced or extended to the new amount of memory. If the value is 0, the memory is freed.																			
<b>OUTPUT</b>		<b>DESCRIPTION</b>																			
		None.																			
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>																		
NO_ERROR		OK																			
MEMORY_MGT		Memory allocation problem	Stop																		



Note: if the SPS memory is reduced with the attribute flag FreeServiceVariables set to 0, then a memory allocation error will be raised if the allocated variables don't fit into the new allocated memory space.

### 8.2.3 Allocate Permanent Variable

<b>ALLOCATE PERMANENT VARIABLE</b>			
<b>Description</b>	Reserves/frees a certain amount of space for a permanent variable in the specified Service Permanent Store memory.  This operation is done under the operator's control.		
<b>LENGTH</b>	<b>VALUE</b>	<b>DESCRIPTION</b>	<b>M/O</b>
1	AllocPermVar Tag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Service Id Element	M
1	SPSTag	Tag	
1	L	Length	
L	Value	Value	
		Permanent Variable Element	M
1	PermVarId	Permanent Variable ID to allocate/free	
		Memory Amount Element	M
1	0-255	New maximum size of the permanent variable value. If the value is greater than 0: if no space is allocated for this variable, allocation is done, else the space is reduced or extended to the new value. If the value is 0: the space reserved for this variable is freed.	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
		None.	
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>
NO_ERROR		OK	
UNDEFINED_REF_SERV		Reference to undefined service Id	Stop
UNDEFINED_REF		Reference to undefined variable Id	Stop
MEMORY_MGT		Memory allocation problem	Stop

### 8.2.4 Set Permanent Variable

<b>SET PERMANENT VARIABLE</b>	



<b>Description</b>	Set the value of a permanent variable.		
<b>LENGTH</b>	<b>VALUE</b>	<b>DESCRIPTION</b>	<b>M/O</b>
1	SetPermVar Tag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Service Id Element	M
1	SPSTag	Tag	
1	X	Length	
X	Value	Value	
		Permanent Variable Element	M
1	PermVarId	Permanent Variable ID	
		Variable Value Element	M
1	InlineValueTag	InlineValue Tag	
1-2	Y	Length (< 255)	
Y	Value	New value of the permanent variable	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
		None.	
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>
NO_ERROR		OK	
UNDEFINED_REF_SERV		Reference to undefined service Id	Stop
UNDEFINED_REF		Reference to undefined variable Id	Stop
MEMORY_MGT		Memory allocation problem	Stop

### 8.2.5 Get Permanent Variable

<b>GET PERMANENT VARIABLE</b>			
<b>Description</b>	Get the value of a permanent variable.		
<b>LENGTH</b>	<b>VALUE</b>	<b>DESCRIPTION</b>	<b>M/O</b>
1	GetPermVar Tag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M



2	0-65535	Request Id	
		Service Id Element	M
1	SPSTag	Tag	
1	X	Length	
X	Value	Value	
		Permanent Variable Element	M
1	PermVarId	Permanent Variable ID	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
		The value of the variable is sent in the browser response using the request id given in the Request Id Element (see 7.2).	
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>
NO_ERROR		OK	
UNDEFINED_REF_SERV		Reference to undefined service Id	Stop
UNDEFINED_REF		Reference to undefined variable Id	Stop

## 8.3 ENVIRONMENT VARIABLES

### 8.3.1 Introduction

Environment Variables define parameters of the browser and the mobile phone. These Environment Variables are used both by the Gateway for administrative purposes and by the applications loaded online to the SIM to determine certain application related (standard) procedures. The settings defined through Environment variables are shared among services.

See Section 12 for a complete description of the environment variables.

