

Source : USAT INterpreter test work item Rapporteur

Title : USAT Interpreter Interoperability Test Specification TS 31.123 v0.3.0

For : Information

Subject: Stable version of USAT Interpreter test specification to be sent to TSG T for information

3GPP TS 31.123 V0.3.0 (2003-11)

Technical Specification



3rd Generation Partnership Project; Technical Specification Group Terminals; USAT Interpreter Interoperability Test Specification

The present document has been developed within the 3rd Generation Partnership Project (3GPP™) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented.

This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification.

Specifications and reports for implementation of the 3GPP™ system should be obtained via the 3GPP Organisational Partners' Publications Offices.

Keywords

USIM, UICC, USAT interpreter, interoperability
test

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

<http://www.3gpp.org>

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© 2000, 3GPP Organizational Partners (ARIB, CWTS, ETSI, T1, TTA, TTC).
All rights reserved.

Contents

Foreword	10
1 Scope	11
2 Normative References	11
3 Definitions, symbols, abbreviations and coding	12
3.1 Definitions	12
3.1.1 Default data formatting	12
3.1.2 Coding Conventions	12
3.2 Symbols	12
3.3 Abbreviations	12
3.4 Terminology	12
4 Introduction	14
4.1 Test definition and applicability	14
4.2 Specification method	14
4.3 Test entities	15
4.3.1 Overview	15
4.3.2 IUT	15
4.3.3 UI-Converter Tool	16
4.3.4 ME simulator	16
4.3.5 USAT Interpreter Gateway System simulator	16
4.3.6 Test client	16
4.3.7 Card reader	16
4.3.8 ME	16
4.3.9 User	17
4.4 Conformance contexts	17
4.5 IUT conditions	17
4.5.1 Security	17
4.5.2 IUT initial configuration	17
5 Test Group Hierarchy	18
5.1 Test groups under 31.113	18
5.1.1 Test groups under MEC	19
5.1.2 Test groups under DS	20
5.1.3 Test groups under UI-CMD	20
5.1.4 Test groups under NC	21
5.2 Test groups under 31.114	22
5.2.1 Test groups under TPT	22
5.2.2 Test groups under ADM	23
5.2.2.1 Test groups under ADM_MEC	23
5.2.2.2 Test groups under ADM_DS	23
5.2.2.3 Test groups under ADM_CMD	23
6 Test Specification for 31.113 : USAT Interpreter general mechanisms (MEC)	24
6.1 Navigation	24
6.1.1 Definition and applicability	24
6.2 Communication externally	24
6.2.1 Definition and applicability	24
6.2.2 Conformance requirements	24
6.2.3 Test cases	25
6.3 Terminal Response Handler	25
6.3.1 Definition and applicability	25
6.3.2 Conformance requirements	25
6.3.3 Test cases	26
6.4 Activation	27

6.4.1	Definition and applicability	27
6.4.2	Conformance requirements	28
6.4.3	Test cases	28
6.5	Variable Management.....	29
6.5.1	Definition and applicability	29
6.5.2	Conformance requirements	29
6.5.2.1	Conformance requirements for Usage areas	29
6.5.2.2	Conformance requirements for Variable substitution.....	30
6.5.3	Test cases	30
6.5.3.1	Test cases for usage areas – environment variables	30
6.5.3.2	Test cases for usage areas – Permanent variables	30
6.5.3.3	Test cases for usage areas – Temporary variables	31
6.5.3.4	Test cases for variable substitution – method 1	32
6.5.3.5	Test cases for variable substitution – method 2	32
6.6	History list	32
6.6.1	Definition and applicability	32
6.6.2	Conformance requirements	33
6.6.3	Test cases	33
7	Test specification for 31.113 : USAT Interpreter data structures (DS).....	36
7.1	Page.....	36
7.1.1	Definition and applicability	36
7.1.2	Keeping variables	36
7.1.2.1	Conformance requirements	36
7.1.2.2	Test cases for Keeping variables	36
7.1.3	Caching	36
7.1.3.1	Conformance requirements	37
7.1.3.2	Test cases for Caching	37
7.1.4	Modified system TRH configuration	37
7.1.4.1	Configuration requirements	37
7.1.4.2	Test cases for Modified TRH configuration	37
7.2	Navigation Unit.....	39
7.2.1	Definition and applicability	39
7.2.2	Conformance requirements	39
7.2.3	Test cases	39
7.3	Anchor Reference	40
7.3.1	Definition and applicability	40
7.3.2	Conformance requirements	40
7.3.3	Test cases	40
7.4	Variable Identifier List	41
7.5	Inline Value	41
7.5.1	Definition and applicability	41
7.5.2	Conformance requirements	41
7.5.3	Test cases	41
7.6	Inline Value 2.....	41
7.7	Input list	41
7.8	Ordered TLV list	41
7.9	Page reference	41
7.9.1	Definition and applicability	41
7.9.2	Conformance requirements	42
7.9.3	Test cases	42
7.10	Submit	43
8	Test specification for 31.113 : USAT Interpreter commands (UI-CMD).....	43
8.1	Set Variable	43
8.1.1	Definition and applicability	43
8.1.2	Conformance requirements	43
8.1.3	Test cases	43
8.2	Assign And Branch	43
8.2.1	Definition and applicability	43

8.2.2	Conformance requirements	44
8.2.3	Test cases	44
8.3	Extract	45
8.3.1	Definition and applicability	45
8.3.2	Conformance requirements	45
8.3.3	Test cases	46
8.4	Go Back.....	46
8.4.1	Definition and applicability	46
8.4.2	Conformance requirements	46
8.4.3	Test cases.....	46
8.5	Branch On Variable Value	46
8.5.1	Definition and applicability	46
8.5.2	Conformance requirements	47
8.5.3	Test cases	47
8.6	Exit	47
8.6.1	Definition and applicability	47
8.6.2	Conformance requirements	47
8.6.3	Test cases	47
8.7	Execute USAT Command.....	47
8.7.1	Definition and applicability	47
8.7.2	Conformance requirements	47
8.8	Execute Native Command	48
8.8.1	Definition and applicability	48
8.8.2	Conformance requirements	48
8.8.3	Test cases.....	48
8.9	Get Length.....	48
8.9.1	Definition and applicability	48
8.9.2	Conformance requirements	49
8.9.3	Test cases	49
8.10	Get TLV Value	49
8.10.1	Definition and applicability	49
8.10.2	Conformance requirements	49
8.10.3	Test cases.....	49
8.11	Display Text	49
8.11.1	Definition and applicability	49
8.11.2	Conformance requirements	50
8.11.3	Test cases.....	50
8.12	Get Input	50
8.12.1	Definition and applicability	50
8.12.2	Conformance requirements	50
8.12.3	Test cases.....	50
9	Test specification for 31.113 : Native commands (NC).....	51
9.1	PKCS#7 Signature Plug-in	51
9.1.1	Definition and applicability	51
9.1.2	Conformance requirements	51
9.1.3	Test cases.....	51
10	Test specification for 31.114 : Transmission protocol (TPT)	53
10.1	Handling of Pull messages.....	53
10.2	Handling of Push messages	54
10.2.1	Definition and applicability	54
10.2.2	Conformance requirements	54
10.2.3	Test cases.....	54
10.3	Handling of Administration messages	54
10.3.1	Definition and applicability	54
10.3.2	Conformance requirements	55
10.3.3	Test cases	55
10.4	Transport bearer.....	55
10.4.1	Definition and applicability	55

10.4.2	Conformance requirements	55
10.4.3	Test cases	55
10.5	23.048 security	55
10.5.1	Definition and applicability	56
10.5.2	Conformance requirements	56
10.5.3	Test cases	56
10.6	Addressing and key selection	56
10.6.1	Definition and applicability	56
10.6.2	Conformance requirements	56
10.6.3	Test cases	57
10.7	USAT Interpreter originated application messages	58
10.7.1	Definition and applicability	58
10.7.2	Conformance requirements	58
10.7.3	Test cases	58
10.7.3.1	Test cases for Pull request	58
10.7.3.2	Test cases for Adm response	59
10.8	Gateway originated application messages (GWO-APP)	59
10.8.1	Definition and applicability	59
10.8.2	Conformance requirements	59
10.8.3	Test cases	60
10.8.3.1	Test cases for Pull response	60
10.8.3.2	Test cases for Push initiation	60
10.8.3.3	Test cases for Administration requests	60
11	Test specification 31.114 : Administration procedures	60
11.1	General administration procedures	60
11.2	Administration data structures	60
11.2.1	ADM Request	60
11.2.1.1	Definition and applicability	60
11.2.1.2	Conformance requirements	60
11.2.1.3	Test cases	61
11.2.2	ADM response	61
11.2.2.1	Definition and applicability	61
11.2.2.2	Conformance requirements	61
11.2.2.3	Test cases	61
11.2.3	ADM result	61
11.3	Administration commands	62
11.3.1	Install Page	62
11.3.1.1	Definition and applicability	62
11.3.1.2	Conformance requirements	62
11.3.1.3	Test cases	62
11.3.2	Remove Page	63
11.3.2.1	Definition and applicability	63
11.3.2.2	Conformance requirements	63
11.3.2.3	Test cases	63
11.3.3	Configure USAT Interpreter entering menu title	63
11.3.3.1	Definition and applicability	63
11.3.3.2	Conformance requirements	63
11.3.3.3	Test cases	64
11.3.4	Configure menu text for page	64
11.3.4.1	Definition and applicability	64
11.3.4.2	Conformance requirements	64
11.3.4.3	Test cases	64
11.3.5	Remove menu text for page	64
11.3.5.1	Definition and applicability	64
11.3.5.2	Conformance requirements	64
11.3.5.3	Test cases	65
11.3.6	Install event	65
11.3.6.1	Definition and applicability	65
11.3.6.2	Conformance requirements	65

11.3.6.3	Test cases.....	65
11.3.7	Remove event	66
11.3.7.1	Definition and applicability.....	66
11.3.7.2	Conformance requirements.....	66
11.3.7.3	Test cases.....	66
11.3.8	Modify environment variables	66
11.3.8.1	Definition and applicability.....	66
11.3.8.2	Conformance requirements.....	66
11.3.8.3	Test cases.....	66
11.3.9	Modify wait state message.....	67
11.3.9.1	Definition and applicability.....	67
11.3.9.2	Conformance requirements.....	67
11.3.9.3	Test cases.....	67
11.3.10	Modify configuration data.....	67
11.3.10.1	Definition and applicability.....	67
11.3.10.2	Conformance requirements.....	67
11.3.10.3	Test cases.....	68
11.3.11	Modify bearer data.....	68
11.3.11.1	Definition and applicability.....	68
11.3.11.2	Conformance requirements.....	68
11.3.12	Install plug-in.....	68
11.3.12.1	Definition and applicability.....	68
11.3.12.2	Conformance requirements.....	69
11.3.12.3	Test cases.....	69
11.3.13	Remove plug-in.....	69
11.3.13.1	Definition and applicability.....	69
11.3.13.2	Conformance requirements.....	69
11.3.13.3	Test cases.....	69
Annex A (Normative): Test Language Definition.....		70
A.1	Overview.....	70
A.1.1	Root element	70
A.1.2	ui-app-msg element.....	71
A.1.3	ui-me-msg element.....	71
A.1.4	The page elements.....	72
A.1.5	The adm-request element.....	73
A.2	Document Type Definition	74
<!ELEMENT response-length (#PCDATA)>.....		81
A.3	Element descriptions.....	81
Annex B (Normative):Test documents		84
B.1	Test documents and test data.....	84
B.1.1	Page	84
B.1.2	Submit data.....	86
B.1.3	Receive pulled page.....	86
B.1.4	Receive pushed page.....	86
B.1.5	Send admin response.....	87
B.1.6	Receive admin request.....	87
B.1.7	Issue USAT command.....	87
B.1.7.1	Display Text.....	87
B.1.7.2	Select Item.....	87
B.1.8	Receive terminal response.....	88
B.1.9	Activation menu selection	88
B.1.10	Activation event.....	88
B.2	Security descriptions.....	88
Annex C (Normative): Test procedures		90
C.1	Test procedure conditions.....	90
C.2	Test procedure notation.....	90
C.3	Test Case Details	92

C.3.1	Test group MEC	92
C.3.2	Test group DS	92
C.3.3	Test group UI-CMD	92
C.3.4	Test group NC	92
C.3.5	Test group TPT.....	92
C.3.6	Test group ADM	92
Annex D (Informative) : Implementation conformance applicability		94
D.1	Specification status reference	94
D.2	Conformance context applicability.....	94
Annex E (Informative): UI-Converter Tool		97
Annex F (Informative): Change history		97

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document provides the Test Specification for a USAT Interpreter as specified in 3G TS 31.113 [4] and USAT Interpreter Protocol and Administration defined in 3G TS 31.114 [5].

In the present document the technical characteristics and methods of test for conformance testing of the USAT Interpreter are specified. It covers characteristics for a minimum level of conformance necessary to achieve interoperability between USAT Interpreter implementations and USAT Interpreter Gateway Systems.

To verify conformance it is necessary to partly specify the technical characteristics concerned with the interface between the USAT Interpreter and the mobile equipment. Even so, that interface is not subject to conformance testing in the present document.

Depending on the modes of operation supported by a USAT Interpreter implementation, different parts of the present document applies. This is reflected by grouping of tests and definition of conformance contexts.

Technical characteristics are specified in terms of conformance requirements, high-level test case descriptions and test data documents (FFS). A mark-up language is defined for the purpose of specifying test data in XML format. A USAT Interpreter byte code converter tool, that generates the corresponding byte code representation (FFS).

Methods of test are specified in terms of test environment descriptions (including descriptions of test equipment necessary to perform conformance testing) and low-level test procedures (FFS).

A skeleton for implementation conformance statements for USAT Interpreter implementations is provided.

2 Normative References

The following documents contain provisions, which through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] ETSI TS 102.221: "UICC-Terminal Interface; Physical and Logical Characteristics".
- [2] 3GPP TS 31.101: "UICC-Terminal Interface; Physical and Logical Characteristics".
- [3] 3GPP TS 31.102: "Characteristics of the USIM application".
- [4] 3GPP TS 31.113: " USAT Interpreter Byte Codes ".
- [5] 3GPP TS 31.114: " USAT Interpreter Protocol and Administration ".
- [6] 3GPP TS 31.111: "3rd Generation Partnership Project (3GPP); USIM Application Toolkit (USAT)"

3 Definitions, symbols, abbreviations and coding

3.1 Definitions

3.1.1 Default data formatting

All numeric data enclosed in single quotes (' ') in this document are hexadecimal data.

Where 'X' is used in place of a hexadecimal digit, X ranges from '0' to 'F'. For example, the data '6X' ranges from '60' to '6F' inclusive.

Where data is expressed as a group of bytes, it shall be in the following format: 'XX XX XX... XX', indicating first byte, second byte, third byte etc. in that order.

A string of digits shall be formatted with a continuous string of numeric data and enclosed with single quotes. For example, the string 'XXXXXXXX' where X ranges from 0 to 9 inclusive.

FFS

3.1.2 Coding Conventions

The following coding conventions apply to the present document:

All lengths are presented in bytes, unless otherwise stated. Each byte is represented by bit b8 to b1, where b8 is the most significant bit (MSB) and b1 is the least significant bit (LSB). In each representation, the leftmost bit is the MSB.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

FFS

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

APDU	Application Protocol Data Unit
CRn	Conformance Requirement 'n'
ICS	Implementation Conformance Specification
IUT	Implementation Under Test
LSB	least significant bit
ME	Mobile Equipment
MSB	most significant bit
NU	Navigation unit
TC	test case
TS	Test Specification
UE	User Equipment
UI	USAT interpreter
UI-CMD	USAT Interpreter command, as specified in 31.113 clause 8
USIM	Universal Subscriber Identity Module

3.4 Terminology

The following terms apply in the context of this specification.

MNEMONICS – are used for overview and readability purposes. The mnemonics are based on the test group structure. Test groups, conformance requirements and test cases are identified and numbered based on the test group mnemonic.

Conformance requirement identifier syntax – <test group mnemonic>:CR<item number> e.g. MEC_NAV: CR 1

Test case identifier syntax – <test group mnemonic>:TC<item number> e.g. MEC_NAV: TC 1

4 Introduction

4.1 Test definition and applicability

The following statements are applicable to all test purposes contained within the present document.

- The tests are performed on a USAT Interpreter as specified in 3G TS 31.113 [4] and 3G TS 31.114 [5].
- FFS

4.2 Specification method

The USAT Interpreter test specification provides documentation, a test data mark-up language and a byte code converter tool (FFS).

The tests contained within the present document are specified by:

- High-level definition of conformance requirements and test cases
- Test documents, specifying the byte code test data in XML format (FFS)
- Test procedures, defining the steps of test execution (FFS)

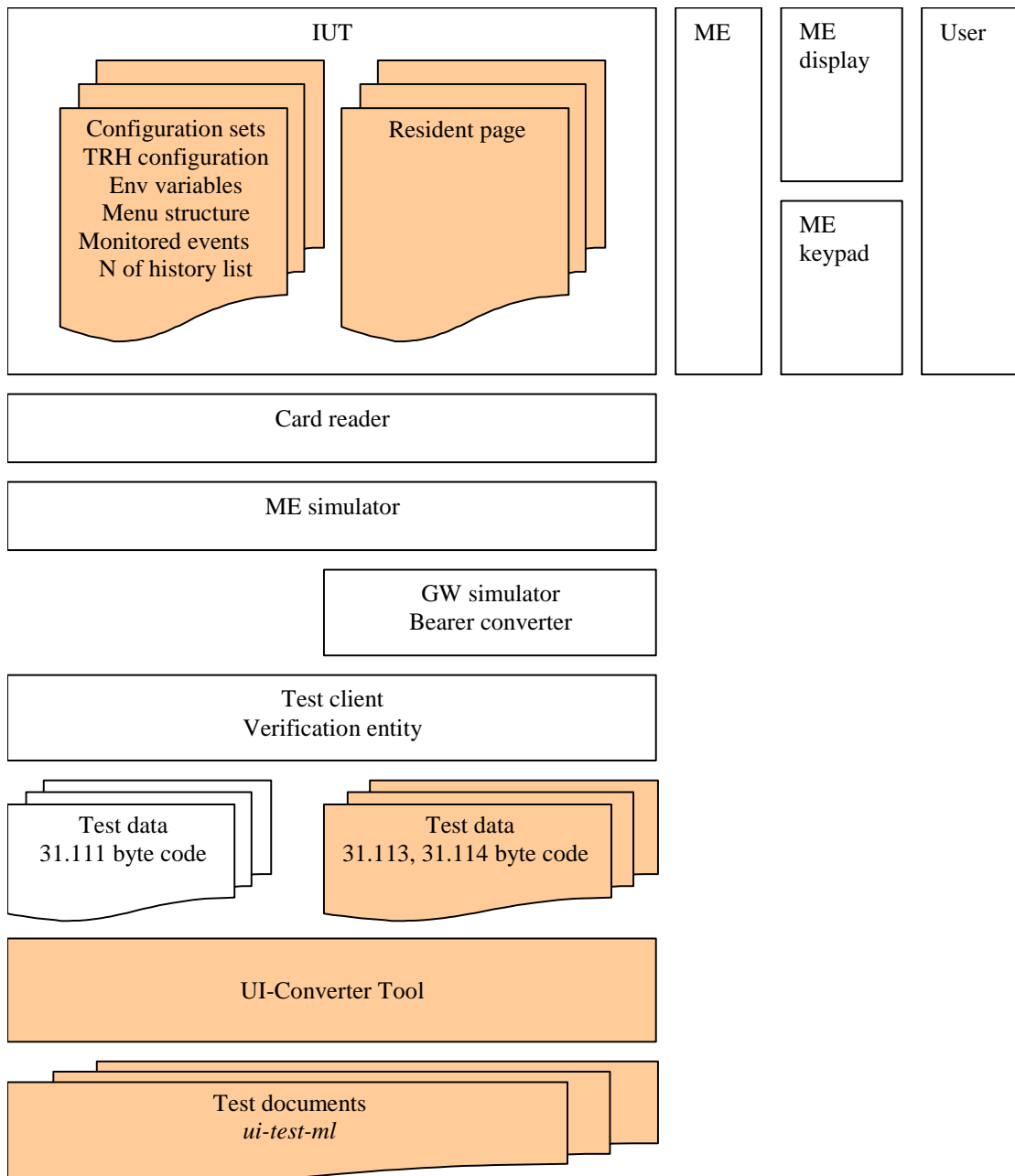
A mark-up language, *ui-test-ml*, has been defined for the purpose of specifying test data on a comprehensive format. The byte code converter tool, the *UI-Converter Tool* (FFS), generates the byte code representation of test data from *ui-test-ml*.

Other test equipment needed to execute tests is described.

4.3 Test entities

4.3.1 Overview

This sub-clause presents an overview of the entities referenced throughout the present document.



4.3.2 IUT

The IUT is composed of the USIM and the USAT Interpreter application. The IUT physically communicates with the ME, the ME-Simulator and/or the GW simulator. The scope of this test specification is to test the USAT Interpreter interfaces and behaviours specified in TS 31.113 and TS 31.114.

