

ETSI TS 102 835 V7.0.0 (2010-04)

Technical Specification

Smart Cards; Test Specification for SCWS Application Invocation API for Java Card™; Test Environment and Annexes (Release 7)



Reference

DTS/SCP-00SCWS

Keywords

API, SCWS, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

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.

© European Telecommunications Standards Institute 2010.
All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™**, **TIPHON™**, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

LTE™ is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	7
Foreword.....	7
1 Scope	8
2 References	8
2.1 Normative references	8
2.2 Informative references.....	9
3 Definitions and abbreviations.....	9
3.1 Definitions.....	9
3.2 Abbreviations	10
4 Applicability.....	10
4.1 Applicability of the present document.....	10
4.2 Applicability of the individual test cases.....	10
4.3 Applicability to different Releases	10
4.4 Definitions.....	11
4.4.1 Format of table of the table of optional features	11
4.4.2 Format of the applicability table	11
4.4.3 Status and Notations	11
4.5 Table of optional features.....	12
4.6 Applicability table	12
5 Test environment.....	13
5.1 Test environment description	13
5.2 Tests format.....	14
5.2.1 Test area reference	14
5.2.1.1 Conformance requirements	14
5.2.1.2 Test area files	15
5.2.1.3 Test procedure.....	15
5.2.1.4 Test coverage	15
5.3 Initial conditions.....	15
5.4 Package name.....	15
5.5 AID coding.....	16
5.6 Test equipment	17
5.6.1 Test tool	17
5.6.2 Interfaces and classes use	17
5.6.3 Util package.....	17
5.6.4 Java Software Development Kit	17
6 Test Cases.....	17
6.1 class uicc.scws.....	17
6.1.1 Class ScwsExtensionRegistry	17
6.1.1.1 Method register	17
6.1.1.1.1 Conformance Requirement.....	18
6.1.1.1.2 Test area files.....	18
6.1.1.1.3 Test coverage.....	18
6.1.1.1.4 Test procedure	19
6.1.1.2 Method deregister	19
6.1.1.2.1 Conformance Requirement.....	20
6.1.1.2.2 Test area files.....	20
6.1.1.2.3 Test coverage.....	20
6.1.1.2.4 Test procedure	20
6.1.2 Interface ScwsExtension.....	21
6.1.3 Class ScwsExtensionService	21
6.1.3.1 Method doDelete.....	21
6.1.3.1.1 Conformance Requirement.....	21
6.1.3.1.2 Test area files.....	21

- 6.1.3.1.3 Test coverage.....21
- 6.1.3.1.4 Test procedure22
- 6.1.3.2 Method doGet22
 - 6.1.3.2.1 Conformance Requirement.....22
 - 6.1.3.2.2 Test area files.....22
 - 6.1.3.2.3 Test coverage.....22
 - 6.1.3.2.4 Test procedure23
- 6.1.3.3 Method doHead.....23
 - 6.1.3.3.1 Conformance Requirement.....23
 - 6.1.3.3.2 Test area files.....23
 - 6.1.3.3.3 Test coverage.....23
 - 6.1.3.3.4 Test procedure24
- 6.1.3.4 Method doOptions.....24
 - 6.1.3.4.1 Conformance Requirement24
 - 6.1.3.4.2 Test area files.....24
 - 6.1.3.4.3 Test coverage.....24
 - 6.1.3.4.4 Test procedure25
- 6.1.3.5 Method doPost25
 - 6.1.3.5.1 Conformance Requirement.....25
 - 6.1.3.5.2 Test area files.....25
 - 6.1.3.5.3 Test coverage.....25
 - 6.1.3.5.4 Test procedure26
- 6.1.3.6 Method doPut.....26
 - 6.1.3.6.1 Conformance Requirement.....26
 - 6.1.3.6.2 Test area files.....26
 - 6.1.3.6.3 Test coverage.....26
 - 6.1.3.6.4 Test procedure27
- 6.1.3.7 Method doTrace27
 - 6.1.3.7.1 Conformance Requirement.....27
 - 6.1.3.7.2 Test area files.....27
 - 6.1.3.7.3 Test coverage.....27
 - 6.1.3.7.4 Test procedure28
- 6.1.4 Interface HttpRequest28
 - 6.1.4.1 Method findAndCopyKeywordValue28
 - 6.1.4.1.1 Conformance Requirement.....28
 - 6.1.4.1.2 Test area files.....28
 - 6.1.4.1.3 Test coverage.....29
 - 6.1.4.1.4 Test procedure29
 - 6.1.4.2 Method findAndCopyKeywordValue29
 - 6.1.4.2.1 Conformance Requirement.....30
 - 6.1.4.2.2 Test area files.....30
 - 6.1.4.2.3 Test coverage.....30
 - 6.1.4.2.4 Test procedure31
 - 6.1.4.3 Method getContentLength32
 - 6.1.4.4 Method getContentType.....32
 - 6.1.4.5 Method getRequestHttpVersion.....32
 - 6.1.4.5.1 Conformance Requirement.....32
 - 6.1.4.5.2 Test area files.....33
 - 6.1.4.5.3 Test coverage.....33
 - 6.1.4.5.4 Test procedure33
 - 6.1.4.6 Method readContent.....33
 - 6.1.4.6.1 Conformance Requirement.....33
 - 6.1.4.6.1.1 Normal execution33
 - 6.1.4.6.3 Test coverage.....34
 - 6.1.4.6.4 Test procedure34
 - 6.1.4.7 Method readContentType.....35
 - 6.1.4.7.1 Conformance Requirement.....35
 - 6.1.4.7.2 Test area files.....35
 - 6.1.4.7.3 Test coverage.....36
 - 6.1.4.7.4 Test procedure36
- 6.1.5 Interface HttpResponse36
 - 6.1.5.1 Method appendContent36

6.1.5.1.1	Conformance Requirement.....	37
6.1.5.1.2	Test area files.....	37
6.1.5.1.3	Test coverage.....	37
6.1.5.1.4	Test procedure.....	38
6.1.5.2	Method appendHeaderVariable(byte[] data, short offset, short length).....	40
6.1.5.2.1	Conformance Requirement.....	40
6.1.5.2.2	Test area files.....	40
6.1.5.2.3	Test coverage.....	41
6.1.5.2.4	Test procedure.....	41
6.1.5.3	Method appendHeaderVariable (byte[] name, short nameOffset, short nameLength, byte[] value, short valueOffset, short valueLength).....	43
6.1.5.3.1	Conformance Requirement.....	43
6.1.5.3.2	Test area files.....	44
6.1.5.3.3	Test coverage.....	44
6.1.5.3.4	Test procedure.....	45
6.1.5.4	Method appendHeaderVariable (short headerKeywordNameId, byte[] value, short valueOffset, short valueLength).....	48
6.1.5.5	Method enableChunkMode.....	48
6.1.5.5.1	Conformance Requirement.....	48
6.1.5.5.2	Test area files.....	49
6.1.5.5.3	Test coverage.....	49
6.1.5.5.4	Test procedure.....	49
6.1.5.6	Method finalizeHeader.....	52
6.1.5.6.1	Conformance Requirement.....	52
6.1.5.6.2	Test area files.....	52
6.1.5.6.3	Test coverage.....	52
6.1.5.6.4	Test procedure.....	53
6.1.5.7	Method flush.....	54
6.1.5.7.1	Conformance Requirement.....	54
6.1.5.7.2	Test area files.....	54
6.1.5.7.3	Test coverage.....	55
6.1.5.7.4	Test procedure.....	55
6.1.5.8	Method getRemainingResponseBufferSize.....	55
6.1.5.8.1	Conformance Requirement.....	56
6.1.5.8.2	Test area files.....	56
6.1.5.8.3	Test coverage.....	56
6.1.5.8.4	Test procedure.....	57
6.1.5.9	Method reset.....	58
6.1.5.9.1	Conformance Requirement.....	58
6.1.5.9.2	Test area files.....	58
6.1.5.9.3	Test coverage.....	58
6.1.5.9.4	Test procedure.....	59
6.1.5.10	Method sendError.....	60
6.1.5.10.1	Conformance Requirement.....	60
6.1.5.10.2	Test area files.....	60
6.1.5.10.3	Test coverage.....	61
6.1.5.10.4	Test procedure.....	62
6.1.5.11	Method setContentType.....	64
6.1.5.11.1	Conformance Requirement.....	65
6.1.5.11.2	Test area files.....	65
6.1.5.11.3	Test coverage.....	65
6.1.5.11.4	Test procedure.....	65
6.1.5.12	Method writeStatusCode.....	68
6.1.5.12.1	Conformance Requirement.....	68
6.1.5.12.2	Test area files.....	68
6.1.5.12.3	Test coverage.....	69
6.1.5.12.4	Test procedure.....	69
6.2	SCWS Runtime Environment.....	73
6.2.1	Applet state.....	73
6.2.1.1	Invocation of applets not in state selectable.....	73
6.2.1.1.1	Conformance Requirement.....	73
6.2.1.1.2	Test area files.....	74

6.2.1.1.3 Test coverage.....74

6.2.1.1.4 Test procedure74

6.2.1.2 Registration remains valid if applet is not in selectable state.....76

6.2.2 Response sending.....76

6.2.3 Exception handling77

6.2.3.1 No exception shall be propagated as HTTP error to the terminal77

6.2.3.1.1 Conformance Requirement.....77

6.2.3.1.2 Test area files.....77

6.2.3.1.3 Test coverage.....77

6.2.3.1.4 Test procedure78

Annex A (normative): Class, methods and SCWSFramework tests acronyms.....79

A.1 Smart Card Web Server part79

A.1.1 Class ScwsExtensionRegistry79

A.1.2 Class ScwsExtensionService79

A.1.3 HttpRequest interface79

A.1.4 HttpResponse interface.....79

A.2 Acronyms for SCWS Framework tests80

A.2.1 Applet state.....80

A.2.3 Exception handling.....80

Annex B (normative): Test file description.....81

Annex C (normative): uicc.scws.test.util package and interfaces82

Annex D (normative): Test Area files.....83

Annex E (informative): HTTP-Request and HTTP-Response handling.....84

Annex F (informative): Bibliography85

Annex G (informative): Change history86

History87

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP decides to modify the contents of the present document, it will be re-released by EP SPC 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 covers the minimum characteristics considered necessary in order to provide compliance to TS 102 588 [2].