### 8.3.2 Admin Set Env

<b>SET ENV</b>			
<b>Description</b>	Update the value of one modifiable environment variables.		
<b>LENGTH</b>	<b>VALUE</b>	<b>DESCRIPTION</b>	<b>M/O</b>
1	AdmSetEnvTag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Environment Variable to modify	M
1	EnvVarRef	Environment Variable ID	



		Value to assign	M
1	InlineValueTag	InlineValue Tag	
1-3	X	InlineValue length	
X	Value	Value to assign to the environment variable.	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
		None.	
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>
NO_ERROR		OK	No Stop
UNDEFINED_REF		Reference to undefined environment variable.	Stop
MEMORY_MGT		Memory allocation problem or access condition not fulfilled	Stop

### 8.3.3 Admin Get Env

<b>ADMIN GET ENV</b>			
<b>Description</b>	Get the value of one environment variables (i.e. generate a string corresponding to the element required into a variable) and send it to the S@T gateway.		
<b>LENGTH</b>	<b>VALUE</b>	<b>DESCRIPTION</b>	<b>M/O</b>
1	AdmGetEnv Tag	Bytecode (+ Attribute presence flag)	M
1-3	L	Length of subsequent data (length coded in BER-TLV)	M
		Request Id Element	M
2	0-65535	Request Id	
		Environment Variable Reference	M
1	EnvVarRef	ENV Variable ID to get	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
		Value of the specified environment variable as defined in chapter 12. The value of the variable is sent in the browser response using the request id given in the Request Id Element (see 7.2).	
<b>ERROR CODE</b>		<b>DESCRIPTION</b>	<b>Action</b>
NO_ERROR		OK	No Stop

Error handling procedure :

If the reference does not exist, return a NULL string.



## 8.4 CONTEXTUAL MENUS

The update of a contextual menu by the operator is done using the same logic as the one defined in [SBC]. The command is like the Manage Contextual Menu Item macro of SBC, but with a RequestId in it.

In the case of administrative commands, the Operator Item bit is always set to 1.

### 8.4.1 Admin Manage Contextual Menu Item

<i>ADMIN_MANAGE_CONTEXTUAL_MENUITEM</i>																														
<i>DESCRIPTION</i>	This function is used to display/hide system contextual menu items, to add/update/remove application specific items.  If an application menu item id is already existing, then it is updated with the new value.																													
<i>LENGTH</i>	<i>VALUE</i>	<i>DESCRIPTION</i>	<i>M/O</i>																											
1	AdmManageContextualMenu ItemTag	Byte code (+ attribute presence flag)	M																											
1-3	L	Length of subsequent data (length coded in BER-TLV)	M																											
		Optional byte code attributes	O																											
1	0xXX	Attribute byte :  <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px;">Bit#</td> <td style="padding: 2px;">7</td> <td style="padding: 2px;">6</td> <td style="padding: 2px;">5</td> <td style="padding: 2px;">4</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;">2</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> </table>   <table style="margin: auto; text-align: center;"> <tr> <td style="padding: 5px;">Follow Bit</td> <td style="padding: 5px;">Remove/Hide</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> <td style="padding: 5px;">RFU</td> </tr> </table>   <b>Remove/Hide Attribute :</b>            Default Value : 0 : Add/Display the item specified            If Set : 1 : Remove/Hide the item specified.         </div>	Bit#	7	6	5	4	3	2	1	0										Follow Bit	Remove/Hide	RFU	RFU	RFU	RFU	RFU	RFU	RFU	
Bit#	7	6	5	4	3	2	1	0																						
Follow Bit	Remove/Hide	RFU	RFU	RFU	RFU	RFU	RFU	RFU																						
		Request Id Element	M																											
2	0-65535	Request Id																												





		Contextual menu item identifier (see /SBC/ for coding)	M
1	Id	Either system or application depending on the coding.	
		Couple of contextual menu item text and URL to go, if item was selected This field is used to add or to update application menu item	O
1	CoupleTag	Tag couple	
1-3	L	Length	
X	TLV	Inline Value or Variable Reference	
Y	TLV	URL reference	
<b>OUTPUT</b>		<b>DESCRIPTION</b>	
OutputVariable		None.	
<b>ERROR CODES</b>		<b>DESCRIPTION</b>	
<b>Action</b>			
NO_ERROR		OK	NoStop
UNDEFINED_REF		Reference to undefined	Stop
MEMORY_ALLOCATION		Too many items allocated	Stop
TYPE_MISMATCH		System/Application menu item or Card/Operator bit mismatch.	Stop



# 9 CONNECTION PROCEDURE FOR AN ADMINISTRATION SESSION

An Administration session is initiated by the S@T Gateway.