The IUT communicates with the ME or ME simulator.

4.3.3 UI-Converter Tool

The UI-Converter Tool shall be able to convert test data between *ui-test-ml* XML documents and USAT Interpreter byte code as specified in TS 31.114 and TS 31.113.

The UI-Converter Tool is provided in this specification:

FFS

4.3.4 ME simulator

The ME simulator shall be able to generate and send any input data to- and receive any input data from the IUT.

The ME simulator shall behave as a real mobile. It shall exchange data with the other entities using either APDU (physical connection) or sockets (software connection). Then, it generates APDU commands as described in TS 31.111 and processes the corresponding responses issued from the IUT. The ME simulator communicates with the test client.

Moreover, the ME simulator would need: input socket for APDU, input socket for bearer formatted MT messages and output socket for bearer formatted MO message.

The interface between the ME simulator and the IUT (i.e. USAT) is not subject to test in this specification. Instead the test data required to test the IUT by the ME simulator are specified only on high-level in *ui-test-ml* as XML documents. The ME simulator shall be able to generate commands to the USAT Interpreter, these commands are required to be correctly encoded and conveyed to the USAT Interpreter.

The implementation of the ME simulator is out of scope of this specification.

4.3.5 USAT Interpreter Gateway System simulator

The USAT Interpreter Gateway System simulator (**GW simulator**) shall be able to communicate with the IUT over the transmission protocol specified in TS 31.114. It simulates the transport bearer level (including security) as well as the application level functionality.

The test data required by the GW simulator are specified both in *ui-test-ml*, as XML documents, and as the corresponding byte code representation, generated by the *UI-Converter Tool*.

The GW simulator may implement the interface to the IUT over a Bearer converter entity that would interface the ME simulator, so that the GW simulator does not communicate directly with the IUT.

FFS

4.3.6 Test client

A test client performs tests by initiating an verifying test execution in the simulated environment. The test client includes the verification entity. The test client communicates/includes the ME simulator and the GW simulator.

4.3.7 Card reader

FFS

4.3.8 ME

The usage of an ME (Mobile Equipment) can be interesting in some cases (e.g. test the user interface and study the look and feel of some USAT commands issued from the IUT to the mobile screen). For some test cases verification of conformance may be executed in a ME This equipment shall be at least a phase 2+ mobile phone.

4.3.9 User

A user is a person using an ME to perform tests by initiating and verifying test execution.

4.4 Conformance contexts

Conformance contexts are defined to express the level of conformance that is actually required for different modes of operations of an IUT. For example, remote navigation is only applicable if the IUT is a USAT interpreter implementation that is tested for conformance to features requiring communication with an external system.

The conformance requirements specified in this documents may belong to one, all or a subset of the conformance contexts. The following conformance contexts are applicable to this test specification:

- Resident
- Remote (pull and push)
- Administration

4.5 IUT conditions

4.5.1 Security

Unless otherwise stated, all the necessary security required to use the USAT Interpreter in the test procedures shall be initially enabled.

4.5.2 IUT initial configuration

The test cases specified requires an IUT with the following initial configuration:

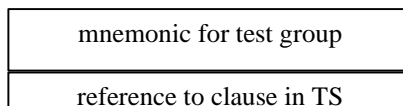
1. The default system TRH configuration as specified in TS 31.113 clause 4.3.
2. System environment variables are written with the following values....FFS
3. Menu structure installed ...FFS
4. Resident pages installed ...FFS
5. Monitored events....FFS
6. Available configuration sets ...FFS

....

5 Test Group Hierarchy

This clause contains the Test Group Hierarchy diagrams which provide a structure for the test purposes and also a correlation between the base specifications, ICS and the present document.

The information contained in each node for each of the diagrams is as follows:



As the Test Group Hierarchy for the USAT Interpreter is large, it has been broken down into sections to aid presentation.

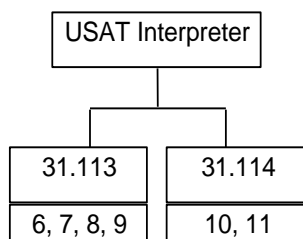


Figure 2: Top Level Test Group Hierarchy

The Test Groups "31.113" and "31.114", shown in figure 2, are broken down further as shown in the following diagrams.

5.1 Test groups under 31.113

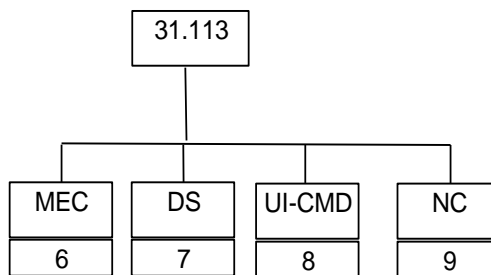
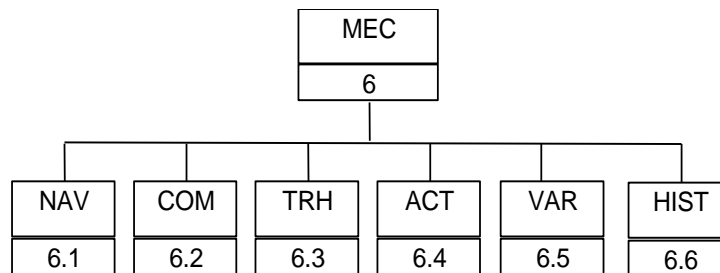


Figure 3: Breakdown of "31.113" Test Group

The mnemonics for this test group are as follows:

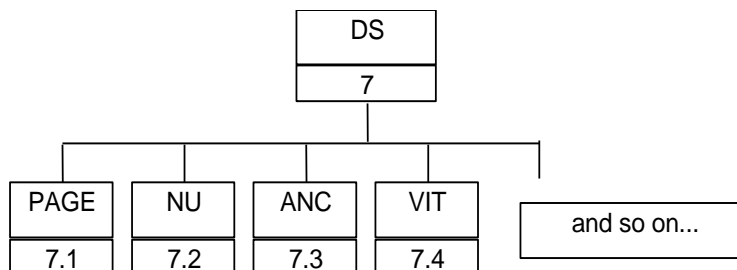
- MEC : General Mechanisms as specified in TS 31.113 clause 4, 5 and 6
- DS : Basic data structures as specified in TS 31.113 clause 7
- UI-CMD : USAT Interpreter commands as specified in TS 31.113 clause 8
- NC : Native commands as specified in TS 31.113 clause 9
- E2E : End to end security as specified in TS 31.113 clause 10, FFS
- MODE : Modes of operation as specified in TS 31.113 clause 11, FFS

5.1.1 Test groups under MEC



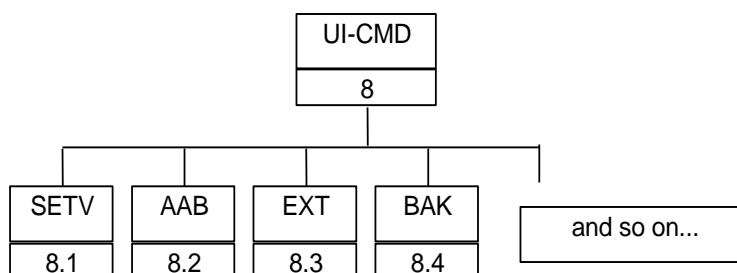
- MEC_NAV : Navigation as specified in TS 31.113 clause 4.1
- MEC_COM : Communication with the external system entity, TS 31.113 clause 4.2
- MEC_TRH : Terminal Response Handler as specified in TS 31.113 clause 4.3
- MEC_ACT : Activation as specified in TS 31.113 clause 4.4
- MEC_VAR : Variable Management as specified in TS 31.113 clause 6
- MEC_HIST : History List Management as specified in TS 31.113 clause 4.6

5.1.2 Test groups under DS



- DS_PAGE : Page Structure as specified in TS 31.113 clause 7.1
- DS_NU : Navigation Unit Structure as specified in TS 31.113 clause 7.2
- DS_ANC : Anchor Reference Structure as specified in TS 31.113 clause 7.3
- DS_VIT : Variable identifier TLV, as specified in TS 31.113 clause 7.4
- DS_INL : Inline Value Structure as specified in TS 31.113 clause 7.5
- DS_INL2 : In line value 2 as specified in TS 31.113 clause 7.6
- DS_INPL : Input list, as specified in TS 31.113 clause 7.7
- DS_OTL : Ordered TLV list, as specified in TS 31.113 clause 7.8
- DS_PREF : Page Reference Structure as specified in TS 31.113 clause 7.9
- DS_SUB : Submit Configuration Structure as specified in TS 31.113 clause 7.10

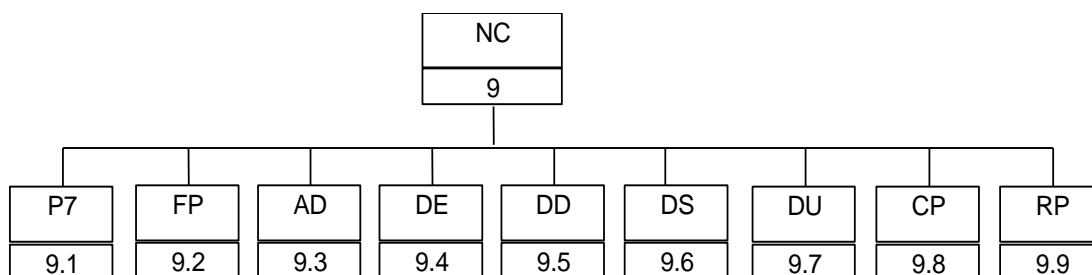
5.1.3 Test groups under UI-CMD



- UI-CMD_SETV : Set Variable Command as specified in TS 31.113 clause 8.1
- UI-CMD_AAB : AssignAndBranch Command as specified in TS 31.113 clause 8.2
- UI-CMD_EXT : Extract Command as specified in TS 31.113 clause 8.3
- UI-CMD_BAK : GoBack Command as specified in TS 31.113 clause 8.4
- UI-CMD_BOVV : BranchOnVariable Value Command as specified in TS 31.113 clause 8.5
- UI-CMD_EXIT : Exit Command as specified in TS 31.113 clause 8.6
- UI-CMD_USATC : Execute USAT command as specified in TS 31.113 clause 8.7

- UI-CMD_ENC : Execute Native Command as specified in TS 31.113 clause 8.8
- UI-CMD_GLEN : Get Length Command as specified in TS 31.113 clause 8.9
- UI-CMD_GTLV : Get TLVValue Command as specified in TS 31.113 clause 8.10
- UI-CMD_DISP : Display Text Command as specified in TS 31.113 clause 8.11
- UI-CMD_GINP : Get Input Command as specified in TS 31.113 clause 8.12

5.1.4 Test groups under NC



- NC_P7 : PKCS#7 Signature Plug-in as specified in TS 31.113 clause 9.1.2.1
- NC_FP : Fingerprint Plug-in as specified in TS 31.113 clause 9.1.2.2
- NC_AD : Asymmetric Decryption Plug-in as specified in TS 31.113 clause 9.1.2.3
- NC_DE : Triple DES Encryption Plug-in as specified in TS 31.113 clause 9.1.3.1
- NC_DD : Triple DES Decryption Plug-in as specified in TS 31.113 clause 9.1.3.2
- NC_DS : Triple DES Sign Plug-in as specified in TS 31.113 clause 9.1.3.3
- NC_DU : Triple DES Unwrap Plug-in as specified in TS 31.113 clause 9.1.3.4
- NC_CP : Change PIN Plug-in as specified in TS 31.113 clause 9.1.4.1
- NC_RP : Reset PIN Plug-in as specified in TS 31.113 clause 9.1.4.2

5.2 Test groups under 31.114

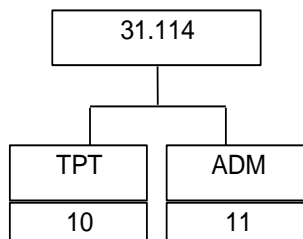


Figure 3: Breakdown of "31.114" Test Group

The mnemonics chosen for this test group are as follows:

- TPT : Transmission Protocol as specified in 3G TS 31.114 [5], clause 5
- ADM : Administration procedures as specified in 3G TS 31.114 [5], clause 6

The Test Groups "TPT" and "ADM", shown in figure 3, are broken down further, as shown in the following diagrams.

5.2.1 Test groups under TPT

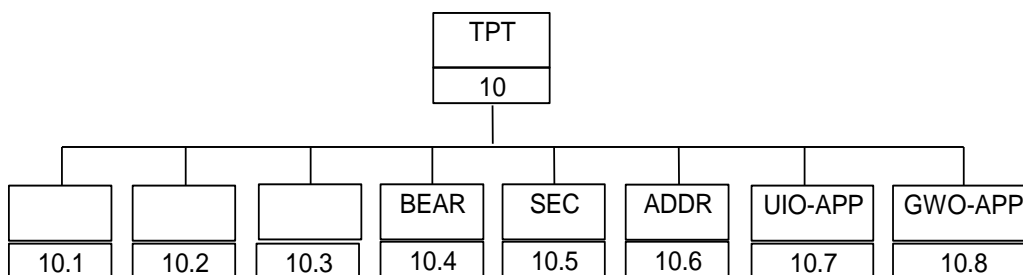


Figure 4: Breakdown of "TPT" Test Group

The mnemonics chosen for this test group are as follows:

- TPT_HAND_PULL: Handling of pull mechanisms as specified in
- TPT_HAND_PUSH : Handling of push mechanisms as specified in [31.114] clause 5.6
- TPT_HAND_ADM : Handling of administrative procedures as specified in [31.114] clause 5.7
- TPT_BEAR : Transport Bearer as specified in [31.114] clause 5.1
- TPT_SEC : 23.048 Security as specified in [31.114] clause 5.2
- TPT_ADDR : Addressing And Key selection as specified in [31.114] clause 5.3
- TPT_UIO-APP : USAT Interpreter originated application messages as specified [31.114] in clause 5.4
- TPT_GWO-APP : Gateway originated application messages as specified in [31.114] clause 5.5

5.2.2 Test groups under ADM

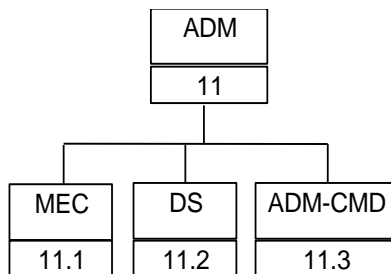


Figure 5: Breakdown of "ADM" Test Group

The mnemonics chosen for this test group are as follows:

- ADM_MEC: UI behaviour and mechanisms for administration as specified in TS 31.114 clause 6
- ADM_DS : Administration data structures (UI-ADM-TLV) as specified in TS 31.114 clause 6.1, 6.1.1, 6.2
- ADM_CMD : Administration commands (ADM-CMD) as specified in TS 31.114 clause 6.1.2

5.2.2.1 Test groups under ADM_MEC

Not further broken down

5.2.2.2 Test groups under ADM_DS

- ADM_DS_REQ : Data structure for ADM request, as specified in 31.114 clause 6.1, 6.1.1
- ADM_DS_RESP : Data structure for ADM response, as specified in 31.114 clause 6.2.1
- ADM_DS_RESU : Data structure for ADM result, as specified in 31.114 clause 6.2.2

5.2.2.3 Test groups under ADM_CMD

- ADM_CMD_INSP: Install Page command as specified in [31.114] clause 6.1.2.1
- ADM_CMD_RMP : Remove Page command as specified in [31.114] clause 6.1.2.2
- ADM_CMD_EMT : Configure Entering Menu Title command as specified in [31.114] clause 6.1.2.3
- ADM_CMD_MTP : Configure Menu Text for Page command as specified in [31.114] clause 6.1.2.4
- ADM_CMD_RMTP : Remove Menu Text for Page command as specified in [31.114] clause 6.1.2.5
- ADM_CMD_INSE : Install Event command as specified in [31.114] clause 6.1.2.6
- ADM_CMD_RME : Remove Event command as specified in [31.114] clause 6.1.2.7
- ADM_CMD_ENV : Modify environment variables, as specified in [31.114] clause 6.1.2.8
- ADM_CMD_WSM : Modify wait state message, as specified in [31.114] clause 6.1.2.9
- ADM_CMD_CD : Modify configuration data, as specified in [31.114] clause 6.1.2.10
- ADM_CMD_BEARD : Modify bearer data, as specified in [31.114] clause 6.1.2.11
- ADM_CMD_INSPI : Install plug-in, as specified in [31.114] clause 6.1.2.12

- ADM_CMD_RMPI : Remove plug-in, as specified in [31.114] clause 6.1.2.13

6 Test Specification for 31.113 : USAT Interpreter general mechanisms (MEC)

This clause specifies on a high-level each of the tests the test group MEC.

6.1 Navigation

MEC_NAV

6.1.1 Definition and applicability

This test group presents reference to the other test groups of this document, where test cases for Navigation are provided. Specification reference: TS 31.113 clause 4.1.

Refer to test groups UI-CMD_AAB, UI-CMD_BOVV, DS_NU(chaining)

6.2 Communication externally

6.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified for communication with an external system. Specification reference: TS 31.113 clause 4.2

The CRs defined are applicable to *Remote* conformance context only.

Conformance is divided further into the conformance areas applicable to communication externally:

- Processing behaviour - no wait state
- Processing behaviour - wait state

6.2.2 Conformance requirements

MEC_COM:CR 1 Submission of outgoing data can be triggered by the UI-CMDs “Assign and branch” and “Branch on variable value”. (Submit UI-TLV).

All information provided by the USAT Interpreter to the external system entity shall be formatted as a Submit TLV.

In the direction from the USAT Interpreter to the external system entity, all variable references within the Submit Data TLV contained in the Submit Configuration TLV are substituted according to method 2 in clause 6.3. The resulting Submit Data TLV containing the substituted variable references with variable content shall then be used within the Submit TLV to be submitted by the USAT Interpreter to the external system entity.

Specification reference : 31.113 clause 4.2.2

MEC_COM:CR 2 Wait state processing behaviour

Specification reference: 31.113 clause 4.2.3

6.2.3 Test cases

- MEC_COM:TC 1 Render a page containing “Assign and branch “ UI-CMD with the Page Reference to a remote page. Verify that a Submit TLV is generated and the data is communicated to the external system. Verify that variable substitution has been performed with method 2.
- MEC_COM:TC 2 As above... but with Branch on variable value.
- Additional tests for CR 2:
- MEC_COM:TC 3 Pull page (TC 1 or TC 2), verify displayed text as rendered (cleared after delay) and retrieval of pulled page.
- MEC_COM:TC 4 Pull page (TC 1 or TC 2) + verify the default-display text (cleared after delay) and retrieval of pulled page
- MEC_COM:TC 5 Pull page, let the gateway system send Pull Response messages with pages to the IUT. Verify that incorrect messages (non-matching request id) are discarded during wait state.
- MEC_COM:TC 6 Pull page, terminate wait state by user interaction. Verify that the Pull Response message containing the pulled page is then discarded when it arrives.

6.3 Terminal Response Handler

MEC_TRH

6.3.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the TRH mechanisms. Specification reference: TS 31.113 clause 4.3.

The conformance is divided into the conformance areas applicable to the TRH mechanism:

- Operation with system TRH configuration only [CR 1, 2, 3, 5]
- Operation with modified system TRH configuration [CR 4]

The CR's specified does not specify the full scope of functionality for a USAT Interpreter implementation operating with temporarily modified system TRH configuration. For further CRs, refer to test group DS_PAGE_TRH

6.3.2 Conformance requirements

- MEC_TRH:CR 1 Process the corresponding action: If the A_{GR} contains only one action (a), then the single action (a) in A_{GR} shall be performed by the USAT Interpreter without user confirmation.
- MEC_TRH:CR 2 several actions configured for a result If there are several actions in the A_{GR}, then the USAT Interpreter shall issue a SELECT ITEM command to let the user select one action (a) out of A_{GR} that shall be used by the USAT Interpreter.
- MEC_TRH:CR 3 In case of exception or no configured action
- If there is no action (a) in A_{GR} the exception action shall be performed by the USAT Interpreter.
- In case of an exception the exception action will apply.
- MEC_TRH:CR 4 The system terminal response handler configuration can be modified temporarily by the terminal response handler modifier (see clause 7.1.8). Refer to test group: DS_PAGE_TRH

MEC_TRH:CR:5 The USAT Interpreter may support storage of texts for user notification for the general result ranges of the system terminal response handler configuration. If texts for user notification are available, the texts shall be used according to clause 7.1.8.3. Refer to DS_PAGE_TRH

6.3.3 Test cases

- Render a page with UI-CMD Execute USAT command, let the IUT receive a terminal response with a result value, verify that the action configured for that result gets executed

Test cases for default system TRH configuration

MEC_TRH:TC 1 Process corresponding action: general result **transaction terminated**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '14'. Verify that the default action "quit USAT Interpreter" is executed.

MEC_TRH:TC 2 Process corresponding action: general result **ok**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '0F'. Verify that the default action "process next bytecode" is executed. Let the page rendered have a second UI-CMD in the NU. Verify that the second NU is rendered.