It describes the technical characteristics and methods for testing the SCWS API for Java Card™ (TS 102 588 [2]) implemented in a UICC platform. It specifies the following parts:

- test applicability;
- test environment description;
- tests format;
- test area reference;
- conformance requirements;
- test suite files;
- test procedure;
- test coverage; and
- a description of the associated testing tools that shall be used.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
 - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
 - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

- [1] ISO/IEC 9646-7 (1995): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

- [2] ETSI TS 102 588: "Smart Cards; Application invocation Application Programming Interface (API) by a UICC webserver for Java Card™ platform".
- [3] ETSI TS 102 483: "Smart cards; UICC-Terminal interface; Internet Protocol connectivity between UICC and terminal".
- [4] OMA: "Smartcard -Web Server Enable Architecture", OMA-AD-Smartcard-Web-Server-V1-0-20080421-A.
- NOTE: Available at http://www.openmobilealliance.org/technical/release_program/SCWS_v1_0.aspx.
- [5] OMA: "Smartcard-Web-Server", OMA-TS-Smartcard-Web-Server-V1-0-20080421-A.
- NOTE: Available at http://www.openmobilealliance.org/technical/release_program/SCWS_v1_0.aspx.
- [6] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Application Programming Interface".
- [7] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Runtime Environment Specification".
- [8] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Virtual Machine Specification".
- NOTE: SUN Java Card Specifications can be downloaded at <http://java.sun.com/products/javacard>.
- [9] ETSI TS 102 268: "Smart Cards; Test specification for the UICC Application Programming Interface (AP)I for Java Card™".
- [10] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".
- [11] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".
- [12] ETSI TS 102 241: "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card (TM)".
- [13] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

2.2 Informative references

The following referenced documents are not essential to the use of the ETSI deliverable but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- [i.1] ETSI TS 102 600: "Smart Cards; UICC-Terminal interface; Characteristics of the USB interface".
- [i.2] HttpClient project.

NOTE: Available at <http://hc.apache.org/httpcomponents-client/index.html>.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

applet installation parameters: values for applet installation parameters

Conformance Requirement Reference (CRR): description of the expected SCWS behaviour according to TS 102 588 [2]

expected state: state in which the UICC is supposed to be after the execution of the test procedure applied on the relevant initial conditions

security parameters: minimum security requirements defined for the applet installation process

test area: set of Test Cases applicable to a specific part (class method, SCWS RE behaviour, etc.) of TS 102 588 [2]

test case: elementary test that checks for compliance with one or more Conformance Requirement References

test procedure: sequence of actions/commands to perform all the test cases defined in a test area

test source file: java file containing methods that will load and install test applet in the card, execute and verify the test results, and restore the Default Initial Conditions on the UICC (when possible)

test toolkit applet: applet designed to test a specific functionality of the SCWS API (TS 102 588 [2])

3.2 Abbreviations

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

AID	Application Identifier
API	Application Programming Interface
BIP	Bearer Independent Protocol according to TS 102 223 [11]
CAT	Card Application Toolkit
CRR	Conformance Requirements Reference
CRRC	Conformance Requirement Reference Context Error
CRRN	Conformance Requirement Reference Normal
CRRP	Conformance Requirement Reference Parameter Error
FFS	For Further Study
IP	Internet Protocol
JCRE	Java Card™ Run-time Environment
HTTP	HyperText Transfer Protocol
RE	Runtime Environment
SCWS	Smart Card based Web Server according to OMA specifications [4] and [5]
USB	Universal Serial Bus interface according to TS 102 600 [i.1]

4 Applicability

4.1 Applicability of the present document

This present test specification applies to UICCs containing a SCWS invocation API according to TS 102 588 [2].

4.2 Applicability of the individual test cases

Table A.1 lists the optional features for which the supplier of the implementation states the support.

4.3 Applicability to different Releases

The applicability to different releases specified in table B.1 of the present document shall apply, unless otherwise specified.

4.4 Definitions

4.4.1 Format of table of the table of optional features

The columns in table A.1 have the following meaning.

Column	Meaning
Option:	The optional feature supported or not by the implementation.
Status:	See clause 4.4.3 'Status and Notations'
Support:	The support columns shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [1], are used for the support column in table A.1. Y or y supported by the implementation N or n not supported by the implementation N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status)
Mnemonic:	The mnemonic column contains mnemonic identifiers for each item.

4.4.2 Format of the applicability table

The applicability of every test in table B.1 is formally expressed by the use of Boolean expression defined in the following clause.

The columns in table B.1 have the following meaning.

Column	Meaning
Test case:	The "Test case" column gives a reference to the test case number(s) detailed in the present document and required to validate the implementation of the corresponding item in the "Description" column
Release:	The "Release" column gives the Release applicable and onwards, for the item in the "Test case" column
Support:	The "Support" column is blank in the proforma, and shall be completed by the manufacturer in respect of each particular requirement to indicate the choices, which have been made in the implementation.

4.4.3 Status and Notations

The following notations, defined in ISO/IEC 9646-7 [1], are used for the status column:

M	mandatory - the capability is required to be supported.
O	optional - the capability may be supported or not.
N/A	not applicable - in the given context, it is impossible to use the capability.
X	prohibited (excluded) - there is a requirement not to use this capability in the given context.
O.i	qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies an unique group of related optional items and the logic of their selection which is defined immediately following the table.
Ci	conditional - the requirement on the capability ("M", "O", "X" or "N/A") depends on the support of other optional or conditional items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." shall be used to avoid ambiguities.

4.5 Table of optional features

Table A.1: Options

Item	Option	Status	Support	Mnemonic
1	ScwsExtension.doGet()	M		doGet
2	ScwsExtension.doPost()	M		doPost
3	ScwsExtension.doHead()	M		doHead
4	ScwsExtension.doPut()	M		doPut
5	ScwsExtension.doDelete()	M		doDelete
6	ScwsExtension.doOptions()	O		doOptions
7	ScwsExtension.doTrace()	O		doTrace

4.6 Applicability table

Table B.1: Applicability of tests

Test case	Description	Release	Rel-7	Support
	Class ScwsExtensionRegistry			
6.1.1.1	Method register	Rel-7	M	
6.1.1.2	Method deregister	Rel-7	M	
	Class ScwsExtensionService			
6.1.3.1	Method doDelete	Rel-7	M	
6.1.3.2	Method doGet	Rel-7	M	
6.1.3.3	Method doHead	Rel-7	M	
6.1.3.4	Method doOptions	Rel-7	C001	
6.1.3.5	Method doPost	Rel-7	M	
6.1.3.6	Method doPut	Rel-7	M	
6.1.3.7	Method doTrace	Rel-7	C002	
	Interface HttpRequest			
6.1.4.1	Method findAndCopyKeywordValue	Rel-7	M	
6.1.4.2	Method findAndCopyKeywordValue	Rel-7	M	
6.1.4.3	Method getContentLength	Rel-7	M	
6.1.4.4	Method getContentType	Rel-7	M	
6.1.4.5	Method getRequestHttpVersion	Rel-7	M	
6.1.4.6	Method readContent	Rel-7	M	
6.1.4.7	Method readContentType	Rel-7	M	
	Interface HttpResponse			
6.1.5.1	Method appendContent	Rel-7	M	
6.1.5.2	Method appendHeaderVariable(byte[] data, short offset, short length)	Rel-7	M	
6.1.5.3	Method appendHeaderVariable (byte[] name, short nameOffset, short nameLength, byte[] value, short valueOffset, short valueLength)	Rel-7	M	
6.1.5.4	Method appendHeaderVariable (short headerKeywordNameId, byte[] value, short valueOffset, short valueLength)	Rel-7	M	
6.1.5.5	Method enableChunkMode	Rel-7	M	
6.1.5.6	Method finalizeHeader	Rel-7	M	
6.1.5.7	Method flush	Rel-7	M	
6.1.5.8	Method getRemainingResponseBufferSize	Rel-7	M	
6.1.5.9	Method reset	Rel-7	M	
6.1.5.10	Method sendError	Rel-7	M	
6.1.5.11	Method setContentType	Rel-7	M	
6.1.5.12	Method writeStatusCode	Rel-7	M	
	SCWS Runtime Environment: Applet state			
6.2.1.1	Invocation of applets not in state selectable	Rel-7	M	
6.2.1.2	Registration remains valid if applet is not in selectable state	Rel-7	M	
	SCWS Runtime Environment: Exception handling			
6.2.3.1	No exception shall be propagated as HTTP error to the terminal	Rel-7	M	
C001:	IF (doOptions) THEN M ELSE N/A.			
C002:	IF (doTrace) THEN M ELSE N/A.			

5 Test environment

This clause specifies requirements that shall be met and the testing rules that shall be followed during the test procedure.