To do so, the S@T Gateway sends a CONNECT\_REQ message to the S@T Browser.

Additional information may be sent in EXPRESS\_DATA just after the CONNECT\_REQ message.

The value of the Protocol ID to be sent in a CONNECT\_REQ message is 02 (S@TAdministration protocol).

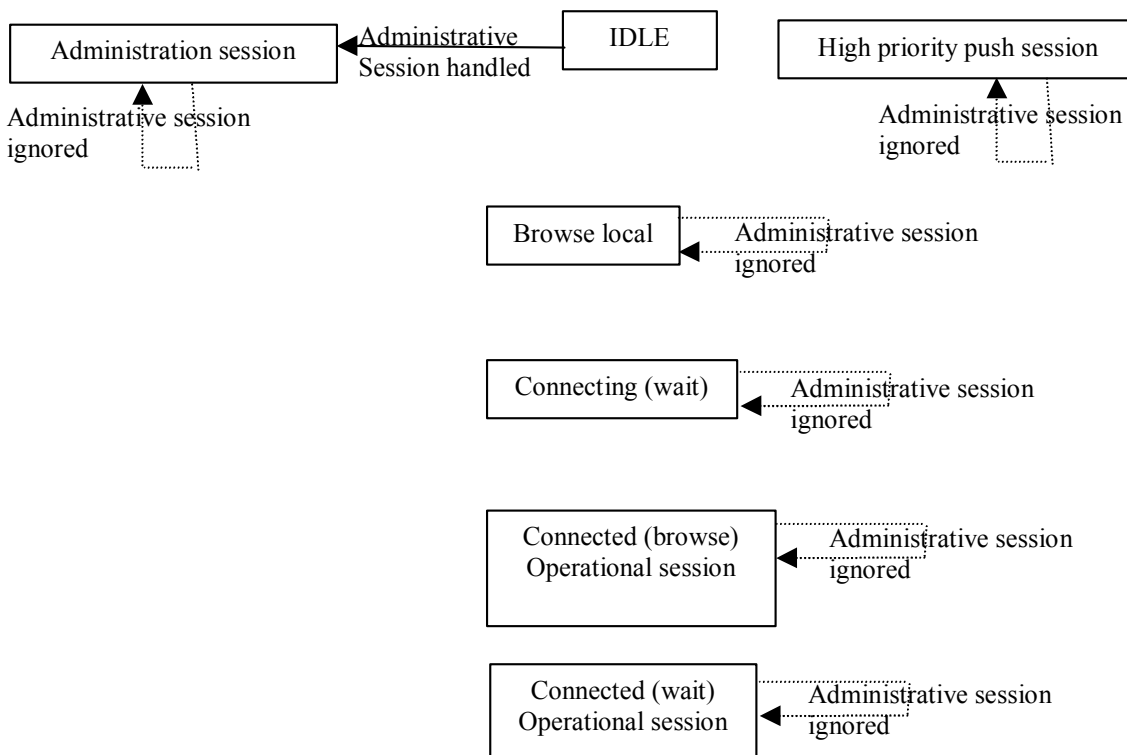
For security reasons, an Administration Session can NOT be mixed with a Browsing Session.

If the browser does not support multiple sessions, an administrative session MUST NOT interrupt a browsing or a high priority push session.

The diagram below recaps the browser behaviour:

Notes :