MEC_TRH:TC 3 Process corresponding action: general result **help request**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '13'. Verify that the default action "retry last proactive command within current USAT Interpreter navigation unit" is executed. Let the page rendered have a second UI-CMD in the NU. Verify that the UI-CMD is rendered again.

MEC_TRH:TC 4 Process corresponding action: general result **no response from user**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '12'. Verify that the default action "quit USAT Interpreter" is executed.

MEC_TRH:TC 5 Process corresponding action: general result **back ward move requested**

Render a page with two NUs, the second NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '11' in response to the second NU. Verify that the default action "go back one entry in history list" is executed.

MEC_TRH:TC 6 Process corresponding action: general result **quit**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '10'. Verify that the default action "quit USAT Interpreter" is executed.

MEC_TRH:TC 7 Process corresponding action: general result **worth to re-try**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '2F'. Verify that the default action "quit USAT Interpreter" is executed.

MEC_TRH:TC 8 Process corresponding action: general result **not worth to re-try**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '3F'. Verify that the default action "quit USAT Interpreter" is executed.

Exception handling:

MEC_TRH:TC 9 Process corresponding action: general result **a value with no configured action**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '00'. Verify that the default action for exception, i.e. "quit USAT Interpreter" is executed.

MEC_TRH:TC 10 Process corresponding action: general result **a value with no configured action**

Render a page with a NU with a UI-CMD: execute USAT command. Let the IUT receive a terminal response with general result value '00'. Verify that the default action for exception, i.e. "quit USAT Interpreter" is executed.

Additional test cases for modified system TRH configuration

MEC_TRH:TC 11 **Several actions** configured

Render a page with a NU with a UI-CMD: execute USAT command.

Let the rendered page have a TRH modifier that configures two actions to the general result **ok**. Let the modifier configure a navigation action in addition to the "process next byte code".

Let the IUT receive a terminal response with general result value '0F'. Verify that the user is given the option to select between the two actions.

Let the IUT receive a menu selection indicating that the temporarily configured navigation action is selected. Verify that the configured navigation action is executed (i.e. the configured NU rendered)

MEC_TRH:TC 12 **Text for user notification**

Render a page with a NU with a UI-CMD: execute USAT command.

Let the rendered page have a TRH modifier that configures a text for user notification for the general result **ok**. (this test case is combined with the previous into one test procedure).

Verify that the test is displayed, before the corresponding action are executed.

6.4 Activation

MEC_ACT

6.4.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified for activating the USAT Interpreter. Specification reference: TS 31.113 clause 4.4.

The following conformance contexts apply to activation:

- Resident
- Remote [CR 4, CR 5]

The Remote conformance context may be divided into the following conformance areas applicable to activation:

- Processing behaviour - no wait state
- Processing behaviour - wait state

6.4.2 Conformance requirements

- MEC_ACT:CR 1 It shall be possible to activate the UI through a UE menu selection
- MEC_ACT:CR 2 It shall be possible to activate the UI through a UE event
- MEC_ACT:CR 3 If an event occurs while the USAT Interpreter is not in idle state, the USAT Interpreter shall queue the event and shall postpone executing the event until the USAT Interpreter enters idle state again.

The USAT Interpreter shall be able to queue at least one event.
- MEC_ACT:CR 4 It shall be possible to activate the UI through a incoming retrieved page
- MEC_ACT:CR 5 It shall be possible to activate the UI through a incoming pushed page
- MEC_ACT:CR 6 Rendering shall be independent of means of activation

6.4.3 Test cases

- MEC_ACT:TC 1 Menu selection.

Let the IUT be in idle state. Let the IUT receive a menu selection indicating a page to render. Verify that the UI is activated and the selected page rendered.
- MEC_ACT:TC 2 UE event

Let the IUT be in idle state. Let the IUT receive an event indicating **location status**. Verify that the page configured for the event is rendered.
- MEC_ACT:TC 3 Verify identical rendering for different methods of activation

Let the TC1 and TC 2 initiate rendering of the same resident page. Verify that the rendering is "identical"...
- MEC_ACT:TC 4 Queue one event in Rendering state

Let the IUT be in "Rendering a page" state. (e.g. wait for a user to clear a displayed text). Let the IUT receive an event to be queued (the location status event from TC 2). Then let the IUT receive the response clearing the message.

Verify that the page configured for the queued event is rendered.

? good?

Additional test cases for Remote conformance context:

- MEC_ACT:TC 5 Incoming retrieved page

Render a page that initiates a pull request for a remote page, without wait state. Verify that the UI is activated and renders the page when it is received.
- MEC_ACT:TC 6 Incoming pushed page. Verify that the UI is activated and renders the pushed page.

For Remote conformance context with wait state processing:

- MEC_ACT:TC 7 Queue one event in Wait state

Be in wait state, queue up one event. Verify that the event triggers activation after the wait state, as soon as it goes into idle mode..
- MEC_ACT:TC 8 Optional: Queue two events in Wait state

Be in wait state, queue up two events. Verify that both events triggers activation after the wait state, as soon as it goes into idle mode.. (Optional)

6.5 Variable Management

MEC_VAR

6.5.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the variable management. Specification reference TS 31.113 clause 6.

6.5.2 Conformance requirements

The conformance requirements for variable management is split into the two conformance areas

- Usage areas
- Variable substitution.

6.5.2.1 Conformance requirements for Usage areas

MEC_VAR_AREA

Environment variables

MEC_VAR_AREA:CR 1 Environment area – UI system info

- Read – shall be accessible to any page

MEC_VAR_AREA:CR 2 Environment area – USIM issuer info

- Write shall be possible by USIM issuer by admin means (UI administration means according to 31.114) <test spec reference to admin test case>
- Read - shall be accessible to any page

MEC_VAR_AREA:CR 3 Environment area – end user info

- Read shall be possible to any page

Permanent variables

MEC_VAR_AREA:CR 4 Permanent area

- Variables shall be accessible after USIM reset
- Each entry consists of the service ID of the page storing the variable in this area, the variable ID and the content of the variable.
- Any page which provides a service ID may store permanent variables.
- Read shall be possible for page with matching **Service id**

Temporary variables

MEC_VAR_AREA:CR 5 Temporary area

- To keep or not to keep feature (specified with page attribute) clause 6.1.3
- Read shall be possible for current page (i.e. same as wrote it)
- Read of locked-by-preceding-page variables shall be possible according to KeepAll/OTP/KeepAlive combination clause 6.1.3.1.

- If a page does not verify the One Time Password set by the previous page, it may not read the temporary variables.

Page string

No CRs

6.5.2.2 Conformance requirements for Variable substitution

MEC_VAR_SUBST

MEC_VAR_SUBST:CR 1 Method 1

Refer to DS_INL:CR 1

MEC_VAR_SUBST:CR 2 Method 2

Refer to MEC_COM:CR 1

6.5.3 Test cases

6.5.3.1 Test cases for usage areas – environment variables

System info

MEC_VAR_AREA:TC 1 Read variable from the USAT Interpreter system information partition. Verify that the variables specified have some information stored after personalization, one test case for each (00 – 13 are listed in 6.1.1.1)

MEC_VAR_AREA: TC 2 Attempt to (over)write an environment variable in the system information partition. Verify that the variable value has not been modified.

Issuer info

MEC_VAR_AREA:TC 3 Read variable from the USIM Issuer information partition. Verify that data may be freely accessed by any page, variable ids 14 – 28

End user info

MEC_VAR_AREA:TC 4 Read User E-mail address and user name, verify by comparing with IUT specification.

6.5.3.2 Test cases for usage areas – Permanent variables

MEC_VAR_AREA_PERM:TC 1 Write a permanent variable with a Service Id, verify that the variable may be read by another page providing the Service Id.

MEC_VAR_AREA_PERM:TC 2 Let a page with the wrong Service Id attempt to read the variable stored in TC1. Verify that a permanent variable may not be read with wrong Service Id. Verify that “Reference to undefined” is generated by the USAT Interpreter.

MEC_VAR_AREA_PERM:TC 3 Let a page with no Service Id provided, attempt to read the variable stored in TC1. Verify that a “Security error” is generated by the USAT Interpreter.

MEC_VAR_AREA_PERM:TC 4 Reset the USIM. Let a page with the correct Service Id read the variable stored in TC1. Verify that the variable is still readable.

MEC_VAR_AREA_PERM:TC 5 Write to the same variable id as in TC1, but with another Service Id. Let the value stored be different than in TC1. Verify that the two different Service Id's may share the same variable id.

6.5.3.3 Test cases for usage areas – Temporary variables

MEC_VAR_AREA_TEMP:TC 1 Write, then read from within the current page

Write, then read from next page – KeepAll/OTP/KeepAlive combination variants I to VIII according to table in clause 6.1.3.1. Summary of table:

Variant	KeepAll flag	OTP TLV	KeepAlive TLV	UI Behaviour
I	set	present	present	KeepAll attribute shall be ignored "not valid"
II	set	present	not present	
III	set	not present	present	KeepAll attribute shall be ignored variables listed in the Keep Alive List TLV shall be kept for the following page "not valid"
IV	set	not present	not present	
V	not set	present	present	
VI	not set	present	not present	no variables to be kept for the following page "not valid"
VII	not set	not present	present	
VIII	not set	not present	not present	no variables to be kept for the following page

MEC_VAR_AREA_TEMP:TC 2 Write a variable to keep locked. Verify that the variable is accessible to a next page if correct PUC.

Write page : Variant I

Read page: correct PUC

MEC_VAR_AREA_TEMP:TC 3 Write a variable to keep locked. Verify that the variable is not accessible to a next page if incorrect PUC.

Write page : Variant I

Read page: incorrect PUC

MEC_VAR_AREA_TEMP:TC 4 Write a variable to keep locked. Verify that the variable is accessible to the next page if correct PUC.

Write page : Variant II

Read page: correct PUC

MEC_VAR_AREA_TEMP:TC 5 Write a variable to keep locked. Verify that the variable is not accessible to the next page if incorrect PUC.

Write page : Variant II

Read page: incorrect PUC

MEC_VAR_AREA_TEMP:TC 6 Write a variable to keep. Verify that the variable is accessible to the next page.

Write page : Variant III

Read page: PUC not provided.

MEC_VAR_AREA_TEMP:TC 7 Write a variable to keep. Verify that the variable is accessible to the next page.

Write page : Variant IV

Read page: PUC not provided

MEC_VAR_AREA_TEMP:TC 8 Write a variable to keep locked. Verify that the variable is accessible to the next page if correct PUC.

Write page : Variant V

Read page: Correct PUC

MEC_VAR_AREA_TEMP:TC 9 Variant of MEC_VAR_AREA_TEMP:TC 7

Write a variable to keep locked. Verify that the variable is not accessible to the next page if incorrect PUC.

Write page : Variant V

Read page: Incorrect PUC

MEC_VAR_AREA_TEMP:TC 10 Write a variable that will not be kept. Verify that the variable is not accessible to the next page (even if correct PUC).

Write page : Variant VI

Read page: Correct PUC

MEC_VAR_AREA_TEMP:TC11 Write a variable to keep. Verify that the variable is accessible to the next page.

Write page : Variant VII

Read page: PUC not provided

MEC_VAR_AREA_TEMP:TC12 Write a variable that will not be kept. Verify that the variable is not accessible to the next page.

Write page : Variant VIII

Read page: PUC not provided.

6.5.3.4 Test cases for variable substitution – method 1

6.5.3.5 Test cases for variable substitution – method 2

6.6 History list

MEC_HIST

6.6.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified for handling the history list. Specification reference TS 31.113 clause 4.6.

The history list mechanisms may be verified for conformance in resident or remote conformance context:

- Resident [CR 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Remote [CR 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

The following conformance areas apply to history list handling:

- History List handling disabled [CR 4, 9]
- History list handling enabled [CR 1, 2, 3, 5, 6, 7, 8, 9, 10, 11]

6.6.2 Conformance requirements

- MEC_HIST:CR 1 An anchor reference is added to the history list when the corresponding Navigation Unit has been completely rendered.
- MEC_HIST:CR 2 The anchor reference of a Navigation Unit is not added in the history list if the "DoNotHistorize" attribute flag of the Navigation Unit is set.
- MEC_HIST:CR 3 The anchor reference of a Navigation Unit is not added in the history list if the Navigation Unit does not have any Anchor TLV.
- MEC_HIST:CR 4 If the maximum number of entries in the history list is zero, the history list mechanism and related navigation actions become deactivated.
- MEC_HIST:CR 5 If the history list is full, the bottom-most entry is removed from the list in order to free space for a new top-most entry.
- MEC_HIST:CR 6 The history is reset (is emptied) whenever the USAT Interpreter is initialised.
- MEC_HIST:CR 7 The action "Go Back one entry in history list" means that the navigation unit corresponding to the pointed anchor shall be rendered, and the anchor pointer is immediately moved down in the list. The origin of this action can be either the system action '02' in terminal response handler configuration, or the "Go Back" byte code command.
- MEC_HIST:CR 8 If the anchor-pointer reaches the bottom of the history list or the history list does not contain any entry, and if a "Go Back" action has to be performed in this situation, then the exception case of the terminal response handler mechanism shall be performed.
- MEC_HIST:CR 9 Retry-last-proactive-command, system action '03' of the terminal response handler configuration shall not modify the history list.
- MEC_HIST:CR 10 If, at any time, the anchor-pointer does not point to the top-most anchor in the history list, and if a navigation action other than "Go Back" (e.g. Assign and Branch) is performed, then any anchors between the anchor-pointer and the top-most entry are deleted from the history list, that means the entry referenced by the anchor-pointer becomes the top-most entry in the history list.
- MEC_HIST:CR 11 If the USAT Interpreter does not find the requested anchor locally while processing a "Go Back" action, an outgoing message shall be sent to the external system entity to retrieve the page the requested anchor belongs to. The Submit TLV shall be formatted in the same way as the previously used Submit TLV to retrieve this page and the USAT Interpreter shall start to render the navigation unit the anchor points to.

6.6.3 Test cases

History List handling disabled:

- MEC_HIST:TC 1 Render a page containing a "Go Back" byte-code. Verify that appropriate exception/system action '01' is executed.
- MEC_HIST:TC 2 Perform the system action '02' defined in the system TRH configuration. Verify that system action '01' is executed.
- MEC_HIST:TC 3 Perform the system action '03' defined in the system TRH configuration. Verify that the IUT issues the last proactive command again.

History List Handling enabled, Resident conformance context (and “No caching mechanism available”)

MEC_HIST:TC 4 User interaction navigate back

Render a NU that gets historized. Render another NU. Navigate backwards, via user interaction backward move (i.e. back button on the ME). Verify that the previous NU is rendered.

MEC_HIST:TC 5 Do not historize

Refer to DS_NU:TC 4

Consider a page with 3 NU inside. Each NU has got an anchor TLV. Render the first NU (NU1). Render the second NU whose “DoNotHistorize” flag is set (NU2). While rendering the third NU (NU3), navigate backward. Verify that the first NU (NU1) is rendered.

MEC_HIST:TC 6 Historize

Refer to DS_NU:TC 3

Consider a page with 3 NU inside. Each NU has got an anchor TLV. Render the first NU (NU1). Render the second NU whose “DoNotHistorize” flag is reset (NU2). While rendering the third NU (NU3), navigate backward. Verify that the second NU (NU2) is rendered.

MEC_HIST:TC 7 No anchor in NU

Consider a page with 3 NU inside. Render the first NU (NU1) which has an anchor TLV. Render the second NU (NU2) which does not have any anchor TLV. While rendering the third NU (NU3), navigate backward. Verify that the first NU (NU1) is rendered.

MEC_HIST:TC 8 Remove bottom-most entry

Let’s assume that the maximum number of entries in the history list is 3. Consider a page with 5 NU. Each NU can be historized. Render NU1, NU2, NU3 and NU4. While rendering NU5, navigate backward three times. Check that NU2 is rendered. Navigate backward again. Verify that the exception case is performed (quit the USAT Interpreter).

MEC_HIST:TC 9 Reset history list

Initialise the USAT Interpreter (by resetting the USIM for instance). Render the first NU of a page. Navigate backward. Verify that the exception case is performed (quit the USAT Interpreter).

MEC_HIST:TC 10 Execute USAT Interpreter command: Go Back

Refer to UI-CMD_BAK:TC 1

MEC_HIST:TC 11 Consider a page with 2 NU inside. Render the first one. While rendering the second NU, perform the system action ‘03’. The last proactive command is issued again. Navigate backward. Verify that the first NU is rendered.

MEC_HIST:TC 12 Consider a page with 3 NU inside and a TRH modifier at the page level which adds a navigation action. Render the 2 first NU (NU1 and NU2). While rendering the third NU (NU3), navigate backward and render NU2. While rendering NU2, do a branch to another page using a navigation action of the current TRH configuration. While rendering the first NU of this new page, navigate backward. Verify that NU2 is rendered. Navigate backward again. Verify that NU1 is rendered.

Additional test case for Remote conformance context

MEC_HIST:TC 13 Remote entry in history list

While rendering a page, the IUT performs a navigation action to a distant page. An outgoing message with a submit TLV is issued by the IUT. While rendering the first NU of the received

page, navigate backward. Verify that the IUT issues an outgoing message with exactly the same submit TLV content as previously.

7 Test specification for 31.113 : USAT Interpreter data structures (DS)

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structures specified for data structures. Specification reference: TS 31.113 clause 7

7.1 Page

DS_PAGE

7.1.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structures specified for Page.

Specification reference: TS 31.113 clause 7.1

The following conformance contexts apply to page:

- Resident
- Remote

The test cases are divided into the conformance areas:

- Keeping variables
- Caching
- Modified system TRH configuration

And DCS handling... FFS

7.1.2 Keeping variables

DS_PAGE_VAR

7.1.2.1 Conformance requirements

DS_PAGE_VAR:CR 1 The USATI implementation shall keep or not keep variables available to a following page according to Page UI-TLV attribute.

Specification reference: 31.113 clause 7.1.1, 7.1.3, 7.1.4, 7.1.5 and 6.1.3

Test group reference: **MEC_VAR_AREA Temporary variables**

7.1.2.2 Test cases for Keeping variables

Refer to test cases for MEC_VAR_AREA temporary.

7.1.3 Caching

DS_PAGE_CACH

7.1.3.1 Conformance requirements

DS_PAGE_CACH:CR 1 The USAT Implementation shall cache or not cache a page according to Page UI-TLV attribute.

Details: UI may cache the page if page attribute bit b2 is not set, UI shall not cache the page if page attribute bit b2 is set.

7.1.3.2 Test cases for Caching

DS_PAGE_CACH:TC 1 Caching allowed retrieve a remote page that has cache attribute = 0, the page may be cached or not cached.

Verify FFS

DS_PAGE_CACH:TC 2 Caching not allowed

Retrieve a remote page that has the caching attribute bit set, verify that the page is not accessible with resident navigation.

7.1.4 Modified system TRH configuration

DS_PAGE_TRH

7.1.4.1 Configuration requirements

DS_PAGE_TRH:CR 1 The current TRH configuration can be modified temporarily by the modifier (terminal response handler modifier UI-TLV)

DS_PAGE_TRH:CR 2 If a syntax error or a logical error occurs in the terminal response handler modifier, the current terminal response handler configuration remains unchanged.

DS_PAGE_TRH:CR 3 Terminal response handler modifier operations. Specification reference: 31.113 clause 7.1.8.1

DS_PAGE_TRH:CR 4 Modification context: All terminal response handler modifications are valid only within the context they have been introduced.

DS_PAGE_TRH:CR 5 Text for user notification: This text is displayed by a DISPLAY TEXT command whenever a general result in response to a proactive command is received : If a Terminal Response Handler modifier contains a text for user notification TLV, then the text is handled by the USAT Interpreter according to the operation descriptions in clause 7.1.8.1.

After this DISPLAY TEXT command has been issued by the USAT Interpreter the actions defined for the general result are to be handled regardless of the general result of the DISPLAY TEXT command itself.

DS_PAGE_TRH:CR 6 In case of several actions (action list) assigned to the same general result, a SELECT ITEM command shall be constructed by the USAT Interpreter using the corresponding action descriptions as items.

Specification reference: TS 31.113 clause 7.1.8.4.4 and test group reference MEC_TRH

7.1.4.2 Test cases for Modified TRH configuration

CR 1 and CR 4

DS_PAGE_TRH:TC 1 Modification context: page and NU – action: branch to another NU

Render a page with a NU with a UI-CMD: execute USAT command.

Let the TRH modifier of the page specify: For general result **quit** – action Anchor TLV for a resident page. Let the NU attribute *TerminalResponseHandlerConfigurationInheritance* be 0, indicating that the page modifier shall apply for the NU.

Let the IUT receive a terminal response with general result value ‘10’.

Verify that the temporarily configured action is executed, i.e. the page rendered.

DS_PAGE_TRH:TC 2 Modification context: Page but not NU

Repeat TC1, but let the *TerminalResponseHandlerConfigurationInheritance* be 1, indicating that the page modifier shall NOT apply for the NU. Instead the system TRH configuration apply for the NU.

Verify that the default action for **quit** is executed, i.e. “quit USAT Interpreter”

DS_PAGE_TRH:TC 3 Modification context: NU – action branch to another NU

Repeat TC 2, but add a TRH modifier on NU level. Let the NU modifier indicate a another page to branch to.