The test cases described in the present document shall be performed independently of the interface used between the UICC and the terminal. In particular, any test case shall have the same result regardless of using BIP in Server Mode as defined in TS 102 223 [11] or an IP connection according to TS 102 483 [3] and TS 102 600 [i.1].

5.1 Test environment description

The general architecture for the test environment is:

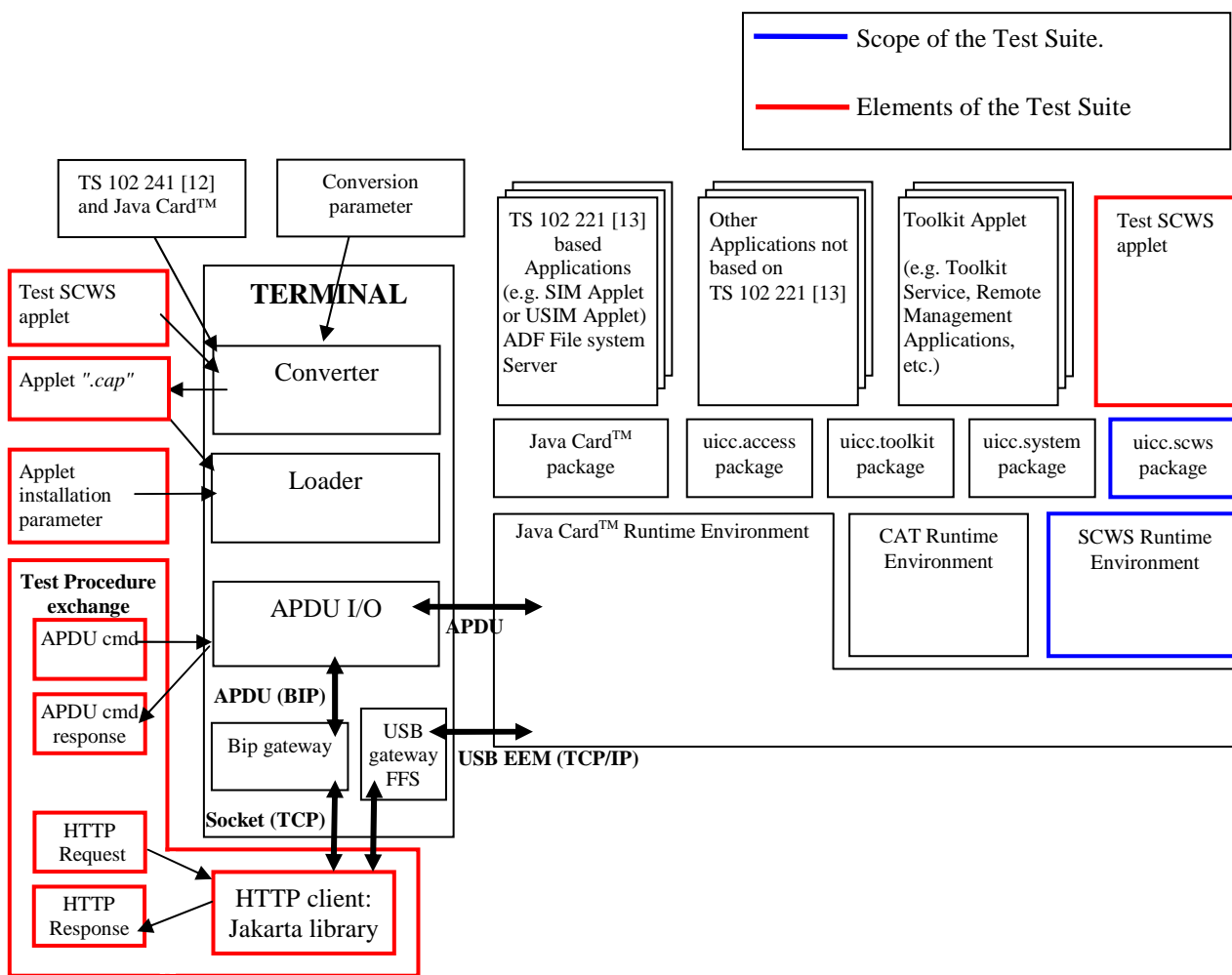


Figure 5.1

5.2 Tests format

5.2.1 Test area reference

Each test area is referenced as follows:

For API Testing: 'API_[package name]_[class name]_[method name]' where

- package name:
uicc.scws: '1'.
- class name:
yyy: 3 letters for each class.
See annex A for full classes acronyms list.
- method name:
zzzz[input parameters]:
See annex A for full methods name acronyms list.

For SCWS Framework testing: 'API_2_[clause name]_[subclause name]' where

- Clause name:
xxx: 3 letters for each clause.
See annex A for full clause acronyms list.
- Subclause name:
yyyy: 4 letters for each subclause.
See annex A for full subclause acronyms list.

5.2.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in TS 102 588 [2].
- Normal execution:
 - Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN).
- Parameters error:
 - Contains parameter errors and incorrect parameter limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP).
- Context error:
 - Contains errors due to the context the method is used in, each referenced as a Conformance Requirement Reference Context Error (CRRC).

5.2.1.2 Test area files

The files included in the Test Area use the following naming convention:

- Test Source: Test_[Test Area Reference].java
- Test Applet: [Test Area Reference]_[Test applet number].java
- Cap File: [Test Area Reference].cap

The applet numbers start from '1'.

The test source shall use the common interfaces defined in annex C.

The Cap File format is described in Java Card™ Virtual Machine Specification [8].

Test files can be run in any order.

All files from the same test area are located in the same subfolder.

5.2.1.3 Test procedure

Each test procedure contains a table to indicate the expected responses form the API and/or the HTTP level as follows:

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
	<i>Test Case detailed description</i>	<i>API and/or SCWS Framework expected behaviour.</i>	<i>Expected response at HTTP level.</i>

5.2.1.4 Test coverage

The table above each test procedure indicates the correspondence between the Conformance Requirements Reference (CRR) and the different test cases.

5.3 Initial conditions

The Initial Conditions are a set of general prerequisites for the (U)SIM prior to the execution of testing. For each test procedure described in the present document, the following rules apply to the Initial Conditions:

- unless otherwise stated, before installing the applet(s) relevant to the current test procedure, all packages specific to other test procedures shall not be present.

Prior to any test the test environment shall have the (U)SIM powered on and performed the PROFILE DOWNLOAD procedure.

5.4 Package name

Java packages integrating this Test Suite shall follow this naming convention:

uicc.scws.test.[Test Area Reference]: Java Card packages containing Test Area References for the TS 102 588 [2] uicc.scws package.

uicc.scws.test.scwsframework.[Test Area Reference]: Java Card packages containing Test Area References for the TS 102 588 [2] SCWS Framework.

uicc.scws.test.util: for the Test util package defined in this Test Suite. Requires package uicc.test.util from TS 102 268 [9].

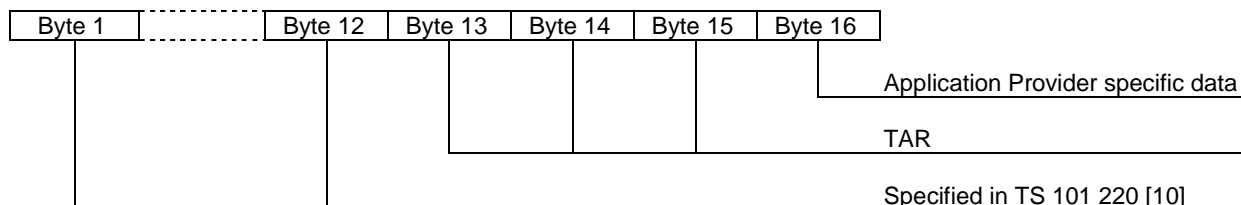
where the Test Area Reference is written in lower case.

EXAMPLE: The package `./uicc.scws.test.[Test Area Reference]` creates the following directory structure `./uicc/scws/test/[Test Area Reference]/Api_1..._[1..n].*`, where `'Api_1..._[1..n].*'` are the different test applets Java source files used in `[Test Area Reference]`.

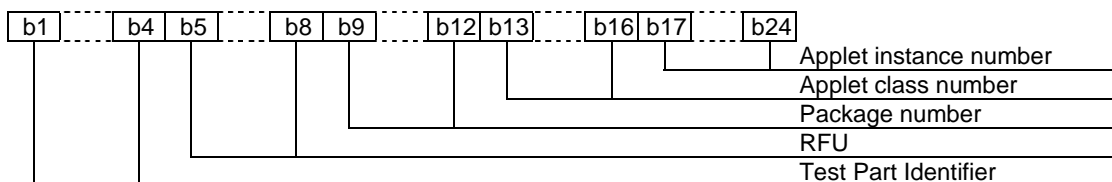
5.5 AID coding

The AID coding for the Test Packages, Applet classes and Applets shall be as specified in TS 101 220 [10]. In addition, the following TAR and Application Provider specific data values are defined for use within the present document:

AID coding



TAR coding (3 bytes/ 24 bits):



Applet instance number, Applet Class number, Package number:

- For package AID, package number shall start from 0 and class and instance numbers shall be 0.
- For class AID, package number is the number of the class package, class number shall start from 1 and instance shall be 0.
- For instance AID, package and class number are the number of class and package of which instance belongs, and instance number shall start from 1.

Test part Identifier (bits b1-b4):

- 0000 reserved (as TAR= '00.00.00' is reserved for Issuer Security Domain).
- 0010 API uicc.scws.
- 0101 SCWS Framework.
- other values are RFU.

Application Provider specific data (1 byte):

- '00' for Package.
- '01' for Applet class.
- '02' for Applet Instance.