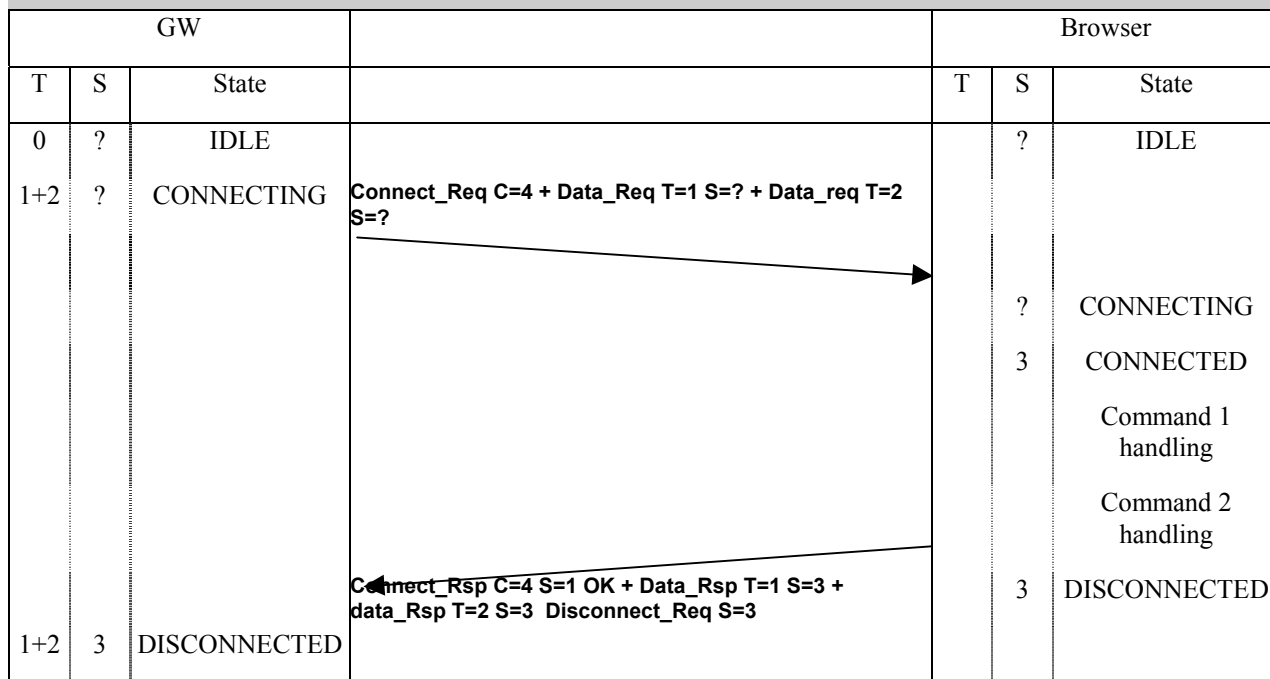
- Browse local state means that a deck is currently browsed without any session established with the gateway.
- Connecting (wait) state means that no session is established and a session establishment is requested.
- Connected (browse) state means that a session is established and a deck is currently browsed.
- Connected (wait) state means that a session is established and a deck is requested.
- This diagram is valid if multi session is not supported.



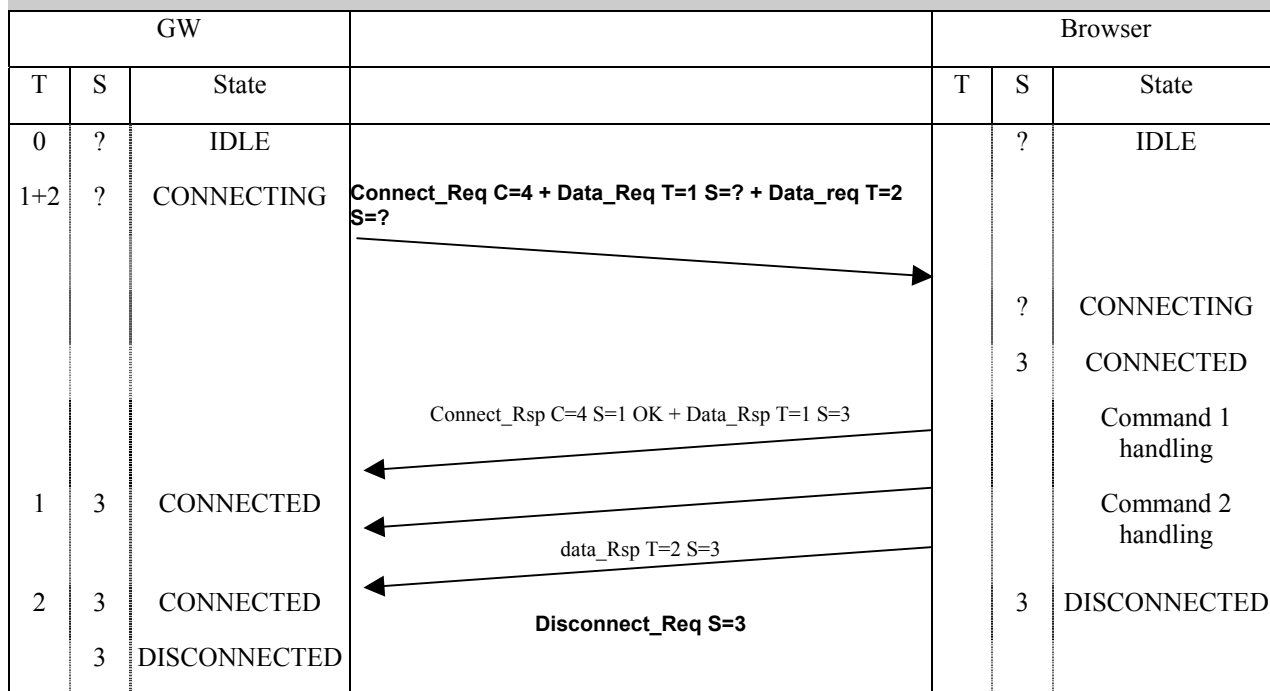
The gateway MUST send all the commands (install deck,...) in a single S@T 1.20 message, transmitted in only one 03.48 packet. Commands can be answered in one or several S@T 1.20 messages, as illustrated below :