Verify that that the page indicated in the NU TRH modifier is branched to and rendered next.

DS_PAGE_TRH:TC 4 Repeat TC 1, replace action to: branch to another page

DS_PAGE_TRH:TC 5 Repeat TC 1, replace action to: Execute Native command byte code

DS_PAGE_TRH:TC 6 Repeat TC 1, replace action to: Execute Display Text byte code

Execution handling – next byte code within NU

in the TRH modifier – Let the Action TLV attribute “Execution handling” be 0.

Verify that after the configured action has been executed, the next UI-CMD of the NU is executed. The NU has only one UI-CMD so that means – the no more byte code exception applies. I.e. verify that “quit USAT Interpreter” is executed.

DS_PAGE_TRH:TC 7 Repeat TC 1, replace action to: Execute Display Text byte code

Execution handling – execute USAT command again

in the TRH modifier – Let the Action TLV attribute “Execution handling” be 1.

Verify that after the configured action has been executed, the USAT command is executed again.

DS_PAGE_TRH:TC 8 Repeat TC 1, replace action to: Set Variable byte code

DS_PAGE_TRH:TC 9 Repeat TC 1, replace action to: Execute USAT command byte code

CR 2

DS_PAGE_TRH:TC 10 Syntax error in modifier

Modification context: page and NU – action: branch to another NU

Render a page with a NU with a UI-CMD: execute USAT command.

Let there be a syntax error in this modifier! Let the TRH modifier of the page specify: For general result **help request** – action Anchor TLV for a resident page. Let the NU attribute *TerminalResponseHandlerConfigurationInheritance* be 0, indicating that the page modifier shall apply for the NU.

Let the IUT receive a terminal response with general result value ‘13’.

Verify that the temporarily configured action is NOT executed, instead the default action for help request shall be executed, i.e. “retry last proactive command within current USAT Interpreter navigation unit”.

CR 3

DS_PAGE_TRH:TC 11 Replace operation

DS_PAGE_TRH:TC 12 Add/Append operation

CR 5

DS_PAGE_TRH:TC 13 Text for user notification , refer to MEC_TRH:TC 12

DS_PAGE_TRH:TC 14 Text for user notification – display text TR not ok

CR 6

Need any more test cases for several actions?? MEC_TRH:TC 11

7.2 Navigation Unit

DS_NU

7.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for the Navigation Unit UI-TLV. Specification reference: TS 31.113 clause 7.2

The CRs defined corresponds directly to the functionality described in the NU attributes specification.

7.2.2 Conformance requirements

DS_NU:CR 1 ResetVar

DS_NU:CR 2 DoNotHistorize

DS_NU:CR 3 Chaining mechanism, i.e. execute exception action or render next NU

DS_NU:CR 4 Inherit TRH configuration, or not, from surrounding page context

7.2.3 Test cases

DS_NU:TC 1 Keep variables from previous NU

Render a page with two NUs. Let the first NU be chained with the second, and let the ResetVar attribute be 0. Let the first NU assign a value to a temporary variable.

Verify that variables are kept.

DS_NU:TC 2 Reset temporary variables

Render a page with two NUs. Let the first NU be chained with the second, and let the ResetVar attribute be 1. Let the first NU assign a value to a temporary variable.

Verify that variables are reset when rendering the second NU.

DS_NU:TC 3 Render page with NU indicating “historize”, navigate backwards, verify that the previous NU is rendered

- DS_NU:TC 4 Render page with NU indicating “do not historize”, navigate backwards, verify that previous NU is not rendered again
- DS_NU:TC 5 Chaining – execute exception action.
- The IUT is configured with the default system TRH configuration.
- Render a page with a NU with no branch instruction in NU, and indicating ChainNextNu not set.
- Render a page with a NU without branch instruction. After the page has been rendered, the “no more byte code” exception applies and the corresponding action of the system TRH configuration shall be executed.
- Verify that the system action ‘01’ is executed (i.e. quit USAT interpreter)
- DS_NU:TC 6 Chaining – render next Nu in page
- Let the IUT render a page with two NUs, where the first contains no branch instruction. Let the first NU indicate ChainNextNU = 1.
- Verify that the second NU is rendered.

Additional test cases for modified TRH configuration:

- DS_NU:TC 7 Inherit TRH configuration from page. Refer to DS_PAGE_TRH:TC 1
- DS_NU:TC 8 Do not inherit page TRH configuration. Refer to DS_PAGE_TRH:TC 2
- DS_NU:TC 9 Do not inherit, use NU level modifier. Refer to DS_PAGE_TRH:TC 3

7.3 Anchor Reference

DS_ANC

The anchor reference data structure will be part of a TC for navigation, both local and remote navigation within a service.

7.3.1 Definition and applicability

- Applicable in Resident conformance context.

The anchor reference data structure may be contained in

- Action TLV, of the TRH modifier, if the action specifies the action to be performed as a navigation action.
- Page Reference TLV, if referring to an anchor of a NU in a resident page or in the current page

7.3.2 Conformance requirements

- DS_ANC:CR 1 If the page identification part is omitted the reference is to an anchor on the current page.
- DS_ANC:CR 2 If the anchor name part is omitted the reference is to the first navigation unit of the referenced page.

7.3.3 Test cases

Refer to test cases for Assign and Branch , and modify the values of Anchor Reference

7.4 Variable Identifier List

DS_VIT

7.5 Inline Value

DS_INL

7.5.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to interpret the data structure Inline value as specified in TS 31.113 clause 7.5

7.5.2 Conformance requirements

DS_INL:CR 1 The possibly available constant data values and variable references have to be rendered according to clause 6.3 Method 1 during processing of this TLV by the USAT Interpreter.

DS_INL:CR 2 UI shall perform variable value type conversion. If the type of the possibly substituted variable values is different from the type indicated in the attribute of this TLV, the USAT Interpreter shall perform a type conversion or generate an error according to table: 31.113 clause 7.5

7.5.3 Test cases

Test sequence where variable substitution is taking place, invoking type conversion according to the table in 7.5

7.6 Inline Value 2

DS_INL2

This UI-TLV is included in the UI-CMD Get Input and the UI-TLV Ordered TLV List. Type conversion tests, as for Inline Value, applies to this test group too.

7.7 Input list

DS_INPL

This UI-TLV is included in the UI-CMD Execute Native Command.

7.8 Ordered TLV list

DS_OTL

This UI-TLV is included in the UI-CMDs Assign and Branch and Branch on Variable Value

7.9 Page reference

DS_PREF

7.9.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structures specified for Page Reference. Specification reference: TS 31.113 clause 7.9

The CRs are applicable to the conformance contexts:

- Resident
- Remote

7.9.2 Conformance requirements

- DS_PREF:CR 1 If the Anchor Reference TLV or the Variable Identifier List TLV is available, then the USAT Interpreter shall start rendering the requested locally stored Anchor. If the Anchor is not found locally, an error shall be generated.
- Test spec reference: MEC_NAV
- DS_PREF:CR 2 If the Submit Configuration TLV is available (that indicates that the page is not locally stored on the USIM, i.e. e.g. stored at an external system entity), then the USAT Interpreter shall build a request to the external system entity according to clause 7.10 .
- DS_PREF:CR 3 In case of remote branching, the UI shall include sending page reference depending on the attribute b1 of the Submit configuration UI-TLV included in a Page reference UI-TLV
- Specification reference: TS 31.113 clause 7.9.3.1
- DS_PREF:CR 4 In case of remote branching, the UI shall enter wait state or keep on rendering next page depending on the attribute b2 of the Submit configuration UI-TLV included in a Page reference UI-TLV.
- Specification reference: TS 31.113 clause 7.9.3.1
- DS_PREC:CR 5 If the transmission to the external system entity fails, an USAT Interpreter transmission error shall be generated by the USAT Interpreter and the execution shall stop.

7.9.3 Test cases

Render a page with a Branch on Variable Value UI-CMD.

- DS_PREF:TC 1 Navigate to local NU.
- A Page reference UI-TLV has an Anchor reference UI-TLV in it, where the anchor is present locally, and the navigation to it is performed correctly. Verify that the correct NU is rendered.
- DS_PREF:TC 2 Navigate to local NU, unknown NU
- A Page reference UI-TLV has an Anchor reference UI-TLV in it, where the anchor is NOT present locally, an error shall be generated by UI. Verify that an error is generated...?

Additional test cases for remote conformance context:

- DS_PREF:TC 3 Navigate to remote page, include sending page reference.
- Verify that the sending page reference is sent in the Pull message.
- DS_PREF:TC 4 Navigate to remote page, do not include sending page reference.
- Verify that the sending page reference is not sent.
- DS_PREF:TC 5 Navigate to remote page, enter wait state until the page has been retrieved. Verify that the UI is in wait state until the retrieved page has arrived (i.e. the next NU should not be rendered until wait state is exited)
- DS_PREF:TC 6 Navigate to remote page, do not enter wait state, instead render next "byte code" . Verify that the "next byte code " is rendered.
- DS_PREF:TC 7 Transmission to external system fails

Render a page with a branch instruction to a remote page. Let the IUT be disconnected from the test environment, so that it may not communicate with the GW simulator.

Verify that USAT Interpreter transmission error is generated...??? How?

7.10 Submit

DS_SUB

This UI-TLV is included in USAT Interpreter originated pull messages. Refer to TPT_UI-APP_PULLREQ and DS_PREF:TC 3 and TC4.

8 Test specification for 31.113 : USAT Interpreter commands (UI-CMD)

8.1 Set Variable

UI-CMD_SETV

8.1.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for the UI-CMD Set Variable. Specification reference: TS 31.113 clause 8.1.

8.1.2 Conformance requirements

UI-CMD_SETV:CR 1 This byte code sets one or more variables either to a value contained in the corresponding Inline Value TLV or to the concatenated contents of the referenced variables in the Variable Identifier List TLV.

8.1.3 Test cases

UI-CMD_SETV:TC 1 Render a page that has a NU with this UI-CMD. Copy the content of one variable to another variable. Verify that the variable have been set to the correct value.

UI-CMD_SETV:TC 2 Render a page that has a NU with this UI-CMD. Concatenate a list of variable (say two) into another variable. Verify that the variable have been set to the correct value.

UI-CMD_SETV:TC 3 ...Copy several variables...

UI-CMD_SETV:TC 4 ...Attempt to copy the content of a non-existing variable., Verify that the correct error Code is issued (Reference to undefined).

8.2 Assign And Branch

UI-CMD_AAB

8.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for the UI-CMD Assign and Branch. Specification reference: TS 31.113 clause 8.2.

8.2.2 Conformance requirements

- UI-CMD_ABB:CR 1 This byte code displays a menu on the UE and assigns a selected value to a variable according to the selection of the user.
- UI-CMD_ABB:CR 2 General variable assignments and navigation operations may be performed by the "Assign and Branch" byte code dependent on the data provided in the Ordered TLV List TLV.

8.2.3 Test cases

- UI-CMD_ABB:TC 1 "Display, Assign and Branch"
- Render a page that has three NUs, first NU with one UI-CMD Assign and Branch that provides the user with an opportunity to select between two colours, RED and BLUE.
- Verify that the two colours are displayed for user selection
- Let the colour red be selected.
- Let there be two more NUs in the page, one "red NU" and one "blue NU". Let the red NU display a text to verify that this is the red NU being rendered, and then display the value that shall have been assigned to the destination variable.
- Verify that the "red NU" is rendered.
- Verify that the value for "red" has been assigned to the destination variable.
- UI-CMD_ABB:TC 2 "Set Variable Selected"
- This test case is a simplified version of UI-CMD_ABB:TC 1.
- Render a page with one NU with a UI-CMD Assign and Branch that provides the user to select between two colours, RED and BLUE. The NU also have a second UI-CMD DisplayText that displays the value assigned to the destination variable.
- Let the colour BLUE be selected.
- Verify that the value for the BLUE option has been assigned to the destination variable.
- UI-CMD_ABB:TC 3 "Go Selected"
- This test case is a simplified version of UI-CMD_ABB:TC 1
- Render a page with three NUs. Let the first NU specify an Assign and Branch UI-CMD that provides the user with the opportunity to select between two colours, RED and BLUE.
- Let there be two more NUs in the page, one "red NU" and one "blue NU". Let these NUs have a displayText UI-CMD each to indicate rendering.
- Verify that the two colours are displayed for selection.
- Let the colour BLUE be selected.
- Verify that the UI branches to the BLUE NU by verifying that the "BLUE text" is displayed.
- UI-CMD_ABB:TC 4 "Display and Process next byte code"
- Render a page with one NU with a UI-CMD Assign and Branch that provides the user to select between two colours, RED and BLUE. The NU also have a second UI-CMD DisplayText.
- Let the colour RED or BLUE be selected.

Verify that the second UI-CMD is rendered next.

UI-CMD_ABB:TC 5 “Assign and Branch”

Render a page with two NUs. Let the first NU specify an Assign and Branch UI-CMD and the second NU a DisplayText UI-CMD that displays the value assigned to the destination variable.

Verify that the UI branches to the specified NU, and that the correct value has been assigned to the destination variable.

UI-CMD_ABB:TC 6 “Set Variable”

Render a page with one NU. Let the NU specify two UI-CMDs, the first an Assign and Branch and the second a Display Text. Let the Display Text UI-CMD display the value assigned to the destination variable of the Assign and Branch UI-CMD.

Verify that the correct value has been assigned to the destination variable.

UI-CMD_ABB:TC 7 “Direct Go” – Navigation with a branch instruction

Render a page that holds the UI-CMD Assign and Branch with a branch instruction to another page, verify that the page branched to is rendered next (i.e. navigated to).

Error cases:

UI-CMD_ABB:TC 8 No response from user

This test case is a variant of UI-CMD_ABB:TC 3. Let there be no response from the user on the SELECT Item USAT command.

Verify that the UI exits.

Inconsistency in Ordered TLV list TLV’s composition:

Let an Assign and Branch UI-CMD have two Ordered TLV list TLVs that have different composition, so that they indicate two different kind of functionality of the Assign and Branch UI-CMD:

UI-CMD_ABB:TC 9 One Ordered TLV list TLV with Inline Value 2 present, one without.

Verify that the UI ignores the second Ordered TLV list TLV

UI-CMD_ABB:TC 10 Two Ordered TLV list TLVs without Inline value 2.

Verify that the UI ignores the second one.

8.3 Extract

UI-CMD_EXTR

8.3.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for the UI-CMD Extract. Specification reference: TS 31.113 clause 8.3.

8.3.2 Conformance requirements

UI-CMD_EXTR:CR 1 This byte code extracts a byte array from a value and stores the result in a variable.

8.3.3 Test cases

- UI-CMD_EXTR:TC 1 Let the UI-CMD specify the specific bytes to extract. Verify that the bytes have been extracted from variable 1 to variable 2.
- UI-CMD_EXTR:TC 2 Let the UI-CMD specify that all remaining bytes shall be extracted. Verify that the bytes have been extracted from variable 1 to variable 2.
- UI-CMD_EXTR:TC 3 Let the UI-CMD specify extraction from a non-existing variable. Verify that the correct error message is issued.

Verify all the error actions by invoking the errors...FFS

8.4 Go Back

UI-CMD_BAK

8.4.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the UI_CMD Go back mechanisms and data structure.
Specification reference: TS 31.113 clause 8.4

The specified CR is applicable to both Remote and Resident conformance context.

8.4.2 Conformance requirements

- UI-CMD_BAK:CR 1 When rendering UI-CMD Go Back the UI shall render a NU as indicated by go back attribute bit b1. That is, either the current NU is re-rendered or branching to the last anchor pushed on the history list is initiated.

8.4.3 Test cases

- UI-CMD_BAK:TC 1 Render a page with a NU to historize, then one with a UI-CMD go back with b1 = 0, i.e. back to previous NU anchor according to history list. Verify that previous NU is rendered.
- UI-CMD_BAK:TC 2 UI-CMD go back with b1 = 1, i.e. current NU is re-rendered. Verify that the current NU is re-rendered.
- UI-CMD_BAK:TC 3 UI-CMD go back with b1 = 0, back to previous NU anchor according to history list when no anchor in history list. Verify that Note: update with the new history list spec

Additional remote conformance context:

- UI-CMD_BAK:TC 4 UI-CMD go back, after that a remote non-cached NU has been historized. Verify that the retrieval of the page with this NU is initiated by the UI

8.5 Branch On Variable Value

UI-CMD_BOVV

8.5.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the UI_CMD BOVV mechanism and data structure specified in TS 31.113 clause 8.5.

These tests cover the “matching” part of the mechanism not the actual navigation mechanisms that are handled in the MEC_NAV test group. It should be enough to run tests in the resident conformance context.

8.5.2 Conformance requirements

UI-CMD_BOVV:CR 1 Branch on variable value mechanism

8.5.3 Test cases

UI-CMD_BOVV:TC 1 Match found, verify that match local page is executed

UI-CMD_BOVV:TC 2 No match found, verify that the no-match local page is executed

UI-CMD_BOVV:TC 3 No match found, no-match reference page not given, verify that "next byte code" is rendered.

8.6 Exit

UI-CMD_EXIT

8.6.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanism and data structure specified for UI_CMD Exit. Specification reference: 31.113 clause 8.6

8.6.2 Conformance requirements

UI-CMD_EXIT:CR 1 When Exit UI-CMD is rendered , the attribute bit b1 determines whether the UI shall exit or return to TRH.

8.6.3 Test cases

UI-CMD_EXIT:TC 1 Render a page that has a NU with an Exit Ui-CMD in it: Exit immediately. Verify immediate exit.

UI-CMD_EXIT:TC 2 Render a page that has a NU with an Exit Ui-CMD in it: Return to TRH and do the TRH action assigned to "Proactive SIM session terminated by the user". Verify that the configured action is executed.

8.7 Execute USAT Command

UI-CMD_USATC

8.7.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanism and data structure specified for UI_CMD Execute USAT command. Specification reference: 31.113 clause 8.7

8.7.2 Conformance requirements

UI-CMD_USATC:CR 1 A USAT Interpreter shall be able to execute all USAT commands, as specified in 31.111, + some variants in behaviour according to attributes shall be supported.

At least one test case per USAT command according to 31.111:

DISPLAY TEXT

GET INKEY

GET INPUT.....

8.8 Execute Native Command

UI-CMD_ENC

8.8.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms for executing native commands, as specified in TS 31.113 clause 8.8

The following conformance areas are applicable:

- No native commands available
- Native commands available

In the second case, there must be at least one native command available to the IUT.

8.8.2 Conformance requirements

UI-CMD_ENC:CR 1 Execution of native command with either returning or exiting the UI, according to attribute

8.8.3 Test cases

Native command not supported:

Normal case:

UI-CMD_ENC:TC 1 Function call, then return to UI.

Let the Execute Native Command specify an NCI of a not supported Native Command. Let the ExitFlag attribute be not set.

Verify that an error is issued by the UI.

Error case:

UI-CMD_ENC:TC 2 Syntax error

Let the UI-CMD have a syntax error, e.g. a missing mandatory TLV.

Verify that the syntax error is issued by the UI.

Additional test cases for Native commands available:

Refer to test group NC. One normal test case of either supported native command shall be performed.

8.9 Get Length

UI-CMD_GLEN

8.9.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for the UI-CMD Get Length. Specification reference: TS 31.113 clause 8.9.

8.9.2 Conformance requirements

UI-CMD_GLEN:CR 1 This byte code instructs the USAT Interpreter to calculate the length of all variable contents of the variables in the Variable List and to assign the result to the output variable.

8.9.3 Test cases

UI-CMD_GLEN:TC 1 Render the Get Length UI_CMD for a variable of known length. Verify that the correct length is calculated.

UI-CMD_GLEN:TC 2 Render the Get Length UI_CMD for a non-existing variable. Verify that the correct error code is issued (Reference to undefined).

8.10 Get TLV Value

UI-CMD_GTLV

8.10.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for the UI-CMD Get TLV Value. Specification reference: TS 31.113 clause 8.10.

8.10.2 Conformance requirements

UI-CMD_GTLV:CR 1 This byte code instructs the USAT Interpreter to extract the value part of a TLV from a sequence of TLVs and to assign the resulting value to the output variable.

UI-CMD_GTLV:CR 2 If the requested tag value is not found in the sequence of TLVs, the output variable is generated with no content (i.e. the length of content of the variable is 0).

8.10.3 Test cases

UI-CMD_GTLV:TC 1 normal case

UI-CMD_GTLV:TC 2 tag not found

UI-CMD_GTLV:TC 1 variable not found

8.11 Display Text

UI-CMD_DISP

8.11.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for the UI-CMD Display Text. Specification reference: TS 31.113 clause 8.11.

The following conformance areas have been defined for the UI-CMD Display Text:
Normal cases, DCS handling and error handling.

8.11.2 Conformance requirements

- UI-CMD_DISP:CR 1 This byte code instructs the USAT Interpreter to issue a DISPLAY TEXT command according to 3GPP TS 31.111 [1].
- UI-CMD_DISP:CR 2 The USAT Interpreter shall use the DCS value according to the indication given in the attributes of the Inline Value TLV. If no attributes are given in the Inline Value TLV, the coding scheme indication of the current page shall be used.