EXAMPLE: The AID of Package uicc.scws is 'A0 00 00 00 87 10 05 FF FF FF FF 89 20 00 00 00'.

5.6 Test equipment

These clauses recommend a minimum specification for each of the items of test equipment referenced in the tests.

5.6.1 Test tool

This test tool shall meet the following requirements:

- be able to send and receive HTTP messages to the SCWS in the UICC;
- the result of I/O commands must be presented at the application layer;
- be able to provide results of the tests;
- shall send and/or compare all HTTP messages specified in the test files.

5.6.2 Interfaces and classes use

The SCWS test tool extends the UICC test tool defined in TS 102 268 [9]. Therefore the SCWS test tool cannot be run without having implemented the UICC test tool.

The SCWS test tool shall use the interfaces and classes as defined in Annex C. They define the only allowed methods to write the test sources.

Interfaces and classes are defined as follow:

- ScwsApplicationManagementService defines methods to manage applications;
- ScwsCardManagementService defines methods to manage the UICC;
- ScwsToolkitService defines methods to manage toolkit commands;
- ScwsHttpManagement defines methods to send and receive HTTP messages;
- ScwsAPITestCardService defines the static method to get a reference of the class implementing all interfaces.

5.6.3 Util package

Annex C includes java source code of TestScwsApplet abstract class of the uicc.scws package. Each test applet shall extend this abstract class in order to retrieve test results when selecting it.

5.6.4 Java Software Development Kit

Java Card™ software development kit (SDK) version supported by Java Card 2.2.2 specifications ([6], [7],[8]) is 1.4.1.

6 Test Cases

6.1 class uicc.scws

6.1.1 Class ScwsExtensionRegistry

6.1.1.1 Method register

Test Area Reference: API_1_Ser_Regi.

6.1.1.1.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```

static void register(ScwsExtension scwsExtension,
                    byte[] appId,
                    short offset,
                    short length)
    throws java.lang.NullPointerException,
           java.lang.ArrayIndexOutOfBoundsException,
           ScwsException

```

6.1.1.1.1.1 Normal execution

- CRRN1: registers the scwsExtension object to the SCWS.

6.1.1.1.1.2 Parameter Errors

- CRRP1: if scwsExtension is null, a NullPointerException is thrown.
- CRRP2: if appId is null, a NullPointerException is thrown.
- CRRP3: if offset or length or both would cause access outside of the array bounds, or if length is negative an ArrayIndexOutOfBoundsException is thrown.

6.1.1.1.1.3 Context errors

- CRRC1: if the appId is already registered, an ScwsException with reason code SCWSEXTENSION_REGISTRY_ERROR is thrown.

6.1.1.1.2 Test area files

Specific triggering: Unrecognized envelope:

- Test Source: Test_Api_1_Ser_Regi.java.
- Test Applet: Api_1_Ser_Regi_1.java.
- Cap File: Api_1_Ser_Regi.cap.

6.1.1.1.3 Test coverage

CRR number	Test case number
N1	8
P1	1
P2	2
P3	3, 4, 5, 6, 7
C1	9, 10

6.1.1.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an object implementing the <code>scwsExtension</code> interface		
1	Null <code>scwsExtension</code> <code>register()</code>	<code>NullPointerException</code> is thrown	
2	Null <code>appld</code> <code>register()</code>	<code>NullPointerException</code> is thrown	
3	<code>offset > appld.length</code> <code>register()</code> <code>appId.length = 5</code> <code>offset = 6</code> <code>length = 0</code>	<code>ArrayIndexOutOfBoundsException</code> is thrown	
4	<code>offset < 0</code> <code>register()</code> <code>appId.length = 5</code> <code>offset = -1</code> <code>length = 1</code>	<code>ArrayIndexOutOfBoundsException</code> is thrown	
5	<code>length > appld.length</code> <code>register()</code> <code>appId.length = 5</code> <code>offset = 0</code> <code>length = 6</code>	<code>ArrayIndexOutOfBoundsException</code> is thrown	
6	<code>offset + length > appld.length</code> <code>register()</code> <code>appId.length = 5</code> <code>offset = 3</code> <code>length = 3</code>	<code>ArrayIndexOutOfBoundsException</code> is thrown	
7	<code>length < 0</code> <code>register()</code> <code>appId.length = 5</code> <code>offset = 0</code> <code>length = -1</code>	<code>ArrayIndexOutOfBoundsException</code> is thrown	
	register the object implementing the <code>scwsExtension</code> interface with the name "Name1" to the SCWS.		
8	successful registration <code>register()</code> <code>appId = (byte) 'N', (byte) 'a', (byte) 'm', (byte) 'e', (byte) '1'</code> <code>offset = 0</code> <code>length = 5</code>		
	register the same object a second time with the same name		
9	unsuccessful registration <code>register()</code> <code>appId = (byte) 'N', (byte) 'a', (byte) 'm', (byte) 'e', (byte) '1'</code> <code>offset = 0</code> <code>length = 5</code>	<code>ScwsException</code> with reason code <code>SCWSEXTENSION_REGISTRY_ERROR</code> is thrown	
	retrieve a second object implementing the <code>scwsExtension</code> interface		
10	register different object with the same <code>appld</code> <code>register()</code> <code>appId = (byte) 'N', (byte) 'a', (byte) 'm', (byte) 'e', (byte) '1'</code> <code>offset = 0</code> <code>length = 5</code>	<code>ScwsException</code> with reason code <code>SCWSEXTENSION_REGISTRY_ERROR</code> is thrown	

6.1.1.2 Method deregister

Test Area Reference: `API_1_Ser_Dreg`, `Api_1_Ser_Dreg_Helper`

6.1.1.2.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
static void deregister(ScwsExtension scwsExtension)
    throws java.lang.NullPointerException,
           ScwsException,
           java.lang.SecurityException
```

6.1.1.2.1.1 Normal execution

- CRRN1: deregisters the scwsExtension object from the SCWS.

6.1.1.2.1.2 Parameter Errors

- CRRP1: if scwsExtension is null, a NullPointerException is thrown.

6.1.1.2.1.3 Context errors

- CRRC1: if the scwsExtension was not registered before, an ScwsException with reason code SCWSEXTENSION_REGISTRY_ERROR is thrown.
- CRRC2: if the scwsExtension was previously registered by a different JCRE context than it is used for the deregistration, a java.lang.SecurityException shall be thrown.
- CRRC3: if the scwsExtension was already deregistered, an ScwsException with reason code SCWSEXTENSION_REGISTRY_ERROR is thrown.

6.1.1.2.2 Test area files

Specific triggering: Unrecognized envelope:

- Test Source: Test_Api_1_Ser_Dreg.java.
- Test Applet: Api_1_Ser_Dreg_1.java.
Api_1_Ser_Dreg_Helper.java.
- Cap File: Api_1_Ser_Dreg.cap.
- Cap File: Api_1_Ser_Dreg_Helper.cap.

6.1.1.2.3 Test coverage

CRR number	Test case number
N1	3
P1	2
C1	1
C2	5
C3	4

6.1.1.2.4 Test procedure

Id	Description	Test Case	
		API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an object implementing the scwsExtension interface.		
1	No previous registration deregister() Register the object to the SCWS	ScwsException with reason code SCWSEXTENSION_REGISTRY_ERROR is thrown	
2	Null scwsExtension deregister() scwsExtension = null	NullPointerException is thrown	

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
3	Successful deregistration deregister()		
4	Second deregistration deregister()	ScwsException with reason code SCWSEXTENSION_REGISTRY_ERROR is thrown	
5	Different context Request the SCWSExtension object from a different context and deregister it.	java.lang.SecurityException is thrown	

6.1.2 Interface ScwsExtension

This interface is implicitly tested by the tests covering the SCWSExtensionService.

6.1.3 Class ScwsExtensionService

6.1.3.1 Method doDelete

Test Area Reference: Api_1_Ses_Ddel.

6.1.3.1.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doDelete(HttpServletRequest request,
                    HttpServletResponse response)
    throws ScwsException
```

6.1.3.1.1.1 Normal execution

- n/a.

6.1.3.1.1.2 Parameter Errors

- n/a.

6.1.3.1.1.3 Context errors

- CRRC1: if a ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application then an appropriate HTML status page is returned by the SCWS (status code 5xx, 4xx).

6.1.3.1.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Ddel.java.

Test Applet: Api_1_Ses_Ddel_1.java.

Cap File: Api_1_Ses_Ddel.cap.

6.1.3.1.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doDeleteScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Ddel" in the SCWS and send a HTTP Delete request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.2 Method doGet

Test Area Reference: Api_1_Ses_Dget.

6.1.3.2.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doGet(HttpRequest request,
                 HttpResponse response)
    throws ScwsException
```

6.1.3.2.1.1 Normal execution

- n/a.

6.1.3.2.1.2 Parameter Errors

- n/a.

6.1.3.2.1.3 Context errors

- CRRC1: if an ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.2.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dget.java.

Test Applet: Api_1_Ses_Dget_1.java.

Cap File: Api_1_Ses_Dget.cap.

6.1.3.2.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.2.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doGetScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dget" in the SCWS and send a HTTP Get request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.3 Method doHead

Test Area Reference: Api_1_Ses_Dhea.

6.1.3.3.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doHead(HttpRequest request,
                  HttpResponse response)
    throws ScwsException
```

6.1.3.3.1.1 Normal execution

- n/a.

6.1.3.3.1.2 Parameter Errors

- n/a.

6.1.3.3.1.3 Context errors

- CRRC1: if an ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.3.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dhea.java.

Test Applet: Api_1_Ses_Dhea_1.java.

Cap File: Api_1_Ses_Dhea.cap.

6.1.3.3.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.3.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doHeadScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dhea" in the SCWS and send a HTTP Head request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.4 Method doOptions

Test Area Reference: Api_1_Ses_Dopt.

6.1.3.4.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doOptions(HttpServletRequest request,
                    HttpServletResponse response)
    throws ScwsException
```

6.1.3.4.1.1 Normal execution

- n/a.