Administrative session exchanges (form 1)



Administrative session exchanges (form 2)



The browser MUST explicitly disconnect the gateway after the full “message” has been treated.



## 10 ERROR MANAGEMENT

### 10.1 Error indication

With administrative commands, the error code is sent to the Gateway.

#### **Table of predefined Error Codes**

In case the same error exist in /SBC/, the error code is the same.

Type of error	Coding
Problem in memory management	0x6F05
Reference to undefined	0x6F07
Reference to undefined service	0x6F10
General unspecific error	0x6FFF

In the case no error occurs the value 0x0000 is sent to the Gateway.



## 11 LIST OF COMMANDS TAGS

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

<b>Command</b>	<b>Value</b>
BrowserResponseTag	0x50
InstDeckTag	0x51
UninstDeckTag	0x52
SPSAllocTag	0x53
AllocPermVarTag	0x54
SetPermVarTag	0x55
GetPermVarTag	0x56
AdmSetEnvTag	0x57
AdmGetEnvTag	0x58
AdmManageContextualMenuItemTag	0x59

Unsupported tags must be ignored.



## 12 LIST OF VARIABLE IDENTIFIERS

All the environment variables defined in the S@T v1.00 document are usable in the Admin Get Env macro, as well as proprietary information. . These variables are sent using a TLV format, the value of the variable identifier to use being defined below. T is the ID of a variable defined below, L is the length of its value, V its value

The following variables are defined only for the administrative protocol.

### 12.1 Resident Decks Buffer Size

Variable name: *ResidentDecksBufferSize*

Access Read only

Availability: Mandatory

Length: 2

Coding: binary value of the maximum deck size

### 12.2 Reception Buffer Size

The Reception Buffer Size is the available memory size for data downloaded by the Gateway, i.e. not only for remote Decks downloaded by the Gateway but from SSP level to SBC level, including STLS (0348 header)

Variable name: *ReceptionBufferSize*

Access Read only

Availability: Mandatory

Length: 1

Coding :in multiples of 256 bytes

### 12.3 List of Resident Decks

The list of resident decks may be changed during runtime of the browser if a new deck is installed.

The list of resident decks is maintained by the SIM, i. e. is only readable by the gateway. Any update of this list shall only be performed by the SIM, if a deck has been installed successfully.



Variable name: *ResidentDecksList*

Access Read only

Availability: Mandatory

Length Variable

Coding :

The content field is made of a list of LV elements, containing the Resident deck names.

## 12.4 List of Executable Elements

The list of executable elements defines all available executable elements. This list shall not be changed during runtime of the browser.

The list of executable elements is maintained by the SIM, i. e. is only readable by the gateway. Any update of this list shall only be performed by the SIM, if an executable element has been stored successfully.

Variable name: *ExecutableElementsList*

Access Read only

Availability: Mandatory

Length: Variable

Coding

List of Executable elements IDS (coded According to chapter 'Execute Elements Coding' in /SBC/ document)

## 12.5 Gateway Address

This variable contains the destination address of the gateway providing the SIM with the SBC. All (administrative) data requested by the server (part) of the system are send to this destination address.