8.11.3 Test cases

Normal cases:

- UI-CMD_DISP:TC 1 Render a page with a NU with UI-CMD display text: Display text, wait for end user to clear text. Let the DCS of the Inline Value be Unknown. Let the DCS attribute of the page be "SMS default"
- Verify that the specified text is displayed. Verify that the display is kept until the end-user clears it. Verify that the DCS of the page is used in the issued USAT command. Verify that "High Priority" is used in the issued USAT command.
- UI-CMD_DISP:TC 2 Display text, clear message after delay. Verify that the text is cleared after a delay.

DCS handling:

- UI-CMD_DISP:TC 3 Repeat TC1, but let the DCS of the inline value be: SMS default - packed
- UI-CMD_DISP:TC 4 Repeat TC1, but let the DCS of the inline value be: Binary format
- UI-CMD_DISP:TC 5 Repeat TC1, but let the DCS of the inline value be: USC2

Error cases:

- UI-CMD_DISP:TC 6 Let the UI-CMD have a syntax error: the inline value field missing.
- UI_CMD_DISP:TC 7 Let the UI-CMD refer to a variable that does not exist. Verify that the error code "Reference to undefined" issued and the action Stop is performed

8.12 Get Input

UI-CMD_GINP

8.12.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for the UI-CMD Get Input. Specification reference: TS 31.113 clause 8.12.

8.12.2 Conformance requirements

- UI-CMD_GINP:CR 1 This byte code instructs the USAT Interpreter to request multiple character input from user by issue of a GET INPUT command.

8.12.3 Test cases

- UI-CMD_GINP:TC 1 Get Input – SMS default requested, digits input mode
- Let the IUT render a page with a NU with a UI-CMD Get Input.
- Verify that the correct text is issued in the GET INPUT USAT command, and the other specified values for the USAT command.

Let the IUT receive a terminal response with input data. Verify that the contents stored to the destination variable is equal to the input data received.

UI-CMD_GINP:TC 2

Get Input – UCS2 requested, alphabet set

Repeat TC 1 but let the *Requested character set for input* be 1 and the *Requested input mode of UE* be 1.

9 Test specification for 31.113 : Native commands (NC)

This clause is applicable to IUTs supporting the respective native command.

9.1 PKCS#7 Signature Plug-in

NC_P7

9.1.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for the native command P7 Plug-in. Specification reference: TS 31.113 clause 9.1.2.1.

This sub-clause is applicable if the IUT supports the P7 Plug-in.

9.1.2 Conformance requirements

NC_P7:CR 1 The P7 plug-in is used to provide a digital signature based on a private key stored on the USIM. Plug-in execution shall follow specification in Annex D

9.1.3 Test cases

NC_P7:TC 1 P7 sign with default key

Execute the Native command with ExitFlag not set (i.e. as a function call)

Let the input argument built from the following requests:

Precondition /configuration	Default key available PIN	
Parameter	Value	Functional requests
Key identifier type	'00'	'00' = No key identifier supplied. The plug-in shall choose a default key, if such a key exists, or abort with Plug-in Status Code "PS:No such key"
Character encoding scheme	'04'	'04' = GSM default (unpacked)
Options	0000 0000	None
TTBS	The text...	Text to be signed

Verify that execution returns to the UI.

Verify that the native command generates the correct output parameters

NC_P7:TC 2

P7 sign with default key – continuation (Why is this needed???)

Repeat test case NC_P7:TC 1 procedure but let the execute the native command with ExitFlag set (i.e. as a continuation...no return to page)

Let the output variable be permanent variables!

Verify that the execution does not return to the page.

Verify that the native command generates correct output parameters.

NC_P7:TC 3 P7 sign with default key – variant: **Include options**

Let the input argument built from the following requests:

Precondition	Default key available PIN	
Parameter	Value	Functional requests
Key identifier type	'00'	'00' = No key identifier supplied. The plug-in shall choose a default key, if such a key exists, or abort with Plug-in Status Code "PS:No such key"
Character encoding scheme	'04'	'04' = GSM default (unpacked)
Options	0000 0111	Include the following data in the output <ul style="list-style-type: none"> - TTBS - hash of public key - a URL to public key certificate
TTBS	The text...	Text to be signed

Verify that the native command generates the correct output parameters.

NC_P7:TC 4 P7 sign with default key – variant: **Include options**

Let the input argument built from the following requests:

Precondition	Default key available PIN ICCID available	
Parameter	Value	Functional requests
Key identifier type	'00'	'00' = No key identifier supplied. The plug-in shall choose a default key, if such a key exists, or abort with Plug-in Status Code "PS:No such key"
Character encoding scheme	'04'	'04' = GSM default (unpacked)
Options	0011 1000	Include the following data in the output <ul style="list-style-type: none"> - ICCID - Message digest of TTBS - Index of RSA key
TTBS	The text...	Text to be signed

Verify that the native command generates the correct output parameters.

NC_P7:TC 5 P7 sign with default key – variant: **Include all options**

Let the input argument built from the following requests:

Precondition	Default key available PIN ICCID available	
Parameter	Value	Functional requests
Key identifier type	'00'	'00' = No key identifier supplied. The plug-in shall choose a default key, if such a key exists, or abort with Plug-in Status Code "PS:No such key"
Character encoding scheme	'04'	'04' = GSM default (unpacked)
Options	0011 1111	Include all optional data in the output
TTBS	The text...	Text to be signed

Verify that the native command generates the correct output parameters.

NC_P7:TC 6 P7 sign with default key – variant: **no default key available**

Repeat the test case NC_P7:TC1 but let there be no default key available to the IUT.

Precondition	No default key available
---------------------	--------------------------

Verify that the native command generates output parameter "PS:No such key"

NC_P7:TC 7 P7 sign with default key – variant: **user cancel**

Repeat test case NC_P7:TC1 but let the user cancel the plug-in execution.

Verify that the native command generates output parameters "PS: User cancel"

NC_P7:TC 8 P7 sign with default key – variant: **Wrong PIN at first attempt**

NC_P7:TC 9 P7 sign with default key – variant: **Long TTBS**

NC_P7:TC 10 P7 sign with default key – variant: **Character encoding scheme UCS2**

NC_P7:TC 11 P7 sign with **User key hash**

NC_P7:TC 12 P7 sign with **List of trusted key hashes**

NC_P7:TC 13 P7 sign with **Index of RSA key** NC_P7:TC 14 P7 - **Syntax error in input parameters**

Let the Execute native command specify input parameter with errors and ExitFlag not set.

Verify that the execution returns to UI and "syntax error" is issued by the UI.

FFS

10 Test specification for 31.114 : Transmission protocol (TPT)

10.1 Handling of Pull messages

TPT_HAND_PULL

Refer to MEC_NA V and MEC_COM test groups

10.2 Handling of Push messages

TPT_HAND_PUSH

10.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified for how to handle incoming push messages

Specification reference: TS 31.114 clause 5.6

The CR defined for this test group are applicable within the Remote and Administration conformance context.

10.2.2 Conformance requirements

TPT_HAND_PUSH:CR 1 Handling of push messages depending on state according to table in TS 31.114 clause 5.6

10.2.3 Test cases

TPT_HAND_PUSH:TC 1 Send push message to IUT in idle state, verify that the page included in the Push Message is rendered.

TPT_HAND_PUSH TC 2 Send push message to IUT in “about to parse byte code” state, verify that an “ask user” mechanism is executed. Verify both possible alternative answers.

TPT_HAND_PUSH TC 3 Send push message to IUT in “about to parse resident pages” state, verify that an “ask user” mechanism is executed. Verify both possible alternative answers.

TPT_HAND_PUSH:TC 4 Send a push message to IUT in wait state. Verify that the “ask user” mechanism is executed:

- Ask user general result “ok” or “no response from user” → Verify that wait state is cancelled and the pushed page is rendered.
- General result other → Verify that the pushed page is not rendered, instead the wait state continues.

TPT_HAND_PUSH:TC 5 Send a push message to IUT in “TRH execution environment” state. Verify that the “ask user” mechanism is executed.

Additional test case for Administration conformance context:

TPT_HAND_PUSH:TC 6 Send a push message to IUT in “about to parse administrative code” state. Verify that the Administration message is finalised, and after that the pushed page is rendered.

10.3 Handling of Administration messages

TPT_HAND_ADM

10.3.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified for handling of administration messages. Specification reference: TS 31.114 clause 5.7

The CRs defined for this test group are applicable within Administration conformance context.

10.3.2 Conformance requirements

- TPT_HAND_ADM:CR 1 USAT Interpreter in idle mode shall handle Administration messages. If not in idle mode, administration messages shall be discarded.
- TPT_HAND_ADM:CR 2 If an Administration Message is discarded by the USAT Interpreter, no corresponding ADM Response shall be generated by the USAT Interpreter.

10.3.3 Test cases

- TPT_HAND_ADM:TC 1 Send an Administration Message request to IUT in idle mode, verify execution of the ADM request included in the Administration Message. Further details about the verification, refer to other test case for specific admin command.
- TPT_HAND_ADM: TC 2 Send an administration message to IUT in “about to parse byte code” state. Verify that the administration message is discarded.
- TPT_HAND_ADM: TC 3 Send an administration message to IUT in “about to parse resident pages” state. Verify that the administration message is discarded.
- TPT_HAND_ADM: TC 4 Send an administration message to IUT in wait state. Verify that the administration message is discarded.
- TPT_HAND_ADM: TC 5 Send an administration message to IUT in “TRH execution environment” state. Verify that the administration message is discarded.
- TPT_HAND_ADM: TC 6 Send an administration message to IUT in “about to parse administrative code” state. Verify that the administration message is discarded.

10.4 Transport bearer

TPT_BEAR

10.4.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the transport bearer support.

Specification reference: TS 31.114 clause 5.1

The CRs defined for this test group are applicable within the Remote and Administration conformance context.

10.4.2 Conformance requirements

- TPT_BEAR:CR 1 UI security functionality shall support SMS-PP
- TPT_BEAR:CR 2 Concatenation shall be supported
- TPT_BEAR:CR 3 UI security functionality may support other bearers.

10.4.3 Test cases

- TPT_BEAR:TC 1 Retrieve a page over SMS (i.e. branch remote) Refer to MEC_COM:TC 3.
- TPT_BEAR:TC 2 Retrieve a page that is large enough to require concatenation

10.5 23.048 security

TPT_SEC

10.5.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the security parameters to be supported for security packets

Specification reference: TS 31.114 clause 5.2

10.5.2 Conformance requirements

TPT_SEC:CR 1 Command packet structure as specified in TS 23.048 shall be used in both directions

TPT_SEC:CR 2 UI shall support sending PoR by SMS-SUBMIT to the external system upon request

TPT_SEC:CR 3 UI may support sending PoR by SMS-DELIVER-REPORT

TPT_SEC:CR 4 Security mechanisms - Authentication

TPT_SEC:CR 5 Security mechanisms - Encryption

TPT_SEC:CR 6 Security mechanisms - Counter

TPT_SEC:CR 7 UI shall support KIC modes

TPT_SEC:CR 8 UI shall support KID modes

TPT_SEC:CR 9 UI shall support padding

10.5.3 Test cases

TPT_SEC:TC 1

FFS

10.6 Addressing and key selection

TPT_ADDR

10.6.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the addressing and key selection mechanisms.

Specification reference: 3G TS 31.114, clause 5.3

The following conformance contexts apply to this test group:

- Resident [CR2, CR 3]
- Remote [CR 1, CR 2, CR 3; CR 8, CR 9]
- Administration [CR 4, CR 5, CR 6, CR 10]

10.6.2 Conformance requirements

TPT_ADDR:CR 1 The USAT Interpreter security functionality shall check for each USAT Interpreter terminated secured message if the combination of SPI, least significant nibble of KIC and least significant nibble of KID matches a combination included in the list of this entry for the configuration identified by the TAR value. If no match is found, the secured message can not be processed and shall be discarded.

Specification reference 31.114 clause 5.3.1.1.7

- TPT_ADDR:CR 2 If the USAT Interpreter is requested to execute a USAT command not allowed according to the USAT Command list, the USAT Interpreter shall issue the error '6F0A' "USAT command not allowed" .
Specification reference 31.114 clause 5.3.1.1.8
- TPT_ADDR:CR 3 If the USAT Interpreter is requested to execute a USAT Interpreter command not allowed according to the USAT Interpreter byte code list, the USAT Interpreter shall issue the error '6F0A' "USAT command not allowed".
Specification reference 31.114 clause 5.3.1.1.9
- TPT_ADDR:CR 4 If the USAT Interpreter is requested to execute a USAT Interpreter administration command not allowed according to the USAT Interpreter command list for administration, the USAT Interpreter shall issue the error '6F0A' "USAT administration command not allowed"
Specification reference 31.114 clause 5.3.1.1.10
- TPT_ADDR:CR 5 If the USAT Interpreter is requested to modify a not allowed Configuration, the USAT Interpreter shall issue the error '6F0B' "Modify configuration not allowed"
Specification reference 31.114 clause 5.3.1.1.11
- TPT_ADDR:CR 6 If the USAT Interpreter is requested to modify a not allowed bearer data set, the USAT Interpreter shall issue the error '6F0B' "Modify configuration not allowed"
Specification reference 31.114 clause 5.3.1.1.12
- TPT_ADDR:CR 7 Linked pull tar
- TPT_ADDR:CR 8 For Operational Pull messages (i.e. messages with the TAR value in the Operational Pull Mode TAR value range) received by the USAT Interpreter from the USAT Interpreter Gateway System, the USAT Interpreter Security Functionality shall check, if the received TAR value is equal to the TAR value of the current configuration. If the values are not equal, the received message shall be discarded by the USAT Interpreter Security Functionality
Specification reference: 31.114 clause 5.3.1.2
- TPT_ADDR:CR 9 For Operational Push messages received by the USAT Interpreter from the USAT Interpreter Gateway System, the USAT Interpreter Security Functionality shall check, if there is a configuration in the Operational Push Configuration Set, which contains the received TAR value of the message.... If no matching TAR value is found, the received message shall be discarded by the USAT Interpreter Security Functionality.
Specification reference: 31.114 clause 5.3.1.3
- TPT_ADDR:CR 10 For Administration Messages ... received by the USAT Interpreter from the USAT Interpreter Gateway System, the USAT Interpreter Security Functionality shall check, if there is a configuration in the Administration Configuration Set, which contains the received TAR value of the message. ... If no matching TAR value is found, the received message shall be discarded by the USAT Interpreter Security Functionality.
Specification reference: 31.114 clause 5.3.1.4

10.6.3 Test cases

- TPT_ADDR:TC1 Let a combination of SPI, least significant nibble of KIC and least significant nibble of KID that is not found in the list for the TAR used, be sent to the IUT in a USAT Interpreter terminated message. Verify that the message is discarded.
- TPT_ADDR:TC 2 Render a page with the ui-cmd "execute USAT command" containing a USAT command that is not included in the list of allowed USAT commands for a particular TAR. Verify that the not allowed USAT command does not get executed. Verify the error '6F0A' is issued.

TPT_ADDR:TC 3	Render a page with a ui-cmd that is not allowed according to the list "USAT Interpreter byte code filter" for a particular configuration(TAR). Verify that the correct error is issued. Verify that the ui-cmd does not get executed.
TPT_ADDR:TC 4	Let the IUT render an Administration Request containing a adm-cmd that is not allowed according to the USAT command list for administration for a particular TAR. Verify that the adm-cmd does not get executed and that the correct error code (6F0A) is issued.
TPT_ADDR:TC 5	Let the IUT render an Administration Request to "Modify Configuration Data" for a particular TAR that is not allowed to be modified. Verify that the modification is not performed. Verify that the correct error is issued (6F0B)
TPT_ADDR:TC 6	Render an Administration Request with the adm-cmd "Modify Bearer Data" for a particular TAR that is not allowed to modify. Verify that the modification is not executed and that the correct error '6F0B' "Modify configuration not allowed" is issued.
TPT_ADDR:TC 8	Render a page that pulls a page from the gateway using a particular TAR and request ID. Let the gateway send back a page where the request ID matches, but the wrong TAR has been used. Verify that the received message is discarded.
TPT_ADDR:TC 9	Let the IUT receive a push message with a TAR (configuration selected) that does not exist in the IUT. Verify that the push message is discarded.
TPT_ADDR:TC 10	Let the IUT receive an administration message with a TAR (configuration selected) that does not exist in the IUT. Verify that the message is discarded.

10.7 USAT Interpreter originated application messages

TPT_UIO-APP

10.7.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified and data structures for USAT Interpreter originated application messages. Specification reference: TS 31.114 clause 5.4

The following **conformance contexts** apply to this test group:

- Remote [CR 1]
- Administration [CR 2]

10.7.2 Conformance requirements

TPT_UIO-APP:CR 1 UIO BER-TLV for operational pull – submit message data structure
Specification reference: TS 31.114 clause 5.4.1

TPT_UIO-APP:CR 2 UIO BER-TLV for Administration message – ADM response data structure
Specification reference: TS 31.114 clause 5.4.2

10.7.3 Test cases

10.7.3.1 Test cases for Pull request

TPT_UI-APP_PULLREQ

- TPT_UI-APP_PULLREQ:TC 1 Render a page initiating a pull request. Let Request ID be present in the Submit configuration (i.e. wait state processing). Verify that the Request ID UI-TLV is included in the UIO Pull Message.
- Ref to other test case?
- TPT_UI-APP_PULLREQ:TC 2 Render a page initiating a pull request. Let the Gateway address in the Submit configuration UI-TLV be another than “current gateway address”. Verify that the correct Gateway address is included in the UIO Pull Message.
- TPT_UI-APP_PULLREQ:TC 3 Render a page initiating a pull request. Let the Gateway address in the Submit configuration UI-TLV be the same as “current gateway address”. Verify that the Gateway address is not included in the UIO Pull Message.
- TPT_UI-APP_PULLREQ:TC 4 Render a page initiating a pull request. Let there be no Gateway address in the Submit configuration UI-TLV. Verify that the Gateway address is not included in the UIO Pull Message.
- TPT_UI-APP_PULLREQ:TC 5 Render a page initiating a pull request. Let the Gateway Address TLV be present and have the attribute SendAdditionalInformation set. Let the gateway address contained be other than the “current gateway address”. Verify that Additional info is present. (clause 5.4.1.4) in the UIO Pull Message.

10.7.3.2 Test cases for Adm response

TPT_UI-APP_ADMRSP

- TPT_UI-APP_ADMRSP:TC 1 Render an Administration request that contains the Request ID UI-TLV. Verify that the Request ID is mirrored back in the UI originated Administration message.

10.8 Gateway originated application messages (GWO-APP)

TPT_GWO-APP

10.8.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms specified and data structures for gateway originated application messages. Specification reference: TS 31.114 clause 5.5

The following **conformance contexts** apply to this test group:

- Remote [CR 1, CR 2]
- There might be a need to separate into pull and push?
- Administration [CR 3]

10.8.2 Conformance requirements

- TPT_GWO-APP:CR 1 GO pull response BER-TLV data structure
Specification reference: TS 31.114 clause 5.5.1
- TPT_GWO-APP:CR 2 GO push (initiation) BER-TLV data structure
Specification reference: TS 31.114 clause 5.5.1
- TPT_GWO-APP:CR 3 GO administration BER-TLV for Administration request data structure
Specification reference: TS 31.114 clause 5.5.2

10.8.3 Test cases

10.8.3.1 Test cases for Pull response

TPT_GW-APP_PULLRSP

TPT_GW-APP_PULLRSP:TC 1 Verify that the IUT can correctly interpret the “GO Pull Response BER-TLV” by rendering a contained page. Refer to MEC_COM:TC 1 to TC 3

10.8.3.2 Test cases for Push initiation

TPT_GW-APP_PUSHINI

TPT_GW-APP_PUSHINI:TC 1 Verify that the IUT can correctly interpret the “GO Push BER-TLV” by rendering a contained pushed page. Refer to test group PTP_HAND_PUSH.

10.8.3.3 Test cases for Administration requests

TPT_GW-APP_ADMREQ

TPT_GW-APP_ADMREQ:TC 1 Verify that the IUT can correctly interpret the “GO Administration Message BER_TLV” by rendering the administration request contained. Refer to test group TPT_HAND_ADM

11 Test specification 31.114 : Administration procedures

All conformance requirements and test cases described in this clause are applicable in the **Administration** conformance context only.

11.1 General administration procedures

ADM_MEC

FFS

11.2 Administration data structures

11.2.1 ADM Request

ADM_DS_REQ

11.2.1.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for administration requests.

Specification reference: 31.114 clause 6.1 and 6.1.1

11.2.1.2 Conformance requirements

ADM_DS_REQ:CR 1 For the case that administration procedures can be handled by the UI a response to an ADM request is mandatory.

ADM_DS_REQ:CR 2 Refresh of menu structure

ADM_DS_REQ:CR 3 Refresh of event handling

ADM_DS_REQ:CR 4 If an error occurs during the execution of an ADM command, no further ADM commands of that ADM request are executed. Specification reference TS 31.114 clause 6.2.