6.1.3.4.1.2 Parameter Errors

- n/a.

6.1.3.4.1.3 Context errors

- CRR1: if a ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application (which is Intercept or Non Intercept Application) then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.4.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dopt.java.

Test Applet: Api_1_Ses_Dopt_1.java.

Cap File: Api_1_Ses_Dopt.cap.

6.1.3.4.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.4.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doOptionsScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dopt" in the SCWS and send a HTTP Options request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.5 Method doPost

Test Area Reference: Api_1_Ses_Dpos.

6.1.3.5.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doPost(HttpServletRequest request,
                  HttpServletResponse response)
    throws ScwsException
```

6.1.3.5.1.1 Normal execution

- n/a.

6.1.3.5.1.2 Parameter Errors

- n/a.

6.1.3.5.1.3 Context errors

- CRRC1: if an ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.5.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dpos.java.

Test Applet: Api_1_Ses_Dpos_1.java.

Cap File: Api_1_Ses_Dpos.cap.

6.1.3.5.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.5.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doPostScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dpos" in the SCWS and send a HTTP Post request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.6 Method doPut

Test Area Reference: Api_1_Ses_Dput.

6.1.3.6.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doPut(HttpRequest request,
                 HttpResponse response)
    throws ScwsException
```

6.1.3.6.1.1 Normal execution

- n/a.

6.1.3.6.1.2 Parameter Errors

- n/a.

6.1.3.6.1.3 Context errors

- CRR1: if a ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application (which is Intercept or Non Intercept Application) then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.6.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dput.java.

Test Applet: Api_1_Ses_Dput_1.java.

Cap File: Api_1_Ses_Dput.cap.

6.1.3.6.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.6.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doPutScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dput" in the SCWS and send a HTTP Put request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS

6.1.3.7 Method doTrace

Test Area Reference: Api_1_Ses_Dtra.

6.1.3.7.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public void doTrace(HttpServletRequest request,
                  HttpServletResponse response)
    throws ScwsException
```

6.1.3.7.1.1 Normal execution

- n/a.

6.1.3.7.1.2 Parameter Errors

- n/a.

6.1.3.7.1.3 Context errors

- CRR1: if a ScwsException with reason code METHOD_NOT_SUPPORTED was thrown by the application (which is Intercept or Non Intercept Application) then an appropriate HTML status page is returned by the SCWS (status code 4xx, 5xx).

6.1.3.7.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Ses_Dtra.java.

Test Applet: Api_1_Ses_Dtra_1.java.

Cap File: Api_1_Ses_Dtra.cap.

6.1.3.7.3 Test coverage

CRR number	Test case number
C1	1

6.1.3.7.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
0	Instantiate an doTraceScwsExtensionServiceObject object and register to the SCWS. Map the object to the path "/Test_Api_1_Ses_Dtra" in the SCWS and send a HTTP Trace request to the SCWS triggering the application.		
1	Throw ScwsException with reason code METHOD_NOT_SUPPORTED		Appropriate status code (4xx, 5xx) is returned by the SCWS.

6.1.4 Interface HttpRequest

6.1.4.1 Method findAndCopyKeywordValue

Test Area Reference: Api_1_Hrq_Fckw.

6.1.4.1.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
short findAndCopyKeywordValue(byte[] headerKeywordName,
                             short nameOffs,
                             short nameLength,
                             byte[] buffer,
                             short bufferOffs,
                             short maxLength)
    throws: java.lang.NullPointerException,
           java.lang.ArrayIndexOutOfBoundsException.
```

6.1.4.1.1.1 Normal execution

- CRRN1: Find and copy the value of a header.
- CRRN2: bufferOffs+length of the copied value or KEYWORD_NOT_FOUND in case the keyword is not found.

6.1.4.1.1.2 Parameter Errors

- CRRP1: if headerKeywordName is equal to null a java.lang.NullPointerException is thrown.
- CRRP2: if buffer is equal to null a NullPointerException is thrown.
- CRRP3: if the copy operation would cause access out of bounds an ArrayIndexOutOfBoundsException is thrown.

6.1.4.1.1.3 Context errors

- n/a.

6.1.4.1.2 Test area files

Specific triggering: Unrecognized envelope:

- Test Source: Test_Api_1_Hrq_Fckw_Bss_Bss.java.
- Test Applet: Api_1_Hrq_Fckw_Bss_Bss_1.java.
- Cap File: Api_1_Hrq_Fckw_Bss_Bss.cap.

6.1.4.1.3 Test coverage

CRR number	Test case number
P1	1
P2	2
P3	3, 4, 5, 6, 7
N1	9
N2	8

6.1.4.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	findAndCopyKeywordValue () with a null headerKeywordName	NullPointerException is thrown	
2	findAndCopyKeywordValue () with a null buffer	NullPointerException is thrown	
3	bufferOffs ≥ buffer.length findAndCopyKeywordValue () nameOffs = 0 buffer.length = 2 bufferOffs = 2 maxLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	bufferOffs < 0 findAndCopyKeywordValue() buffer.length = 2 bufferOffs = -1 maxLength = 1	ArrayIndexOutOfBoundsException is thrown	
5	maxLength > buffer.length findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 0 maxLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	bufferOffs + maxLength > buffer.length findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 1 maxLength = 2	ArrayIndexOutOfBoundsException is thrown	
7	maxLength < 0 findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 0 maxLength = -1	ArrayIndexOutOfBoundsException is thrown	
8	findAndCopyKeywordValue() headerKeywordName= HEADER_AB [This header is not present in the request] bufferOffs = 0	ScwsConstants.KEYWORD_NOT_FOUND	
9	findAndCopyKeywordValue() headerKeywordName= HEADER_AB [This header is present in the request] bufferOffs = 1	length of Header AB (1) + bufferOffs (1) = 2	

6.1.4.2 Method findAndCopyKeywordValue

Test Area Reference: Api_1_Hrq_Fckw.

6.1.4.2.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```

short findAndCopyKeywordValue(short keywordId,
                              byte[] buffer,
                              short bufferOffs,
                              short maxLength)
    throws ScwsException,
           NullPointerException,
           ArrayIndexOutOfBoundsException

```

6.1.4.2.1.1 Normal execution

- CRRN1 Find and copy the value of a header or the value of a keyword/value part of the URI indicated by a keywordId. The supported keywords are defined in ScwsConstants Interface.

6.1.4.2.1.2 Parameter Errors

- CRRP1 KEYWORD_NOT_FOUND in case the keyword is not found.
- CRRP2 if the keywordId is not recognized a ScwsException - with reason UNKNOWN_KEYWORD_ID.
- CRRP3: if buffer is equal to null a NullPointerException is thrown.
- CRRP4: if the copy operation would cause access out of bounds an ArrayIndexOutOfBoundsException is thrown.

6.1.4.2.1.3 Context errors

- n/a.

6.1.4.2.2 Test area files

Specific triggering: Unrecognized envelope:

- Test Source: Test_Api_1_Hrq_Fckws_Bss.java.
- Test Applet: Api_1_Hrq_Fckws_Bss_1.java.
- Cap File: Api_1_Hrq_Fckws_Bss.cap.

6.1.4.2.3 Test coverage

CRR number	Test case number
P1	1
P2	2
P3	19
P4	20, 21, 22, 23, 24
N1	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

6.1.4.2.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_ACCEPT_ENCODING [not present in request] bufferOffs = 0	ScwsConstants.KEYWORD_NOT_FOUND	
2	findAndCopyKeywordValue() keywordId = 99 (Unknown) bufferOffs = 0	ScwsException - with reason UNKNOWN_KEYWORD_ID	
3	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_ACCEPT [present in request] bufferOffs = 0		
4	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_ACCEPT_CHARSET [present in request] bufferOffs = 0		
5	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_ACCEPT_ENCODING [present in request] bufferOffs = 0		
6	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_ACCEPT_LANGUAGE [present in request] bufferOffs = 0		
7	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_HOST [present in request] bufferOffs = 0		
8	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_USER_AGENT [present in request] bufferOffs = 0		
9	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_CONTENT_ENCODING [present in request] bufferOffs = 0		
10	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_CONTENT_LANGUAGE [present in request] bufferOffs = 0		
11	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_CONTENT_LENGTH [present in request] bufferOffs = 0		
12	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_CONTENT_TYPE [present in request] bufferOffs = 0		
13	findAndCopyKeywordValue() keywordId = ScwsConstants.HEADER_SERVER [present in request] bufferOffs = 0		
14	findAndCopyKeywordValue() keywordId = ScwsConstants.URI_TAG [present in request] bufferOffs = 0		
15	findAndCopyKeywordValue() keywordId = ScwsConstants.URI_SCHEMA_TAG [present in request] bufferOffs = 0		
16	findAndCopyKeywordValue() keywordId = ScwsConstants.URI_AUTHORITY_TAG [present in request] bufferOffs = 0		
17	findAndCopyKeywordValue() keywordId = ScwsConstants.URI_PATH_TAG [present in request] bufferOffs = 0		
18	findAndCopyKeywordValue() keywordId = ScwsConstants.URI_QUERY_TAG [present in request] bufferOffs = 0		

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
19	findAndCopyKeywordValue() with a null buffer	NullPointerException is thrown	
20	bufferOffs ≥ buffer.length findAndCopyKeywordValue () buffer.length = 2 bufferOffs = 3 maxLength = 1	ArrayIndexOutOfBoundsException is thrown	
21	bufferOffs < 0 findAndCopyKeywordValue() buffer.length = 2 bufferOffs = -1 maxLength = 1	ArrayIndexOutOfBoundsException is thrown	
22	maxLength > buffer.length findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 0 maxLength = 3	ArrayIndexOutOfBoundsException is thrown	
23	bufferOffs + maxLength > buffer.length findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 1 maxLength = 2	ArrayIndexOutOfBoundsException is thrown	
24	maxLength < 0 findAndCopyKeywordValue() buffer.length = 2 bufferOffs = 0 maxLength = -1	ArrayIndexOutOfBoundsException is thrown	

6.1.4.3 Method getContentLength

This test case is FFS.

6.1.4.4 Method getContentType

This test case is FFS.

6.1.4.5 Method getRequestHttpVersion

Test Area Reference: Api_1_Hrq_Grhv

6.1.4.5.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
short getRequestHttpVersion()
```

6.1.4.5.1.1 Normal execution

- CRRN1: Returns the version of the HTTP protocol of the current request.