Variable name: *GatewayAddress*

Access Read Write

Availability: Optional

Length: 3

This variable allows accessing the SSP gateway address used in connection message.



Coding :

See SSP document (Server Address).

## 12.6 Variable identifier Coding

Variable	Support	Description	Value
<i>ResidentDecksBufferSize</i>	Mandatory	Size of the memory for the storage of SIM resident decks	0x40
<i>ReceptionBufferSize</i>	Mandatory	Available memory size for Data downloaded by the Gateway from SSP level to SBC level, including STLS (0348 header	0x41
<i>ResidentDecksList</i>	Mandatory	List of all resident decks in the SIM	0x42
<i>ExecutableElementsList</i>	Mandatory	List of all executable elements in the SIM	0x43
<i>GatewayAddress</i>	Optional	Gateway address	0x44
<b>RFU</b>		Reserved values for this specification	0x45-0x4F

### Note:

Outside the scope of this specification the following identifier ranges are defined:

- 0x00 – 0x3F SBC Environment Variables Identifiers
- 0x50 – 0x5F Operational Mode Identifiers
- 0x60 – 0x6F Administrative Mode Proprietary Identifiers
- 0x70 – 0x7F Operational Mode Proprietary Identifiers
- 0x80 – 0xFF RFU

## 13 Annex : OPTIONAL FEATURES

The following features are optional :

- SPS (service permanent store)





## 14 History

<b>Document history</b>		
Release	Approved by	Comment
1.0.0	SIM Alliance TDG	Document S@T 01.60 renamed to 01.21 + Add macro AdminManageContextualMenu (CR Schlumberger-Admin-31-Mar-2000 approved during meeting 16)
1.0.1	SIM Alliance TDG	Changes after meeting #18 CR : 10017
1.0.2	SIM Alliance TDG	Changes after meeting #18 ad hoc CRs: 10031 and 10040
1.0.3	SIM Alliance TDG	Editorial Changes for Release 2000-06
1.0.4	SIM Alliance TDG	Add CRs: 10060, 10065
1.0.5	SIM Alliance TDG	Add CRs: 10072, 10117, 10120
1.0.6	SIM Alliance TDG	Editorial changes at meeting #30 for publication
2.0.0	SIM Alliance TDG	Editorial changes for publication



## 14.1 Annex: LIST OF CHANGE REQUESTS [informative]

CR Number	CR Identifier	Subject	Document Reference	Status / Meeting No.
10017	GEMPLUS-WG1-MAY-2000#3	PRECISE DEFINITION TO “TEMPORARY DECK BUFFER SIZE”	S@T 1.21 V1.0.0	Accepted #17
10031	GEMPLUS-WG1-MAY-2000#18.3	RFU INDICATION AFTER FOLLOW BIT IN ATTRIBUTE FIELD	S@T 1.21 V1.0.1	Accepted #18
10040	GEMPLUS-WG1-MAY-2000#18.5	COUPLE USAGE IN CONTEXTUAL MENUS	S@T 1.21 V1.0.1	Accepted #18
10042	GEMPLUS-WG1-MAY-2000#18.7	OPTIONAL / MANDATORY FEATURES	S@T 1.21 V1.0.1	Accepted #18
10060	SCHLUMBERGER-WG1 - AUGUST-2000#4	Edit correction	S@T 1.21 V1.0.3	Accepted #20
10065	SCHLUMBERGER-WG1 – AUGUST-2000#9	Admin SetEnv	S@T 1.21 V1.0.3	Accepted #20
10072	SCHLUMBERGER –WG1 – AUGUST 2000#11	Admin Manage Contextual Menu Items	S@T 1.21 V1.0.4	Accepted #26
10117	GEMPLUS-APR-2001#29-5.DOC	SPECIFY END OF ADMIN SESSION TO AVOID DEAD LOCKS	S@T 1.21 V1.0.4	Accepted #28
10120	GEMPLUS-WG1-MAY-2001#28-3	Proposal for state diagrams in 1.21/1.22/1.23	S@T 1.21 V1.0.4	Accepted #29