11.2.1.3 Test cases

ADM_DS_REQ:TC 1 Render an administration request, verify that an administration response is generated by the IUT.

ADM_DS_REQ:TC 2 Render an administration request with Refresh of menu structure requested. Verify that the menu structure gets refreshed Refer to ADM_CMD_MTP:TC 1

ADM_DS_REQ:TC 3 No refresh. Run as test case ADM_DS_REQ:TC 2 but Refresh of menu structure is not requested. Verify that the menu structure remains unchanged.

ADM_DS_REQ:TC 4 Render an administration request with an Install event adm-cmd and request Refresh of event handling. Verify that the installed event is added to the events monitored by the IUT. Refer to ADM_CMD_INSE TC 1

ADM_DS_REQ:TC 5 No refresh of event handling . Render an administration request with an Install event adm-cmd and do not request Refresh of event handling. Verify that the installed event has not been added to the events monitored by the IUT . Refer to ADM_CMD_INSE:TC 2

ADM_DS_REQ:TC 6 Let an error occur when an adm request with several adm commands is sent to the IUT. Verify that the subsequent adm commands are not executed.

11.2.2 ADM response

ADM_DS_RESP

11.2.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure for administration response.

Specification reference: 31.114 clause 6.2 and 6.2.1

11.2.2.2 Conformance requirements

ADM_DS_RESP:CR 1 USAT Interpreter shall generate ADM response and include it in UIO administration BER-TLV for Administration Message data structure

ADM_DS_RESP:CR 2 "If an error occurs during the execution of an ADM command, no further ADM commands of that ADM request are executed. Nevertheless the ADM response shall be sent"

ADM_DS_RESP:CR 3 ADM Response contains a list of results, as in order given in request

11.2.2.3 Test cases

ADM_DS_RESP:TC 1 Render an Administration request. Verify that the UIO BER-TLV for Administration Message data structure is generated with the ADM Response TLV included. One ADM-CMD executed ok, verify response is sent. Refer to ADM_DS_REQ:TC 1 and TPT_UI-APP_ADMRSP:TC 1

ADM_DS_RESP:TC 2 Run test case ADM_DS_REQ:TC 6. (Error) Verify that the adm response is still sent.

11.2.3 ADM result

Specification reference: 31.114 clause 6.2.2 and 6.3

ADM_DS_RESU_TC 1 : One ADM-CMD executed ok, verify result in response

ADM_DS_RESU_TC x : Execute an ADM-CMD, invoke all of the errors listed, verify result

Combine "order" with different error codes...

11.3 Administration commands

ADM_CMD

Specification reference: 31.114 clause 6.1.2

11.3.1 Install Page

ADM_CMD_INSP

11.3.1.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Install page command. Specification reference: 31.114 clause 6.1.2.1

Verification of the result of install page command is applicable within

- Resident [CR 1, CR 2, CR 3] or
- Remote [CR 3]

conformance context.

11.3.1.2 Conformance requirements

ADM_CMD_INSP:CR 1	Install page command stores a page locally.
ADM_CMD_INSP:CR 2	If a page with the same page identification is already stored locally, the Install Page command shall overwrite that page.
ADM_CMD_INSP:CR 3	Access to the installed page, (resident only or not) can be defined with an attribute. Access 0: the page may be accessed by internal resources only (e.g. from the menu structure, from events, from other locally stored pages...) 1: the page may be accessed by any other page and any internal resource
ADM_CMD_INSP:CR 4	The TAR value provided by the USAT Interpreter system shall be available as a configuration in the Operational Pull Configuration Set. If no matching configuration is available, the USAT Interpreter shall issue the error "Configuration not available".

11.3.1.3 Test cases

ADM_CMD_INSP:TC 1	Install page for internal resources only, verify that the page is accessible from another resident page .
ADM_CMD_INSP:TC 2	Install page for access to any other page and any internal resource. Verify in remote conformance context. (Render a remote page that branches to the installed page, verify that the installed page is rendered)
ADM_CMD_INSP:TC 3	Install page in order to overwrite an old one. Verify by rendering the page.

- ADM_CMD_INSP:TC 4 Install page with a TAR representing a configuration that is not available, verify that the correct error is issued. ("Configuration not available")
- ADM_CMD_INSP:TC 5 Let the page pool be nearly full. Install page, verify that the correct error is issued by the USAT Interpreter. The error code shall indicate the remaining memory.

Note: Access between configurations ...FFS

11.3.2 Remove Page

ADM_CMD_RMP

Specification reference: 31.114 clause 6.1.2.2

11.3.2.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Remove page command. Specification reference: 31.114 clause 6.1.2.2

Verification of the result of install page command is applicable within Resident conformance context.

11.3.2.2 Conformance requirements

- ADM_CMD_RMP:CR 1 It shall be possible to remove a resident page, only the page.
- ADM_CMD_RMP:CR 2 It shall be possible to remove a resident page and all registered events and menu entries pointing to that page are to be removed by the USAT Interpreter.

11.3.2.3 Test cases

- ADM_CMD_RMP:TC 1 Remove the page, verify that the page can not be rendered (error message?) by branching to it from another resident page.
- ADM_CMD_RMP:TC 2 Remove the page, verify that the page can not be rendered (error message?) by menu selection. What is the specified behaviour when the menu item is selected?
- Add to list of issues to clarify!
- ADM_CMD_RMP:TC 3 Remove page and links to it (i.e. events and menu entries). Verify that there is no longer a menu entry pointing to the removed page.

11.3.3 Configure USAT Interpreter entering menu title

ADM_CMD_EMT

11.3.3.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Configure USAT Interpreter Entering Menu Title administration command. Specification reference: 31.114 clause 6.1.2.3

11.3.3.2 Conformance requirements

- ADM_CMD_EMT:CR 1 This command is used to specify a text to be used in the menu structure of the UE to reach the menus provided by the USAT Interpreter. Optionally a reference to an icon may be provided.

11.3.3.3 Test cases

- ADM_CMD_EMT:TC 1 Configure a text to the entering menu, verify that the refreshed menu structure has the specified text as the entering menu title.
- ADM_CMD_EMT:TC 2 Configure an icon reference to the entering menu. To where may the icon ref point (EFimg) locally.

11.3.4 Configure menu text for page

ADM_CMD_MTP

11.3.4.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structures specified for the Configure Menu Text for Page administration command. Specification reference: 31.114 clause 6.1.2.4

11.3.4.2 Conformance requirements

- ADM_CMD_MTP:CR 1 This command is used to specify the page to be executed, when the USAT Interpreter receives a menu selection related to the text provided in this command. The USAT Interpreter shall verify the existence of the page linked to this text
- ADM_CMD_MTP:CR 2 If there is already a menu text with the same page identification as the one received in this UI-TLV then menu text of this UI-TLV shall overwrite the existing menu text.

11.3.4.3 Test cases

- ADM_CMD_MTP:TC 1 Render an ADM Request with the administration command “Configure menu text for page” for an existing resident page, which does not have a menu entry linked to it. Let refresh of menu structure be requested in the ADM Request. Verify that the configured text is there in the menu and that the page is rendered when the menu item is selected.
- ADM_CMD_MTP:TC 2 Configure a text for a non existing page Render the administration command “Configure menu text for page” for a non-existing resident page. Verify that the menu text is not configured and error message is issued. (Reference to undefined)
- ADM_CMD_MTP:TC 3 Configure a text in order to overwrite an already configured text. Verify that the menu gets refreshed with the new text and the page rendered.

11.3.5 Remove menu text for page

ADM_CMD_RMTP

11.3.5.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structures specified for the Remove Menu Text for Page administration command. Specification reference: 31.114 clause 6.1.2.5

11.3.5.2 Conformance requirements

- ADM_CMD_MTP:CR 1 This command is used to remove the item from the menu linked to the specified page, without removing the linked page itself.

11.3.5.3 Test cases

ADM_CMD_MTP:TC 1 Render an ADM Request with the Remove Menu Text for Page administration command for an existing menu entry for an existing resident page. Let the Refresh of menu structure attribute of ADM Request be set. Verify that the menu entry pointing at the specified resident page is removed.

Also verify that the page has not been removed. (Branch to the page from another resident page, maybe a reference)

ADM_CMD_MTP:TC 2 Render an ADM Request with the Remove Menu Text for Page administration command. Let the Refresh of menu structure attribute of ADM Request be set. Let the menu text to be removed refer to a non-existing resident page. Verify that the menu is not changed and that an error is issued.(Reference to undefined)?

Add an issue to the list of issues about the error code.

ADM_CMD_MTP:TC 3 Render an ADM Request with the Remove Menu Text for Page administration command for an non-existing menu entry for an existing resident page. Let the Refresh of menu structure attribute of ADM Request be set. Verify that the correct error code is issued (No error + reference to undefined)

11.3.6 Install event

ADM_CMD_INSE

11.3.6.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Install event administration command. Specification reference: 31.114 clause 6.1.2.6

11.3.6.2 Conformance requirements

ADM_CMD_INSE:CR 1 “This command specifies the page to be executed, when specific events are detected by the UE.”

ADM_CMD_INSE:CR 2 “If there is already an event set up with the same Event Identifier as the one received in this UI-TLV then the Page Identification of this UI-TLV shall overwrite the existing Page Identification.”

ADM_CMD_INSE:CR 3 In addition to the event identifiers specified in 3GPP TS 31.111 , the USAT Interpreter shall support the following USAT Interpreter specific internal events:

Event	Event Identifier Coding	Description
USAT Interpreter initialisation	'A0'	Issued once per initialisation of the USAT Interpreter

11.3.6.3 Test cases

ADM_CMD_INSE:TC 1 Install an event, with refresh, verify that the event now triggers the rendering of the configured page. What event?

ADM_CMD_INSE:TC 2 Install an event, without refresh (shall not take effect).

ADM_CMD_INSE:TC 3 install to overwrite with new page for an existing . Render the adm-cmd Install Page where a new page identification is configured for an event identifier. Let the event occur. Verify that the new page is rendered.

ADM_CMD_INSE:TC 4 Install the event “USAT Interpreter initialisation”?

11.3.7 Remove event

ADM_CMD_RME

11.3.7.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Remove event administration command.

Specification reference: 31.114 clause 6.1.2.7

11.3.7.2 Conformance requirements

ADM_CMD_RME:CR 1 “This command removes an event from the current event list”

11.3.7.3 Test cases

ADM_CMD_RME:TC 1 Render the adm-cmd Remove Event that removes an event from the list of monitored events. Let that event occur. Verify that the IUT no longer monitors the event. What event?

ADM_CMD_RME:TC 2 Render the adm-cmd Remove Event that request the IUT to remove a non-existing event from the list of monitored events. Verify that the error code is issued (No error + Reference to undefined). What event?

11.3.8 Modify environment variables

ADM_CMD_ENV

11.3.8.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified Modify Environment Variable administration command.

Specification reference: 31.114 clause 6.1.2.8

11.3.8.2 Conformance requirements

ADM_CMD_ENV:CR 1 This command is used to modify the variables of the USIM issuer information partition.

ADM_CMD_ENV:CR 2 If variable to modify does not exist, it shall be created by USAT Interpreter

ADM_CMD_ENV:CR 3 If a variable to be set by this command does exist already, the content provided by this command shall overwrite the previous content of the variable.

ADM_CMD_ENV:CR 4 If the In line Value UI-TLV is not available in the Modify Environment Variable administration command, the variable referenced by the Variable ID shall be removed by the USAT Interpreter.

ADM_CMD_ENV:CR 5 If the referenced variable to be removed does not exist, no error shall be generated by the USAT Interpreter.

11.3.8.3 Test cases

ADM_CMD_ENV:TC 1 Render an adm-cmd Modify Environment Variable to create the variable <Variable ID>. Verify that the variable is created and the content of the variable is stored.

ADM_CMD_ENV:TC 2	Render an adm-cmd Modify Environment Variable to modify the variable <Variable ID>. Verify that the content of the variable is modified.
ADM_CMD_ENV:TC 3	Render an adm-cmd Modify Environment Variable to modify the variable <Variable ID> Let the Inline Value be not present. Verify that the variable is removed.
ADM_CMD_ENV:TC 4	Render an adm-cmd Modify Environment Variable to modify the variable <Variable ID> .Let the Inline Value be not present and the variable not exist. Verify that no error is issued.
ADM_CMD_ENV:TC 5	Render an adm-cmd Modify Environment Variable to create the variable <Variable ID>. Let the Variable ID be out of range. Verify that the correct error is issued. (Reference to undefined)

11.3.9 Modify wait state message

ADM_CMD_WSM

11.3.9.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified Modify Wait State Message administration command.

Specification reference: 31.114 clause 6.1.2.9

Verification of the test cases are performed within Resident conformance context.

11.3.9.2 Conformance requirements

ADM_CMD_WSM:CR 1 This command is used to modify the default message to be shown during Wait State by the USAT Interpreter.

11.3.9.3 Test cases

ADM_CMD_WSM:TC 1 Render the adm-cmd Modify Wait State Message to configure a new default text to be displayed when the UI goes into wait state. Then render a page that initiates a pull request and wait state. (Refer to MEC_COM:TC 4). Verify that the new text is displayed when the UI goes into wait state.

11.3.10 Modify configuration data

ADM_CMD_CD

11.3.10.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified for Modify Configuration Data administration command.

Specification reference: 31.114 clause 6.1.2.10

11.3.10.2 Conformance requirements

ADM_CMD_CD:CR 1 This command is used to modify the configuration data for a specific TAR value in the Push Operational or Pull Operational mode.

ADM_CMD_CD:CR 2 If the addressed operational configuration does not exist, the operational configuration shall be created and stored by the USAT Interpreter.

ADM_CMD_CD:CR 3 The following configuration data shall be possible to modify with the Modify Configuration Data adm-cmd:

- 23.048 counter UI terminated
- 23.048 counter UI originated
- 23.0348 KIc keys
- 23.048 KID keys
- SPI, KIc and KID bytes for UI originated
- SPI, KIc and KID list to check for UI terminated
- USAT command list
- UI-CMD list
- Linked pull TAR
- Bearer specific data set

11.3.10.3 Test cases

TC: Many cases, one area per parameter that is possible to set:Bad TAR value, create new configuration

ADM_CMD_CD:TC 1

11.3.11 Modify bearer data

ADM_CMD_BEARD

11.3.11.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified Modify Bearer Data administration command.

Specification reference: 31.114 clause 6.1.2.11

11.3.11.2 Conformance requirements

ADM_CMD_BEARD:CR 1 This command is used to modify the configuration data for a specific bearer.

ADM_CMD_BEARD:CR 2 If the addressed bearer specific data set does not exist, the bearer specific data set shall be created and stored by the USAT Interpreter.

Add test cases ex data from table 6.1.2.11.4

CR 2 can not be tested.

11.3.12 Install plug-in

11.3.12.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified Install Plug-In administration command. Specification reference: 31.114 clause 6.1.2.12.

11.3.12.2 Conformance requirements

- ADM_CMD_INSPI:CR 1 This command is used to extend the functionality of the USAT Interpreter by installing plug-ins
- ADM_CMD_INSPI:CR 2 If a plug-in is installed successfully by the USAT Interpreter, the NCI value ... shall be reflected in the USAT Interpreter system information variable '03'...
- ADM_CMD_INSPI:CR 3 If the NCI value given in the Install Plug-in command is already used by an installed plug-in, an error shall be generated.

11.3.12.3 Test cases

FFS

11.3.13 Remove plug-in

11.3.13.1 Definition and applicability

The tests in this sub-clause ensure that the IUT conforms to the mechanisms and data structure specified Remove Plug-In administration command.

Specification reference: 31.114 clause 6.1.2.13

11.3.13.2 Conformance requirements

- ADM_CMD_RMPI:CR 1 This command is used to remove an existing Plug-In.

11.3.13.3 Test cases

FFS

Annex A (Normative): Test Language Definition

A.1 Overview

An introduction to *ui-test-ml* and its elements is given. This sub-clause is informative.

A.1.1 Root element

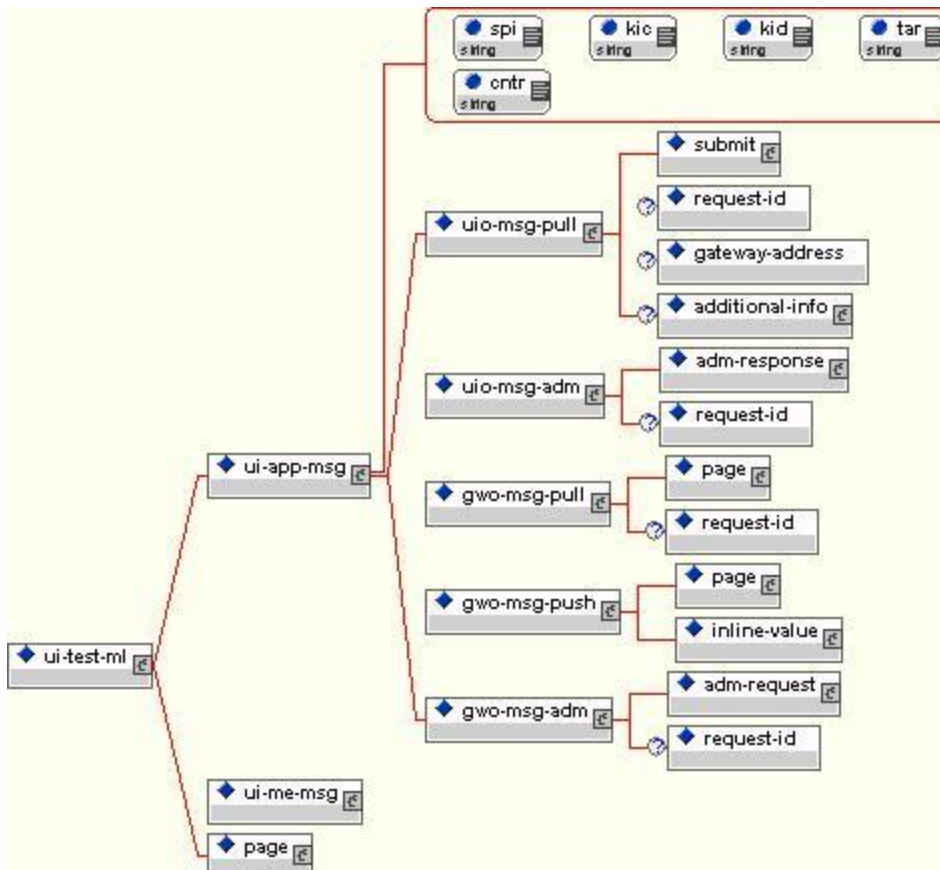
The root element of the UI test language (*ui-test-ml*) is a container for the three categories of test documents:

- Application messages going between the IUT and the gateway simulator (*ui-app-msg*)
- Messages going between the IUT and the mobile equipment simulator (*ui-me-msg*)
- Resident pages installed on the IUT (*page*)

The interface between the IUT and the gateway system (represented by *ui-app-msg*) is specified in the USAT Interpreter core specification, i.e. the TS 31.113 and TS 31.114.

The interface between the IUT and the ME (represented by *ui-me-msg*) is specified in the [USAT spec] and [ME-USIM spec].

The purpose of this test specification is to test the interface between the IUT and the gateway system, the interface between IUT and ME is only used for some verifications.



A.1.2 ui-app-msg element

The *ui-app-msg* element is the container of the five categories of messages going between the IUT and the gateway simulator system. This element holds the test language markup of the transport level (transmission protocol). The 03.48 security information parameters required to encode the messages are provided on attribute level of this element.

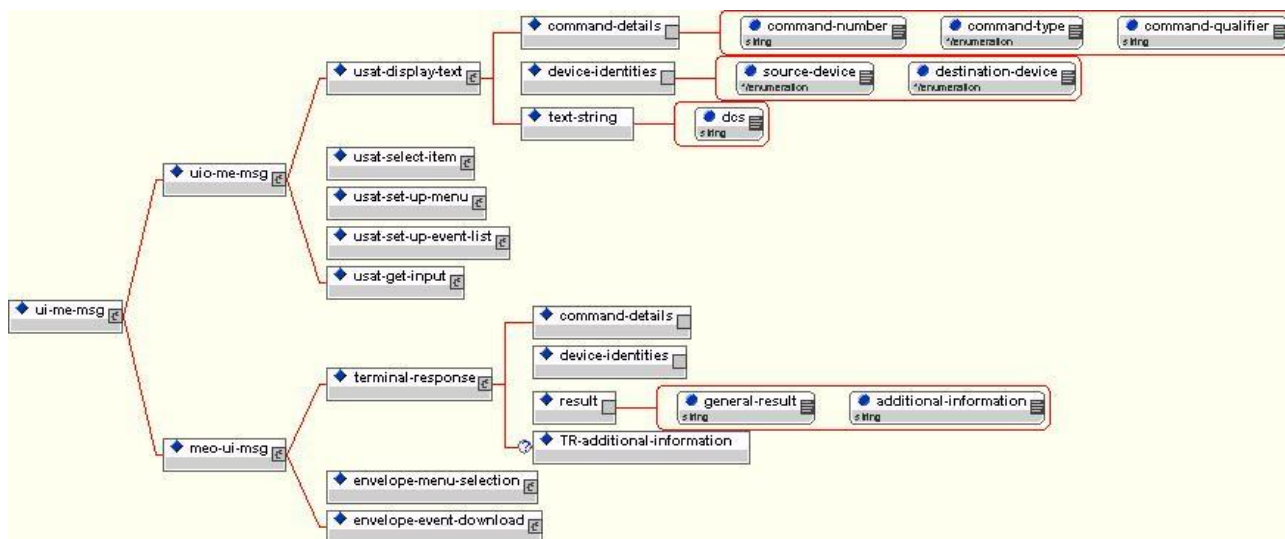
Core specification reference: TS 31.114 clause 5

A.1.3 ui-me-msg element

The *ui-me-msg* element is the container of two categories of messages going between the IUT and the ME. This interface is composed of two sub interfaces, which corresponds to the two data flow directions:

- *uio-me-msg* interface defines some USAT commands sent by IUT to ME: display text, select item, set up menu and set up event list (see TS31.111);
- *meo-ui-msg* interface defines some phase 2+ APDU and commands sent by ME to IUT: envelope menu selection, terminal response and envelope event download (see TS31.111).