6.1.4.5.1.2 Parameter Errors

- None.

6.1.4.5.1.3 Context errors

- None.

6.1.4.5.2 Test area files

Specific triggering: Unrecognized envelope:

- Test Source: Test_Api_1_Hrq_Grhv.java.
- Test Applet: Api_1_Hrq_Grhv_1.java.
- Cap File: Api_1_Hrq_Grhv.cap.

6.1.4.5.3 Test coverage

CRR number	Test case number
N1	1, 2

6.1.4.5.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	Normal execution Send a HTTP request with protocol version set to: HTTP/1.0.	Returns ScwsConstants.HTTP_PROTOCOL_VERSION_10	
2	Normal execution Send a HTTP request with protocol version set to: HTTP/1.1.	Returns ScwsConstants.HTTP_PROTOCOL_VERSION_11	

6.1.4.6 Method readContent

Test Area Reference: Api_1_Hrq_Rcon.

6.1.4.6.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
short readContent(int requestOffset,
                 byte[] data,
                 short offset,
                 short len)
    throws: ScwsException
           java.lang.NullPointerException
           java.lang.ArrayIndexOutOfBoundsException
```

6.1.4.6.1.1 Normal execution

- CRRN1: Read a part of the request content and copy it in the data buffer.

6.1.4.6.1.2 Parameter Errors

- CRRP1: if the data buffer is equal to null, a java.lang.Null PointerException is thrown.
- CRRP2: if the copy operation would cause access out of bounds, a java.lang.ArrayIndexOutOfBoundsException will be thrown.
- CRRP3: if the requested offset is out of the request bounds, a ScwsException.UNREACHABLE_OFFSET will be thrown.

6.1.4.6.1.3 Context errors

- None.

6.1.4.6.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Hrq_Rcon.java.

Test Applet: Api_1_Hrq_Rcon_1.java.

Cap File: Api_1_Hrq_Rcon.cap.

6.1.4.6.3 Test coverage

CRR number	Test case number
N1	1, 2
P1	3
P2	4, 5, 6, 7, 8
P3	9, 10

6.1.4.6.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Normal execution</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() requestOffset = 0 offset = 0 len = 0x10</pre>	Data buffers contains the first bytes of the request content	
2	<p>Normal execution</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() requestOffset = 0 offset = 0 len = 0x20</pre>	Data buffers contains the entire request content	
3	<p>Null pointer exception</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() data[] is set to null</pre>	NullPointerException is thrown	
4	<p>offset > data.length</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() data.length = 0x20 offset = 0x20 length = 0x10</pre>	ArrayIndexOutOfBoundsException is thrown	
5	<p>length > data.length</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() data.length = 0x1F offset = 0x00 length = 0x20</pre>	ArrayIndexOutOfBoundsException is thrown	
6	<p>offset + length > data.length</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() data.length = 0x20 offset = 0x10 length = 0x11</pre>	ArrayIndexOutOfBoundsException is thrown	
7	<p>length < 0</p> <p>Send a request with Content-length set to 0x20</p> <pre>readContent() data.length = 0x20 offset = 0x00 length = -1</pre>	ArrayIndexOutOfBoundsException is thrown	

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
8	offset < 0 Send a request with Content-length set to 0x20 readContent() data.length = 0x20 offset = -1 length = 0x10	ArrayIndexOutOfBoundsException is thrown	
9	requestOffset > request length Send a request with Content-length set to 0x20 readContent() Request length is 0x20 requestOffset = 0x20 data.length = 0x20 offset = 0 length = 0x10	ScwsException.UNREACHABLE_OFFSET is thrown	
10	requestOffset < 0 Send a request with Content-length set to 0x20 readContent() Request length is 0x20 requestOffset = -1 data.length = 0x20 offset = 0 length = 0x10	ScwsException.UNREACHABLE_OFFSET is thrown	

6.1.4.7 Method readContentType

Test Area Reference: Api_1_Hrq_Rcty.

6.1.4.7.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
short readContentType(byte[] data,
                    short offs,
                    short len)
    throws: java.lang.NullPointerException
           java.lang.ArrayIndexOutOfBoundsException
```

6.1.4.7.1.1 Normal execution

- CRRN1: Reads the content type given in the HTTP request and copy it in the data[] buffer. Returns the offset of the last data written in the buffer.

6.1.4.7.1.2 Parameter Errors

- CRRP1: if the data buffer is null, a java.lang.NullPointerException will be thrown.
- CRRP2: if the copy operation would cause access out of bounds, a java.lang.ArrayIndexOutOfBoundsException will be thrown.

6.1.4.7.1.3 Context errors

- None.

6.1.4.7.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_1_Hrq_Rcty.java.

Test Applet: Api_1_Hrq_Rcty_1.java.

Cap File: Api_1_Hrq_Rcty.cap.

6.1.4.7.3 Test coverage

CRR number	Test case number
N1	1, 2
P1	3
P2	4, 5, 6, 7, 8

6.1.4.7.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	Normal execution Send a HTTP request with content type set to: text/html. readContentType() data.length = 0x20 offs = 0x00 len = 0x0A	text/html is written in the data buffer and the method returns 0x08	
2	Normal execution Send a HTTP request with content type set to: text/html. readContentType() data.length = 0x20 offs = 0x00 len = 0x05	text/ is written in the data buffer and the method returns 0x04	
3	Null pointer Send a HTTP request with content type set to: text/html readContentType() data = null offs = 0x00 len = 0x0A	NullPointerException is thrown	
4	offs > data.length Send a HTTP request with content type set to: text/html. readContentType() data.length = 0x20 offs = 0x20 len = 0x10	ArrayIndexOutOfBoundsException is thrown	
5	len > data.length Send a HTTP request with content type set to: text/html. readContentType() data.length = 0x05 offs = 0x00 len = 0x09	ArrayIndexOutOfBoundsException is thrown	
6	offs + len > data.length Send a HTTP request with content type set to: text/html. readContentType() data.length = 0x05 offs = 0x01 len = 0x05	ArrayIndexOutOfBoundsException is thrown	
7	len < 0 Send a HTTP request with content type set to: text/html. readContent() data.length = 0x20 offs = 0x00 Len = -1	ArrayIndexOutOfBoundsException is thrown	
8	offs < 0 Send a HTTP request with content type set to: text/html. readContent() data.length = 0x20 offs = -1 Len = 0x10	ArrayIndexOutOfBoundsException is thrown	

6.1.5 Interface HttpResponse

6.1.5.1 Method appendContent

Test Area Reference: Api_1_Hrs_Acon.

6.1.5.1.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void appendContent(byte[] data,
                  short offset,
                  short length)
    throws ScwsException
```

6.1.5.1.1.1 Normal execution

- CRRN1: Writes the content of `data` into the `HttpResponse`.
- CRRN2: Invoking this method implicitly finalizes the header of the `HttpResponse`.
- CRRN3: In case of chunked transfer encoding hex length of chunked data is automatically added by the framework.

6.1.5.1.1.2 Parameter Errors

- CRRP1: if `data` is null, a `java.lang.NullPointerException` is thrown.
- CRRP2: if parameter `offset` is negative, a `java.lang.ArrayIndexOutOfBoundsException` is thrown.
- CRRP3: if parameter `length` is negative, a `java.lang.ArrayIndexOutOfBoundsException` is thrown.
- CRRP4: if parameter `offset` plus `length` are greater than the length of array `data`, a `java.lang.ArrayIndexOutOfBoundsException` is thrown.

6.1.5.1.1.3 Context errors

- CRRC1: if appending the content would cause an overflow of the response buffer, a `uicc.scws.ScwsException` with reason codes `BUFFER_OVERFLOW` is thrown.
- CRRC2: if the flush method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.
- CRRC3: if the `sendError` method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.

6.1.5.1.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Acon.java.

Test Applet: Api_1_Hrs_Acon_1.java.

Cap File: Api_1_Hrs_Acon.cap.

6.1.5.1.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 14
N2	4
N3	2, 14
P1	6
P2	7
P3	8
P4	9
C1	12, 13
C2	10
C3	11

6.1.5.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>append chunk off</p> <pre>writeStatusCode(200) finalizeHeader() appendContent (data = [Hello World] offset = 0 length = 6) appendContent (data = [Hello World] offset = 6 length = 5) flush()</pre>		<pre>HTTPResponse (statusCode = "200" Header("Content-Length") = "11" responseBody = "Hello World")</pre>
2	<p>append chunk on</p> <pre>writeStatusCode(200) enableChunkMode() finalizeHeader() appendContent (data = [Hello World] offset = 0 length = 6) appendContent (data = [Hello World] offset = 6 length = 5) flush()</pre>		<pre>HTTPResponse (statusCode = "200" Header("Transfer-Encoding") = "chunked" responseBody = "Hello World")</pre>
3	<p>append up to remaining buffer length</p> <pre>writeStatusCode(200) finalizeHeader() appendContent (data = 'aaa...' offset = 0 length = remainingBufferLength) flush()</pre>		<pre>HTTPResponse (statusCode = "200" Header("Content-Length") = <i>bufferLength</i> responseBody = "aaa...")</pre>
4	<p>append implicitly finalizes the header</p> <pre>writeStatusCode(200) appendContent (data= [Hello World] offset= 0 length= 11) flush()</pre>		<pre>HTTPResponse (statusCode = "200" Header("Content-Length") = "11" responseBody = "Hello World")</pre>
5	<p>length = 0</p> <pre>writeStatusCode(200) finalizeHeader() appendContent (data = [Hello World] offset= 0 length= 0) flush()</pre>		<pre>HTTPResponse (statusCode = "200" Header("Content-Length") = "0" responseBody = "")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
6	data[] is null appendContent (data= null offset = 0 length = 10)	java.lang.NullPointerException is thrown : Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
7	offset<0 appendContent (data = [Hello World] offset = -1 length = 10)	java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
8	length<0 appendContent (data = [Hello World] offset = 0 length = -1)	java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
9	offset + length > data.length appendContent (data = [Hello World] offset = 10 length = 11)	java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
10	flush called previously writeStatusCode(200) finalizeHeader() appendContent (data = 'Hello' offset = 0 length = 5) flush() appendContent (data = 'World' offset = 0 length = 5)	} uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode = "200" Header("Content-Length") = "5" contentBody = "Hello")
11	sendError called previously sendError(500) appendContent (data = 'World' offset = 0 length = 5)	} uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode = "500")
12	response buffer overflow writeStatusCode(200) finalizeHeader() appendContent (data = 'aaa...' offset = 0 length = responseBuffer+1)	} uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode = "200" Header("Content-Length") = "0" contentBody = "")
13	Void		
14	Void		
15	Not present writeStatusCode(200) finalizeHeader() flush()		HTTPResponse (statusCode = "200" Header("Content-Length") = "0" ContentBody = "")

6.1.5.2 Method appendHeaderVariable(byte[] data, short offset, short length)

Test Area Reference: Api_1_Hrs_Ahva_Bss.

6.1.5.2.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void appendHeaderVariable(byte[] data,  
                          short offset,  
                          short length)  
    throws ScwsException
```

6.1.5.2.1.1 Normal execution

- CRRN1: Writes a header variable to outgoing buffer, "CRLF" will be appended automatically.
- CRRN2: The header variables content-length or transfer-encoding are inserted by the framework depending on the transfer mode.

6.1.5.2.1.2 Parameter Errors

- CRRP1: if data is null, a java.lang.NullPointerException is thrown.
- CRRP2: if parameter offset is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP3: if parameter length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP4: if parameter offset plus length are greater than the length of array data, a java.lang.ArrayIndexOutOfBoundsException is thrown.

6.1.5.2.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown.
- CRRC2: if appending the header variable would cause an overflow of the response buffer, a uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown.
- CRRC3: if the flush method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.
- CRRC4: if the sendError method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.

6.1.5.2.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Ahva_Bss.java.

Test Applet: Api_1_Hrs_Ahva_Bss_1.java.

Cap File: Api_1_Hrs_Ahva_Bss.cap.

6.1.5.2.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
N2	1, 2, 3, 4
P1	5
P2	6
P3	7
P4	8
C1	9, 10
C2	11
C3	12
C4	13

6.1.5.2.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Basic</p> <pre>writeStatusCode(200) appendHeaderVariable(data= [A: B], offset= 0, length= 4) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode = "200" Header("A") = "B" Header("Content-Length") = "2" responseBody = "OK")</pre>
2	<p>offset>0</p> <pre>writeStatusCode(200) appendHeaderVariable(data= [A: BC: Dd], offset= 4, length= 5) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("C")= "Dd" Header("Content-Length") = "2" responseBody= "OK")</pre>
3	<p>No SP char</p> <pre>writeStatusCode(200) appendHeaderVariable(data= [A:B], offset= 0, length= 3) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" responseBody= "OK")</pre>
4	<p>length = 0</p> <pre>writeStatusCode(200) appendHeaderVariable(data= [A: B], offset = 0, length = 0) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" responseBody= "OK")</pre>
5	<p>data[] is null</p> <pre>appendHeaderVariable(data = null, offset = 0, length = 2)</pre>	<pre>java.lang.NullPointerException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
6	offset<0 appendHeaderVariable(data= [A: B], offset= -1, length= 4)	java.lang.ArrayIndexOutOfBoundsException Exception is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
7	Length<0 appendHeaderVariable(data = [A: B], offset = 0, length = -1)	java.lang.ArrayIndexOutOfBoundsException Exception is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
8	valueOffset + valueLength > value.length appendHeaderVariable(data = [A: B], valueOffset= 1, length = 4)	java.lang.ArrayIndexOutOfBoundsException Exception is thrown. Catch the exception and build a page with status code = 204	HTTPResponse (statusCode = "204")
9	finalizeHeader called previously writeStatusCode(200) appendHeaderVariable(data = [A: B], valueOffset= 0, length = 4) finalizeHeader(); appendHeaderVariable(data = [C: D], valueOffset= 0, length = 4) appendContent ([OK], 0, 2); flush();	Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" HeaderNotPresent("C") contentBody= "OK")
10	appendContent called previously writeStatusCode(200) appendHeaderVariable(data = [A: B], valueOffset= 0, length = 4) appendContent ([OK], 0, 2); appendHeaderVariable(data = [C: D], valueOffset= 0, length = 4) flush();	Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" HeaderNotPresent("C") contentBody= "OK")
11	BUFFER_OVERFLOW writeStatusCode(200) appendHeaderVariable(data = [A: BbB...], valueOffset= 0, length = remainingBufferLength+1)	uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode = "200" Header("Content-Length")= "0" HeaderNotPresent("A"))

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
12	<p>flush called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(data = [A: B], valueOffset= 0, length = 4) appendContent ([OK], 0, 2); flush(); appendHeaderVariable(data = [C: D], valueOffset= 0, length = 4)</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length") = "2" HeaderNotPresent("C") contentBody= "OK")</pre>
13	<p>sendError called previously</p> <pre>sendError(500) appendHeaderVariable(data = [A: B], valueOffset= 0, length = 4)</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "500" HeaderNotPresent("A"))</pre>

6.1.5.3 Method appendHeaderVariable (byte[] name, short nameOffset, short nameLength, byte[] value, short valueOffset, short valueLength)

Test Area Reference: Api_1_Hrs_Ahva_Bss_Bss.