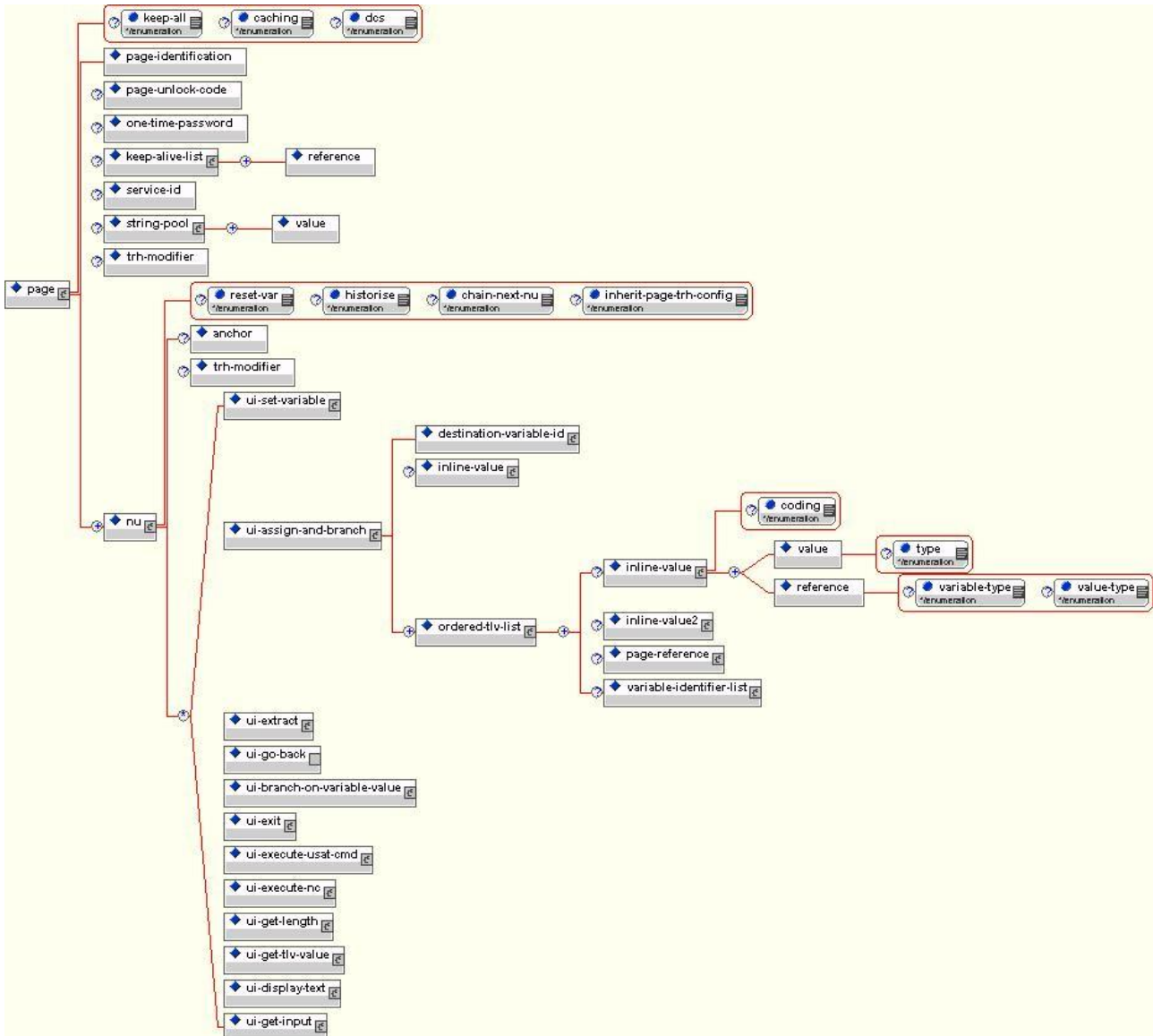
In each sub-interface, only the subset of commands that are used by the IUT are described, because only a few commands are used by the IUT.



A.1.4 The page elements

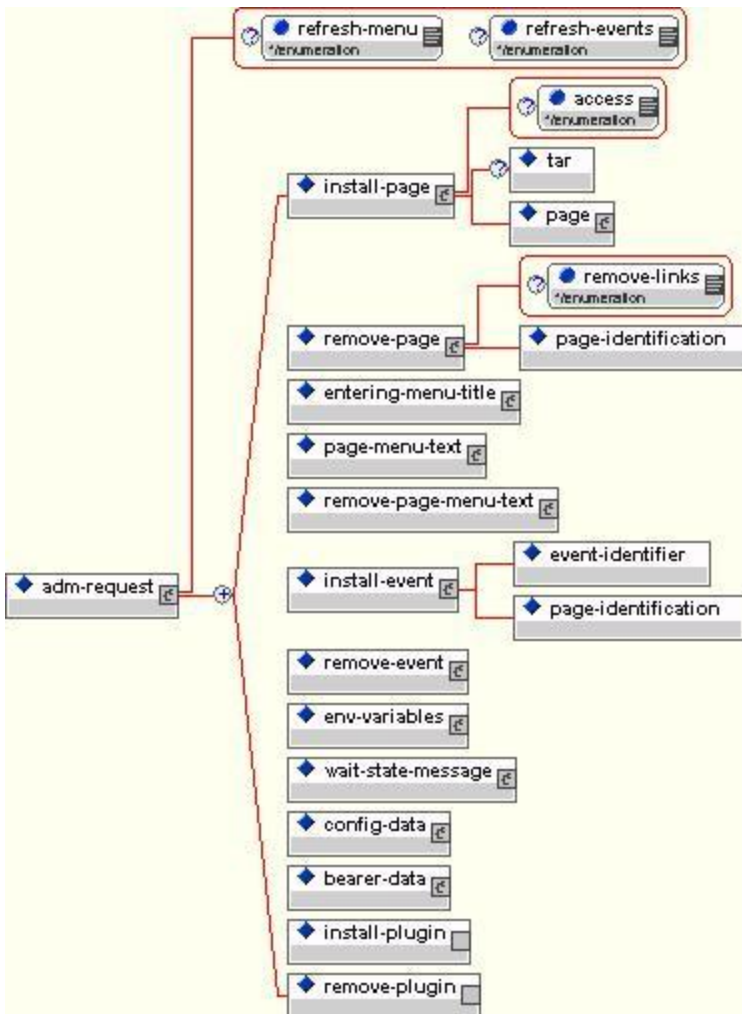
The *page* element represents the test language markup of the whole scope of the “byte code spec” TS 31.113.

The *page* element is the top level element of the TLVs specified in TS31.113. The structure and element naming of *page*, and its sub-elements, strictly follows the core specification. At the navigation unit level the *nu* element is a subsequent top level element for all of the UI commands (“byte codes”) as specified in TS 31.113 clause 8. Best illustrated with a figure:



A.1.5 The adm-request element

The *adm-request* element is the top-level element for all the administration commands specified in TS 31.114 clause 6.



A.2 Document Type Definition

```
<?xml version='1.0' encoding='UTF-8' ?>

<!--===== -->

<!--          ui-test-ml Document Type Definition          -->

<!--          Markup language for UASAT Interpreter test          -->

<!--          related to TS31.123 v first version          -->

<!--          Core spec versions 31.13 6.0.0, 31.114 5.0.0          -->

<!--===== -->

<!ENTITY % action-to-be-performed "page-reference | anchor-reference | ui-execute-nc | ui-display-
text | ui-set-variable | ui-get-input | ui-execute-usat-cmd">

<!ENTITY % ui-cmd "ui-set-variable | ui-assign-and-branch | ui-extract | ui-go-back | ui-branch-on-
variable-value | ui-exit | ui-execute-usat-cmd | ui-execute-nc | ui-get-length | ui-get-tlv-value |
ui-display-text | ui-get-input">

<!--===== -->

<!ELEMENT ui-test-ml (ui-app-msg | ui-me-msg | page)>

<!--===== -->

<!ELEMENT ui-app-msg (uio-msg-pull | uio-msg-adm | gwo-msg-pull | gwo-msg-push | gwo-msg-adm)>

<!ATTLIST ui-app-msg

        spi CDATA #REQUIRED

        kic CDATA #REQUIRED

        kid CDATA #REQUIRED

        tar CDATA #REQUIRED

        cntr CDATA #REQUIRED >

<!ELEMENT ui-me-msg (uio-me-msg | meo-ui-msg)>

<!--===== -->

<!ELEMENT uio-msg-pull (submit , request-id? , gateway-address? , additional-info?)>

<!ELEMENT uio-msg-adm (adm-response , request-id?)>

<!ELEMENT gwo-msg-pull (page , request-id?)>

<!ELEMENT gwo-msg-push (page , inline-value)>

<!ELEMENT gwo-msg-adm (adm-request , request-id?)>

<!--===== -->

<!ELEMENT uio-me-msg (usat-display-text | usat-select-item | usat-set-up-menu | usat-set-up-event-
list | usat-get-input)>

<!ELEMENT meo-ui-msg (terminal-response | envelope-menu-selection | envelope-event-download)>

<!--===== -->

<!ELEMENT submit (submit-data , page-identification?)>

<!ELEMENT request-id (#PCDATA)>

<!ELEMENT additional-info (ui-version, ui-profile, hash-url, reception-buffer-size, transmission-
buffer-size?)>

<!ELEMENT ui-version (#PCDATA)>
```

```
<!ELEMENT ui-profile (#PCDATA)>
<!ELEMENT hash-url (#PCDATA)>
<!ELEMENT reception-buffer-size (#PCDATA)>
<!ELEMENT transmission-buffer-size (#PCDATA)>
<!--=====-->
<!ELEMENT page (page-identification , page-unlock-code? , one-time-password? , keep-alive-list? ,
service-id? , string-pool? , trh-modifier? , nu+)>
<!ATTLIST page
    keep-all (yes | no ) 'no'
    dynamic (yes | no ) 'no'
    dcs (sms | ucs2 ) 'sms' >
<!--=====-->
<!ELEMENT page-unlock-code (#PCDATA)>

<change note: attributes removed in 31.113 v 6.0.0>
<!ELEMENT one-time-password (#PCDATA)>
<!ELEMENT keep-alive-list (reference+)>
<!ELEMENT service-id (#PCDATA)>
<!ELEMENT string-pool (value)+>
<!ELEMENT trh-modifier (inline-value? , action*)>
<!ATTLIST trh-modifier
    type (replace | add | restore | remove ) 'replace'
    general-result-low CDATA #REQUIRED
    general-result-high CDATA #REQUIRED >
<!ELEMENT action (inline-value? , (%action-to-be-performed;)?>
<!ATTLIST action
    id CDATA #REQUIRED
    execution (next | current ) 'next' >
<!--=====-->
<!ELEMENT nu (anchor? , trh-modifier? , (%ui-cmd;)*>
<!ATTLIST nu
    reset-var (yes | no ) 'no'
    do-not-historize (yes | no ) 'no'
    chain-next-nu (yes | no ) 'no'
    do-not-inherit-page-trh (yes | no ) 'no' >
<!--=====-->
<!ELEMENT anchor (#PCDATA)>
<!--=====-->
<!-- UI commands -->
```

```
<!--=====-->
<!ELEMENT ui-set-variable ((destination-variable-id , (inline-value | variable-identifier-list))+)>
<!ELEMENT ui-assign-and-branch (destination-variable-id , inline-value? , ordered-tlv-list+)>
<!ELEMENT ui-extract (destination-variable-id , source-variable-id)>
<!ATTLIST ui-extract
    start CDATA #REQUIRED
    len CDATA #REQUIRED >
<!--=====-->
<!ELEMENT ui-go-back EMPTY>
<!ATTLIST ui-go-back
    restart-current-nu (yes | no ) 'yes' >
<!ELEMENT ui-branch-on-variable-value (source-variable-id , ordered-tlv-list+ , page-reference?)>
<!ELEMENT ui-exit (variable-identifier-list?)>
<!ATTLIST ui-exit
    terminate-session (yes | no ) 'no' >
<!ELEMENT ui-execute-usat-cmd (tr-result-variable-id? , destination-variable-id? , usat-cmd)>
<!ATTLIST ui-execute-usat-cmd
    cmd-type CDATA #REQUIRED
    cmd-qual CDATA #REQUIRED
    dest-device CDATA #REQUIRED
    optimization (yes | no ) 'yes'
    behaviour (continue-any-GR | stop-GRvar-not-present | stop-GRvar-not-present |
    invokeTRH-any-GR) 'stop-GRvar-not-present'>
<!ELEMENT tr-result-variable-id (reference)>
<!ELEMENT usat-cmd ((simple-tlv | simple-tlv-indicator)+)>
<!ELEMENT simple-tlv (#PCDATA)>
<!ELEMENT simple-tlv-indicator (value | reference)+>
<!ATTLIST simple-tlv-indicator
    id CDATA #REQUIRED >
<!ELEMENT ui-execute-nc (nci , input-list? , variable-identifier-list?)>
<!ATTLIST ui-execute-nc
    exit (yes | no ) 'no' >
<!--=====-->
<!ELEMENT nci (#PCDATA)>
<!ELEMENT input-list (variable-identifier-list | inline-value)+>
<!--=====-->
<!ELEMENT ui-get-length (source-variable-id , variable-identifier-list)>
<!ELEMENT ui-get-tlv-value (destination-variable-id, variable-identifier-list)>
<!ATTLIST ui-get-tlv-value
```

```
tag CDATA #REQUIRED>

<!ELEMENT ui-display-text (inline-value)>
<!ATTLIST ui-display-text clear (user | delay ) 'user' >
<!ELEMENT ui-get-input (destination-variable-id , inline-value , inline-value2?)>
<!ATTLIST ui-get-input
    charset (sms | ucs2 ) 'sms'
    format (digit | alpha ) 'digit'
    minlength CDATA '0' >
<!ELEMENT inline-value2 (value | reference)+>
<!ATTLIST inline-value2
    coding (unknown | sms | sms-packed | binary | ucs2 ) 'binary' >
<!--=====-->
<!ELEMENT usat-display-text (command-details, device-identities, text-string)>
<!ELEMENT usat-select-item (command-details, device-identities, alpha-identifier?, item-data-object+)>
<!ELEMENT usat-set-up-menu (command-details, device-identities, alpha-identifier, item-data-object+, icon-descriptor?)>
<!ELEMENT usat-set-up-event-list (command-details, device-identities, event-list)>
<!ELEMENT usat-get-input (command-details, device-identities, text-string, response-length)>
<!--=====-->
<!ELEMENT terminal-response (command-details, device-identities, result, TR-additional-information?)>
<!ELEMENT envelope-menu-selection (device-identities, item-identifier, help-request?)>
<!ELEMENT envelope-event-download (event-list, device-identities, event-download-additional-information?)>
<!--=====-->
<!-- Administration -->
<!--=====-->
<!ENTITY % adm-cmd "install-page | remove-page | entering-menu-title | page-menu-text | remove-page-menu-text | install-event | remove-event | env-variables | wait-state-message | config-data | bearer-data | install-plugin | remove-plugin">
<!ELEMENT adm-request (%adm-cmd;)+>
<!ATTLIST adm-request
    refresh-menu (yes | no ) 'no'
    refresh-events (yes | no ) 'no' >
<!--=====-->
<!-- ui-adm-commands -->
<!--=====-->
<!ELEMENT install-page (tar? , page)>
<!ATTLIST install-page
```

```
        access (resident | open ) 'resident' >
<!ELEMENT remove-page (page-identification)>
<!ATTLIST remove-page
        remove-links (yes | no ) 'no' >
<!ELEMENT entering-menu-title (inline-value , icon-identifier?)>
<!ELEMENT icon-identifier EMPTY>
<!ATTLIST icon-identifier
        qualifier CDATA #REQUIRED
        identifier CDATA #REQUIRED >
<!ELEMENT page-menu-text (inline-value , page-identification)>
<!ELEMENT remove-page-menu-text (page-identification)>
<!ELEMENT install-event (event-identifier , page-identification)>
<!ELEMENT remove-event (event-identifier)>
<!ELEMENT env-variables (destination-variable-id , inline-value?)>
<!ELEMENT wait-state-message (inline-value)>
<!ELEMENT config-data (parameter-list)>
<!ATTLIST config-data tar CDATA #REQUIRED>
<!ELEMENT bearer-data (data-set-index , bearer-type , parameter-list)>
<!ELEMENT data-set-index (#PCDATA)>
<!ELEMENT bearer-type (#PCDATA)>
<!ELEMENT install-plugin (nci, plugin-installation-data)>
<!ELEMENT remove-plugin (nci)>
<!ELEMENT plugin-installation-data(#PCDATA) >
<!--=====-->
<!-- Basic adm-UI-TLV elements used by several adm-cmd elements -->
<!--=====-->
<!ELEMENT parameter-list (parameter+)>
<!ELEMENT tar (#PCDATA)>
<!ELEMENT event-identifier (#PCDATA)>
<!--=====-->
<!--=====-->
<!ELEMENT adm-response (adm-result+)>
<!ELEMENT adm-result (result-info?)>
<!ATTLIST adm-result
        result-code (no-error | memory-mgmt | security | ref-undef | cmd-not-allowed |
                    config-change-not-allowed | no-memory | cmd-unknown | config-unavailable |
                    bad-tar | unknown-error ) #REQUIRED >
<!--=====-->
```

```
<!-- Basic UI-TLV Elements used by several elements -->
<!--=====
<!ELEMENT page-identification (#PCDATA)>
<!ELEMENT inline-value (value | reference)+>
<!ATTLIST inline-value
      coding (unknown | sms | sms-packed | binary | ucs2 ) 'unknown' >
<!ELEMENT value (#PCDATA)>
<!ATTLIST value
      type (cleartext | hexa ) 'cleartext' >
<!ELEMENT reference (#PCDATA)>
<!ATTLIST reference
      variable-type (env | perm | temp | page-string ) #REQUIRED
      value-type (unknown | sms | sms-packed | binary | ucs2 ) 'unknown' >
<!ELEMENT variable-identifier-list (reference)+>
<!ELEMENT destination-variable-id (reference)>
<!ELEMENT ordered-tlv-list (inline-value? , inline-value2? , page-reference? ,
      variable-identifier-list?)+>
<!ELEMENT source-variable-id (reference)>
<!ELEMENT page-reference (anchor-reference | variable-identifier-list | submit-configuration)>
<!ELEMENT gateway-address (#PCDATA)>
<!ATTLIST gateway-address
      send-additional-info (yes | no ) 'no' >
<!ELEMENT submit-data ((value | reference)*)>
<!ELEMENT parameter (#PCDATA)>
<!ATTLIST parameter
      index CDATA #REQUIRED>
<!--=====
<!-- - Basic UI-TLV Elements used by page-reference element -->
<!--=====
<!ELEMENT anchor-reference (#PCDATA)>
<!ELEMENT submit-configuration (submit-data , inline-value? , gateway-address?)>
<!ATTLIST submit-configuration
      send-ref (no | yes ) 'no'
      handling (waitstate | next-uicmd ) 'waitstate' >
<!ELEMENT result-info (#PCDATA)>
<!--=====
<!ELEMENT command-details EMPTY>
<!ATTLIST command-details
```

```
command-number CDATA #REQUIRED

command-type (display-text | select-item | set-up-menu | set-up-event-list | get-input)
#REQUIRED

command-qualifier CDATA #REQUIRED >

<!ELEMENT device-identities EMPTY>

<!ATTLIST device-identities

    source-device (Keypad | Display | Earpiece | AdditionalCardReader0 |
AdditionalCardReader1 | AdditionalCardReader2 | AdditionalCardReader3 | AdditionalCardReader4 |
AdditionalCardReader5 | AdditionalCardReader6 | AdditionalCardReader7 | Channel1 | Channel2 |
Channel3 | Channel4 | Channel5 | Channel6 | Channel7 | UICC | ME | Network) #REQUIRED

    destination-device (Keypad | Display | Earpiece | AdditionalCardReader0 | AdditionalCardReader1
| AdditionalCardReader2 | AdditionalCardReader3 | AdditionalCardReader4 | AdditionalCardReader5 |
AdditionalCardReader6 | AdditionalCardReader7 | Channel1 | Channel2 | Channel3 | Channel4 | Channel5
| Channel6 | Channel7 | UICC | ME | Network) #REQUIRED >

<!ELEMENT text-string (#PCDATA)>

<!ATTLIST text-string

    DCS (sms | sms-packed | ucs2 ) #REQUIRED>

<!ELEMENT alpha-identifier (#PCDATA)>

<!ELEMENT item-data-object (#PCDATA)>

<!ATTLIST item-data-object

    item-identifier CDATA #REQUIRED>

<!ELEMENT icon-descriptor EMPTY>

<!ATTLIST icon-descriptor

    icon-qualifier (self-explanatory | not-self-explanatory) #REQUIRED

    icon-identifier CDATA #REQUIRED >

<!ELEMENT event-list (event-type+)>

<!ELEMENT event-type EMPTY>

<!ATTLIST event-type

    event-type-id (MT-call | call-connected | call-disconnected | location-status | user-activity |
idle-screen-available | card-reader-status | language-selection | browser-termination | data-
available | channel-status) #REQUIRED>

<!ELEMENT result EMPTY>

<!ATTLIST result

    general-result CDATA #REQUIRED

    additional-information CDATA #REQUIRED>

<!ELEMENT TR-additional-information (#PCDATA)>

<!ELEMENT item-identifier (#PCDATA)>

<!ELEMENT help-request EMPTY>

<!ELEMENT event-download-additional-information (#PCDATA)>
```


<!ELEMENT response-length (#PCDATA)>

A.3 Element descriptions

This sub-clause gives specific instructions on how to enter test data into a *ui-test-ml* document.

ELEMENT item-identifier

One byte between 00 och FF representing the menu entry item, to be managed by the UICC

ELEMENT reference

Variable ids: 1 or 2 hexadecimal characters

ELEMENT page-identification

Inserted by the gateway, must not contain “#” character

ELEMENT page-unlock-code

hexa string

ELEMENT one-time-password

Hexa string, max len 8 bytes value (16 chars)

ELEMENT service-id Hexa string, max len 8 bytes value (16 chars)

ELEMENT trh-modifier

action ignored if type=restore

general-result-low content:hexa value (2chars)

general-result-high content:hexa value (2 chars) optional:assigned by gateway to general-result-low if omitted

ELEMENT action

inline-value present only if trh-modifier attr type=add|replace

action-to-be-performed present only if id non system and trh-modifier attr type=add|replace

id content : Hexa value (2 chars) : 00-1F for system, 20-FF for service

ELEMENT anchor

Must not contain a “#”

ELEMENT reference

For string-ids:

content : decimal number to refer to a string from the string-pool (1-based index)

attr coding used for substitution

For environment variable ids:

content: a literal that indicates which environment variable has to be accessed

For permanent variable ids:

content: a decimal number to indicate the permanent variable, service specific management

ELEMENT start-index

default value 0

ELEMENT number-of-bytes

default value 0

ELEMENT usat-cmd

Attribute id : hexa value to assign to the final USAT command TLV (display text...)

serialised as a simple-tlv or tlv indicator

ELEMENT ui-get-tlv-value

Attribute tag: hexa value

ELEMENT ui-get-input

Attribute min: decimal value, default 0

ELEMENT request-id

Content: single byte in hexadecimal format

ELEMENT anchor reference

Content: must not finish by “#”

ELEMENT ui-app-msg

ELEMENT gwo-msg-push

The Inline Value element in gwo-msg-push element may NOT contain variable references, only values.

ELEMENT usat-display-text

command-details / command-number

Any hexadecimal number between ‘01’ and ‘FE’. The UICC is responsible for issuing a command number.

Thus, any number between ‘01’ and ‘FE’ is correct as far as expected result is concerned.

Note: A test client need NOT verify this data field.

command-details / command-type ENUMERATION in DTD, enter display text

command-details / command-qualifier

‘81’ indicates: High priority, wait for user to clear message

‘80’ indicates: High priority, clear message after delay

‘01’ indicates: Normal priority, wait for user to clear message

‘00’ indicates: Normal priority, clear message after delay

Note:

For **UI-CMD DisplayText**, high priority shall always be expected for correct performance.

For **UI-CMD Execute USAT command** and **DISPLAY TEXT** is requested, high or normal is specified in the UI-CMD and thus both may be correct expected results.

ELEMENT terminal-response

ELEMENT command-details – enter the command type only (shall be identical as the command-details of the USAT command that this terminal response is responding to)

ELEMENT device-identities - source ME, destination UICC

ELEMENT result

ELEMENT general-result – refer to TS 31.111 clause 8.12

‘00’ for OK (Command performed successfully)

‘12’ No response from user

ELEMENT additional-info – used in some cases if general result not OK

ELEMENT TR-additional-info

The item of a selected item, in response to a SELECT ITEM

ELEMENT usat-select-item

command-details / command-qualifier:

The expected value is ‘03’.

Note: Quote from 31.113 v600 Assign and Branch: “When a SELECT ITEM command is built by the USAT Interpreter, the command qualifier to be used shall be ‘03’ “

Annex B (Normative): Test documents

B.1 Test documents and test data

Test documents are used to specify test data in detail.

The test data for a test case is specified by:

- High-level description – Test data specified in words, human comprehensible format
- XML test documents - Test data “marked up” in human- and USAT Interpreter byte code converter readable format (i.e. *ui-test-ml*)
- Byte code test data – Test data in USAT Interpreter readable format

In this annex, XML test data and Byte code test data are specified.

A test document is an XML document that conforms to the *ui-test-ml* DTD. Test documents may be input data to the USAT Interpreter byte code converter tool, that generates the corresponding byte code representation according to TS 31.113 and TS 31.114. Test documents may also just be XML documents that specifies test data to be used by a test client or user.

The specification of test documents is divided into sub-clauses corresponding to the USAT Interpreter test procedure operations.

B.1.1 Page

Pages may be installed in the IUT as resident pages or be part of messages received by the IUT as remote pages.

Name	Content
<p>PAGE-001</p> <p>note: this page is included as an example of what a test document looks like</p>	<pre> <ui-test-ml> <page dcs = "sms"> <page-identification>PAGE-001</page-identification> <nu > <anchor>NU-01</anchor> <ui-assign-and-branch > <destination-variable-id > <reference variable-type = "temp">dv1</reference > </destination-variable-id > <ordered-tlv-list > <inline-value2 > <value>RED</value > </inline-value2 > <inline-value coding = "sms"> <value>RED SELECTED</value > </inline-value > </ordered-tlv-list > </ui-assign-and-branch > </nu > </page > </ui-test-ml > </pre>

```
<anchor-reference>NU-RED</anchor-reference>
</page-reference>
</ordered-tlv-list>
<ordered-tlv-list>
  <inline-value2>
    <value>BLUE</value>
  </inline-value2>
  <inline-value coding = "sms">
    <value>BLUE SELECTED</value>
  </inline-value>
</page-reference>
  <anchor-reference>NU-BLUE</anchor-reference>
</page-reference>
</ordered-tlv-list>
</ui-assign-and-branch>
</nu>
<nu >
  <anchor> NU-RED </anchor>
  <ui-display-text clear = "user">
    <inline-value coding = "sms">
      <value type = "cleartext">Text of NU-RED</value>
    </inline-value>
  <ui-display-text>
  <ui-display-text clear = "user">
    <inline-value coding = "sms">
      <reference variable-type = "temp">dv1</reference>
    </inline-value>
  </ui-display-text>
</nu>
<nu >
  <anchor> NU-BLUE </anchor>
  <ui-display-text clear = "user">
    <inline-value coding = "sms">
      <value type = "cleartext">Text of NU-BLUE</value>
    </inline-value>
  </ui-display-text>
  <ui-display-text clear = "user">
    <inline-value coding = "sms">
      <reference variable-type = "temp">dv1</reference>
    </inline-value>
  </ui-display-text>
</nu>
```

	<pre> </ui-display-text> </nu> </page> </ui-test-ml> </pre>
...	...

B.1.2 Submit data

UI → GW

Name	Content
	<pre> <ui-test-ml> <ui-app-msg spi = "" kic = "" kid = "" tar = "" cntr = ""> <uio-msg-pull> <submit> <submit-data/> </submit> ... </uio-msg-pull> </ui-app-msg> </ui-test-ml> </pre>
...	...

B.1.3 Receive pulled page

GW → UI

Name	Content
...	...

B.1.4 Receive pushed page

GW → UI

Name	Content
...	...

B.1.5 Send admin response

UI → GW

Name	Content
...	...

B.1.6 Receive admin request

GW → UI

Name	Content
...	...

B.1.7 Issue USAT command

B.1.7.1 Display Text

UI → ME

Name	Content
...	...

B.1.7.2 Select Item

Name	Content
SELECT-001.xml	<pre> <ui-test-ml> <ui-me-msg> <uio-me-msg> <usat-select-item> <command-details command-number = "" command-type = "select-item" command-qualifier = ""/> <device-identities source-device = "UICC" destination-device = "Display"/> <item-data-object item-identifier = "">RED</item-data-object> <item-data-object item-identifier = "">BLUE</item-data-object> </pre>

	<pre> </usat-select-item> </uio-me-msg> </ui-me-msg> </ui-test-ml> </pre>
...	...

B.1.8 Receive terminal response

ME → UI

Name	Content
...	...

B.1.9 Activation menu selection

ME → UI

Name	Content
MENUSELECT_xxx.xml	
...	...

B.1.10 Activation event

ME → UI

Name	Content
	<pre> <ui-test-ml> <ui-me-msg> <meo-ui-msg> <envelope-event-download>... </envelope-event-download> </meo-ui-msg> </ui-me-msg> </ui-test-ml> </pre>
...	...

B.2 Security descriptions

GW → UI

The attributes of the element <ui-app-msg> in the USAT Interpreter test language describes the security parameters required in the 03.48 header of the transport protocol.

For the scope of this specifications the following “security skeletons” are used:

Name	Content
	<pre><ui-test-ml> <ui-app-msg spi = "" kic = "" kid = "" tar = "" cntr = ""> ... </ui-app-msg> </ui-test-ml></pre>
...	...

Annex C (Normative): Test procedures

C.1 Test procedure conditions

The following statements are applicable to the test procedure clause for all test purposes contained within the present document:

- Unless otherwise stated, all steps within the test procedure shall be carried out in order.
- Unless otherwise stated, all the necessary security required to use USAT Interpreter in the test procedures shall be initially enabled ?
- Where steps within a test procedure involve a ME simulator sending one or more commands to the USAT Interpreter, these commands are required to be correctly encoded and conveyed to the USAT Interpreter, unless otherwise stated in the sub-clause for the test.
- Where steps within a test procedure involve a USAT Interpreter Gateway system simulator sending one or more commands to the USAT Interpreter, these commands are required to be correctly encoded, unless otherwise stated in the sub-clause for the test.
- Unless otherwise stated, the following is default:
 - Initial conditions, refer to clause 4.x
 - FFS

C.2 Test procedure notation

To specify the schematic flow of information between the test environment entities the following notation for operations are used:

Direction	US AT Interpreter test operation	Comments
UI → ME	Issue USAT-command: usat-cmd	Example: Issue USAT-command:DISPLAY TEXT In the simulated test environment this operation expands to: UI → USIM→ME simulator → Test client
ME → UI	Receive Terminal response: result value	Example: Receive Terminal response: No response from user In the simulated test environment this operation expands to: Test client → ME simulator → USIM → UI
ME → UI	Activation: menu selection – item	Activation: menu selection – item identifier In the simulated test environment this operation expands to: Test client → ME simulator → USIM → UI
ME → UI	Activation: EVENT – item	Activation :EVENT – event-identifier In the simulated test environment this operation expands to: Test client → ME simulator → USIM → UI
UI → GW	Submit data	In the simulated test environment this operation expands to: UI→ USIM→ GW simulator → Test client Byte code: UIO BER-TLV for operational pull
UI → GW	Send admin response	In the simulated test environment this operation expands to: UI→ USIM→ GW simulator → Test client

		Byte code: UIO BER-TLV for Administration messages
GW → UI	Receive pulled page	In the simulated test environment this operation expands to: Test client → GW simulator → USIM → UI Byte code: GO Pull Response BER TLV
GW → UI	Receive pushed page	In the simulated test environment this operation expands to: Test client → GW simulator → USIM → UI Byte code: GO Push BER-TLV
GW → UI	Receive admin request	In the simulated test environment this operation expands to: Test client → GW simulator → USIM → UI Byte code: GO Administration Message BER-TLV
User → ME	Select menu item Clear message Input OK	When a live ME is used to execute a test, there might not be an XML test data document. Instead the test data is given directly in the test procedure. Example : User provide input: 1234
User	Verify display Initiate	Verify against a test document or test data given indirectly in the test procedure. Initiate test, the IUT shall start to render a page by e.g. accepting a push message or receiving a menu selection
Test client	Verify expected result Initiate	In the simulated test environment, the verification entity within the test client performs the verification of IUT generated data against a specified test document. Initiate test

The IUT is the USAT Interpreter and it interfaces, as specified by TS 31.113 and 31.114 and not the USIM-ME interface as specified by TS 31.111 (USAT spec).

To verify the full variety of USAT Interpreter functionality the point of verification must sometimes be placed in the interface specified by 31.111. To handle this situation, some transparency regarding the flow of test data has to be allowed. This specification describes a verification entity, however the implementation of such a verification entity is left out of scope. This specification specified only the data concerned with the USAT Interpreter.

The following intermediate steps are considered “transparent” considering the specified test data.

Direction	USAT Interpreter test operation	Intermediate steps	
		The interface USIM-ME is specified in TS 31.111 The interface ME-GW transport bearer	
UI → ME	Issue USAT-command : usat-cmd	UI → USIM USIM → ME	Issue USAT command : DISPLAY TEXT PROACTIVE COMMAND:DISPLAY TEXT
ME → UI	Receive Terminal response : result value	ME → USIM USIM → UI	TERMINAL RESPONSE: usat-cmd Terminal response: result value
ME → UI	Activation: EVENT - item	ME → USIM USIM → UI	ENVELOPE(EVENT DOWNLOAD-event) Activation: EVENT – event identifier
ME → UI	Activation: menu selection -item	ME → USIM USIM → UI	ENVELOPE(MENU SELECTION) Activation: menu selection – item identifier

C.3 Test Case Details

C.3.1 Test group MEC

FFS

C.3.2 Test group DS

FFS

C.3.3 Test group UI-CMD

FFS

C.3.4 Test group NC

FFS

C.3.5 Test group TPT

FFS

C.3.6 Test group ADM

FFS

Annex D (Informative) : Implementation conformance applicability

D.1 Specification status reference

This table summarises the specification status of USAT Interpreter features as specified in TS 31.113 and TS 31.114. It is included for quick reference in order to explain the need for the conformance contexts that are introduced in the test specifications.

Test groups		Status
MEC	USAT Interpreter mechanisms	MANDATORY
DS	USAT Interpreter data structures	MANDATORY
UI-CMD	USAT interpreter commands	MANDATORY
NC	Native Commands	OPTIONAL
TPT	USAT interpreter transmission protocol	OPTIONAL
ADM	USAT interpreter administration procedures	OPTIONAL

D.2 Conformance context applicability

The following table indicates what USAT Interpreter areas of functionality (conformance areas) that are applicable within the three conformance contexts of an IUT.

Test groups	Conformance areas	Conformance context		
		Resident	Remote	Administration
MEC	UI mechanisms			
MEC_NAV	Navigation	x	x	-
MEC_COM	Submission of outgoing data	-	x	x
	Wait state processing behaviour	-	x	-
MEC_TRH	Default system TRH configuration	x	x	-
	Modified system TRH configuration	x	x	-
MEC_ACT	Menu selection	x	x	-
	UE event, 31.113	x	x	-
	Queued events	x	x	-
	Incoming retrieved page	-	x	-
	Incoming pushed page	-	x	-
MEC_VAR_AREA	Environment – system info	x	x	-
	Environment – issuer info	x	x	-
	Environment – end user info	x	x	-

	Permanent	x	x	-
	Temporary – read current variable	x	x	-
	Temporary – read kept variable	x	x	-
MEC_VAR_SUBST	Method 1	x	x	-
	Method 2	?	x	-
MEC_HIST	Disabled	x	x	-
	Enabled	x	x	-
DS	UI data structures			
DS_PAGE_VAR	Keeping variables between pages	x	x	-
DS_PAGE_CACH	Caching of page	-	x	-
DS_PAGE_TRH	TRH modifier	x	x	-
DS_NU	Keeping/resetting variables	x	x	-
	Add or not to history list	x	x	-
	Chaining	x	x	-
	Inherit TRH configuration from page context	x	x	-
DS_ANC	Anchor reference	x	x	-
DS_VIT	Variable identifier list	x	x	-
DS_INL		x	x	-
	Type conversions	x	x	-
DS_INL2		x	x	-
DS_INPL	Input list	x	x	-
DS_OTL	Ordered TLV list	x	x	-
DS_PREF	Navigation - branch locally else error	x	-	-
	Branch remote – according to submit configuration	-	x	-
DS_SUB		-	x	-
UI-CMD	USAT interpreter commands			
UI-CMD_SETV	Set value mechanism	x	x	-
UI-CMD_AAB	Assign and branch mechanism	x	x	-
UI-CMD_EXTR	Extract mechanism	x	x	-
UI-CMD_BAK	Go back mechanism	x	x	-
UI-CMD_BOVV	Branch on variable value mechanism	x	x	-
UI-CMD_EXIT	Exit mechanism : exit immediately or return to TRH	x	x	-
UI-CMD_USATC	Execute USAT command mechanism including variant behaviours	x	x	-

UI-CMD_ENC	Execute native command	x	x	-
UI-CMD_GLEN	Get length mechanism	x	x	-
UI-CMD_GTLV	Get value	x	x	-
UI-CMD_DISP	DISPLAY TEXT	x	x	-
	DCS handling	x	x	
	Error handling	x	x	
UI-CMD_GINP	Request input from user	x	x	-
NC	Native Commands			
NC_P7	PKCS#7 Signature Plug-in	-	-	-
NC_FP	Fingerprint Plug-in	-	-	-
NC_AD	Asymmetric Decryption Plug-in	-	-	-
NC_DE	Triple DES Encryption Plug-in	-	-	-
NC_DD	Triple DES Decryption Plug-in	-	-	-
NC_DS	Triple DES Sign Plug-in	-	-	-
NC_DU	Triple DES Unwrap Plug-in	-	-	-
NC_CP	Change PIN Plug-in	-	-	-
NC_RP	Reset PIN Plug-in	-	-	-
TPT	USAT interpreter transmission protocol			
TPT_HAND_PULL	Comm with external system	-	x	-
TPT_HAND_PUSH	Push message acceptance	-	x	-
TPT_HAND_ADM	Handle in idle mode only	-	-	x
TPT_BEAR	SMS-PP	-	x	x
	Concatenation of SMS	-	x	x
	Other bearer than SMS-PP	-	x	x
TPT_SEC		-	x	x
TPT_ADD		x	x	x
TPT_UIO-APP_PULLREQ	UIO BER-TLV for Pull request	-	x	-
TPT_UIO-APP_ADMRSP	UIO BER-TLV for Adm response	-	-	x
TPT_GW-APP_PULLRSP	GO pull response BER-TLV	-	x	-
TPT_GW-APP_PUSHINI	GO push BER-TLV	-	x	-
TPT_GW-APP_ADMREQ	GO administration request BER-TLV	-	-	x
ADM	USAT interpreter administration procedures			
ADM_MEC	Administration procedure mechanisms	-	-	x
ADM_DS_REQ		-	-	x

ADM_DS_RESP		-	-	x
ADM_DS_RESU		-	-	x
ADM_CMD	USAT adm config - Adm command white list			
ADM_CMD_INSP	Inst all page	-	-	x
ADM_CMD_RMP	Remove page	-	-	x
ADM_CMD_EMT	Configure entering menu title	-	-	x
ADM_CMD_MTP	Configure menu text for page	-	-	x
ADM_CMD_RMTP	Remove menu text for page	-	-	x
ADM_CMD_INSE	Inst all event	-	-	x
ADM_CMD_RME	Remove event	-	-	x
ADM_CMD_ENV	Modify environment variable	-	-	x
ADM_CMD_WSM	Modify wait state message	-	-	x
ADM_CMD_CD	Modify configuration data	-	-	x
ADM_CMD_BEARD	Modify bearer data	-	-	x
ADM_CMD_INSPI	Inst all plug-in	-	-	x
ADM_CMD_RMPI	Remove plug-in	-	-	x

Annex E (Informative): UI-Converter Tool

FFS

Annex F (Informative): Change history

This document is not yet under change control.

<The table below indicates all change requests that have been incorporated into the present document since it was initially approved by 3GPP TSG-T.>

Document history		
v0.1.0	May 17 th , 2002	First draft presented to T3 meeting #23 for information
v0.2.0	May 27, 2002	History list clause included (as agreed in ad-hoc #71)
v0.3.0	Nov 10 th , 2003	Presented to T3 meeting #29 for information.