6.1.5.3.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void appendHeaderVariable(byte[] name,
    short nameOffset,
    short nameLength,
    byte[] value,
    short valueOffset,
    short valueLength)
    throws ScwsException
```

6.1.5.3.1.1 Normal execution

- CRRN1: Appends a header variable to outgoing. The colon and space (":") as well as the "CRLF" will be appended automatically.
- CRRN2: The header variables content-length or transfer-encoding are inserted by the framework depending on the transfer mode.

6.1.5.3.1.2 Parameter Errors

- CRRP1: if name is null, a java.lang.NullPointerException is thrown.
- CRRP2: if value is null, a java.lang.NullPointerException is thrown.
- CRRP3: if parameter nameOffset is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP4: if parameter nameLength is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP5: if parameter nameOffset plus nameLength are greater than the length of array name, a java.lang.ArrayIndexOutOfBoundsException is thrown.

- CRRP6: if parameter valueOffset is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP7: if parameter valueLength is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP8: if parameter valueOffset plus valueLength are greater than the length of array value, a java.lang.ArrayIndexOutOfBoundsException is thrown.

6.1.5.3.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown.
- CRRC2: if appending the header variable would cause an overflow of the response buffer, a uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown.
- CRRC3: if the flush method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.
- CRRC4: if the sendError method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.

6.1.5.3.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Ahva_Bss_Bss .java.
Test Applet: Api_1_Hrs_Ahva_Bss_Bss_1.java.
Cap File: Api_1_Hrs_Ahva_Bss_Bss .cap.

6.1.5.3.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5
N2	1, 2, 3, 4, 5
P1	6
P2	7
P3	8
P4	10
P5	12
P6	9
P7	11
P8	13
C1	14, 15
C2	16
C3	17
C4	18

6.1.5.3.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Basic</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value= [B], value Offset= 0, value Length= 1) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header ("A")= "B" Header ("Content-Length")= "2" contentType= "OK")</pre>
2	<p>nameOffset>0</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [XAa], nameOffset= 1, nameLength= 2, value= [B], value Offset= 0, value Length= 1) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header ("Aa")= "B" Header ("Content-Length")= "2" contentType= "OK")</pre>
3	<p>valueOffset>0</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value= [XBb], valueOffset= 1, valueLength= 2) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header ("A")= "Bb" Header ("Content-Length")= "2" contentType= "OK")</pre>
4	<p>nameLength = 0</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 0, value= [B], valueOffset= 0, valueLength= 1) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>	<p>When header name length is 0 then nothing is append to the responseBuffer.</p>	<pre>HTTPResponse (statusCode= "200" Header ("Content-Length")= "2" contentType= "OK" HeaderNotPresent ("A"))</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
5	<p>valueLength = 0</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 0) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" Header("A")= "" contentType= "OK")</pre>
6	<p>name[] is null</p> <pre>appendHeaderVariable(name= null, nameOffset= 0, nameLength= 1, value=[B], value Offset= 0, value Length= 1)</pre>	<pre>java.lang.NullPointerException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
7	<p>value[] is null</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value= null, value Offset= 0, value Length= 1)</pre>	<pre>java.lang.NullPointerException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
8	<p>nameOffset<0</p> <pre>appendHeaderVariable(name= [A], nameOffset= -1, nameLength= 1, value=[B], valueOffset= 0, valueLength= 1)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
9	<p>valueOffset<0</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= -1, valueLength= 1)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
10	<p>nameLength<0</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= -1, value=[B], valueOffset= 0, valueLength= 1)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
11	<p>valueLength<0</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= -1)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
12	<p>nameOffset + nameLength > name.length</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 2, value=[B], valueOffset= 0, valueLength= 1)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
13	<p>valueOffset + valueLength > value.length</p> <pre>appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 2)</pre>	<pre>java.lang.ArrayIndexOutOfBoundsException is thrown. Catch the exception and build a page with status code = 204</pre>	<pre>HTTPResponse (statusCode = "204")</pre>
14	<p>finalizeHeader called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 1) finalizeHeader(); appendHeaderVariable(name= [C], nameOffset= 0, nameLength= 1, value=[D], valueOffset= 0, valueLength= 1) appendContent ([OK], 0, 2); flush();</pre>	<pre>Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" HeaderNotPresent("C") contentType= "OK")</pre>
15	<p>appendContent called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 1) appendContent ([OK], 0, 2); appendHeaderVariable(name= [C], nameOffset= 0, nameLength= 1, value=[D], valueOffset= 0, valueLength= 1) flush();</pre>	<pre>Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" HeaderNotPresent("C") contentType= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
16	<p>BUFFER_OVERFLOW</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= (remainingBufferLength+1))</pre>	<pre>uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode = "200" Header("Content-Length")= "0" HeaderNotPresent("A"))</pre>
17	<p>flush called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 1) appendContent ([OK], 0, 2); flush(); appendHeaderVariable(name= [C], nameOffset= 0, nameLength= 1, value=[D], valueOffset= 0, valueLength= 1)</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("A")= "B" Header("Content-Length")= "2" HeaderNotPresent("C") contentBody= "OK")</pre>
18	<p>sendError called previously</p> <pre>sendError(500) appendHeaderVariable(name= [A], nameOffset= 0, nameLength= 1, value=[B], valueOffset= 0, valueLength= 1)</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "500" HeaderNotPresent("A"))</pre>

6.1.5.4 Method appendHeaderVariable (short headerKeywordNameId, byte[] value, short valueOffset, short valueLength)

This test case is FFS.

6.1.5.5 Method enableChunkMode

Test Area Reference: Api_1_Hrs_Encm.

6.1.5.5.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void enableChunkMode()
```

6.1.5.5.1.1 Normal execution

- CRRN1: Informs the SCWS to use the chunked mode for sending the response.
- CRRN2: Chunked mode off is the default mode.
- CRRN3: This allows the SCWS to handle large amounts of response data.

6.1.5.5.1.2 Parameter Errors

- n/a.

6.1.5.5.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a `uicc.scws.ScwsException` with reason codes `HEADER_ALREADY_FINALIZED` is thrown.
- CRRC2: if the flush method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.
- CRRC3: if the `sendError` method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.

6.1.5.5.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Encm.java.

Test Applet: Api_1_Hrs_Encm_1.java.

Cap File: Api_1_Hrs_Encm.cap.

6.1.5.5.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 11
N2	4
N3	5, 6
C1	7, 8
C2	9
C3	10

6.1.5.5.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>call in First</p> <pre>enableChunkMode() writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Transfer-Encoding")= "chunked" contentBody= "OK")</pre>
2	<p>write StatusCode call before</p> <pre>writeStatusCode(200) enableChunkMode() appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Transfer-Encoding")= "chunked" contentBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
3	<p>appendHeaderVariable call before</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) enableChunkMode() finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Transfer-Encoding")= "chunked" contentType= "OK")</pre>
4	<p>default mode off</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Content-Length: "2" HeaderNotPresent("Transfer- Encoding") contentType= "OK")</pre>
5	<p>large data several append</p> <pre>writeStatusCode(200) enableChunkMode() finalizeHeader(); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); appendContent ([aaa...], 0, 2048); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Transfer-Encoding")= "chunked" contentType= "aaa...")</pre>
6	<p>large data one append</p> <pre>writeStatusCode(200) enableChunkMode() finalizeHeader(); appendContent ([aaa...], 0, 10240); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Transfer-Encoding")= "chunked" contentType= "aaa...")</pre>
7	<p>header already finalized 01</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); enableChunkMode(); appendContent ([OK], 0, 2); flush();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" HeaderNotPresent("Transfer- Encoding") contentType= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
8	<p>header already finalized 02</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); enableChunkMode(); flush();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" HeaderNotPresent("Transfer- Encoding")) responseBody= "OK")</pre>
9	<p>flush called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); flush(); enableChunkMode();</pre>	<pre>} → Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" HeaderNotPresent("Transfer- Encoding")) responseBody= "OK")</pre>
10	<p>sendError called previously</p> <pre>sendError(500) enableChunkMode();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "500")</pre>
11	<p>Multiple success call</p> <pre>enableChunkMode() writeStatusCode(200) enableChunkMode() appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) enableChunkMode() enableChunkMode() finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Transfer-Encoding")= "chunked" responseBody= "OK")</pre>

6.1.5.6 Method finalizeHeader

Test Area Reference: Api_1_Hrs_Finh.

6.1.5.6.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void finalizeHeader()throws ScwsException
```

6.1.5.6.1.1 Normal execution

- CRRN1: Finalizes the header data in the outgoing buffer. It automatically adds missing header data with default values by the SCWS.

6.1.5.6.1.2 Parameter Errors

- n/a.

6.1.5.6.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown.
- CRRC2: if the adding of missing header data would cause access outside the Response buffer, a uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown.
- CRRC3: if the flush method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.
- CRRC4: if the sendError method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.

6.1.5.6.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Finh.java.

Test Applet: Api_1_Hrs_Finh_1.java.

Cap File: Api_1_Hrs_Finh.cap.

6.1.5.6.3 Test coverage

CRR number	Test case number
N1	1
C1	5, 6
C2	-
C3	7
C4	8

6.1.5.6.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Normal call</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" contentBody= "OK")</pre>
2	<p>header already finalized 01</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" contentBody= "OK")</pre>
3	<p>header already finalized 02</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) finalizeHeader(); appendContent ([OK], 0, 2); finalizeHeader(); flush();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" contentBody= "OK")</pre>
4	<p>flush called previously</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7)) appendContent ([OK], 0, 2); flush(); finalizeHeader();</pre>	<pre>} Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header1" Header("Content-Length")= "2" contentBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
5	sendError called previously <pre>sendError(500) finalizeHeader();</pre>	<pre>Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HttpResponse (statusCode= "500")</pre>

6.1.5.7 Method flush

Test Area Reference: Api_1_Hrs_Flus.

6.1.5.7.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void flush()
```

6.1.5.7.1.1 Normal execution

- CRRN1: Sends all data contained in the HTTP response.
- CRRN2: If this method is invoked in fixed buffer mode after a previous invocation of the same method the SCWS will do nothing.
- CRRN3: If this method is invoked in fixed buffer mode and the sendError() message was invoked previously the SCWS will do nothing.
- CRRN4: Invoking this method for the first time implicitly finalizes the header of the HttpResponse.

6.1.5.7.1.2 Parameter Errors

- n/a.

6.1.5.7.1.3 Context errors

- CRRC1: if the adding of missing header data would cause access outside the Response buffer, a uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown.

6.1.5.7.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Flus.java.

Test Applet: Api_1_Hrs_Flus_1.java.

Cap File: Api_1_Hrs_Flus.cap.

6.1.5.7.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 5
N2	2
N3	6
N4	4
C1	-

6.1.5.7.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Normal case + Chunk off</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent ([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" contentType= "OK")</pre>
2	<p>Repeat flush + chunk off</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent ([OK], 0, 2); flush(); flush(); flush(); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" contentType= "OK")</pre>
3	<p>Normal case + chunk on</p> <pre>writeStatusCode(200) enableChunkMode(); finalizeHeader(); appendContent ([Hello], 0, 5); flush(); appendContent ([], 0, 1); flush(); appendContent ([World], 0, 5); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Transfer-Encoding")= "chunk" contentType= "Hello World")</pre>
4	<p>Finalized the http header</p> <pre>writeStatusCode(200); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "0" contentType= "")</pre>
5	<p>No flush</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent ([OK], 0, 2);</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" contentType= "OK")</pre>
6	<p>SendError before</p> <pre>sendError(500); flush();</pre>		<pre>HTTPResponse (statusCode= "500")</pre>

6.1.5.8 Method getRemainingResponseBufferSize

Test Area Reference: Api_1_Hrs_Grrs.

6.1.5.8.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
int getRemainingResponseBufferSize()
```

6.1.5.8.1.1 Normal execution

- CRRN1: Returns the remaining size of the Response Buffer available for this applet.
- CRRN2: The size returned by this method is guaranteed until the next header data is added to the Response Buffer.
- CRRN3: The size returned by this method is guaranteed until the content data is added to the Response Buffer.
- CRRN4: Returns -1 if chunked mode is switched on.

6.1.5.8.1.2 Parameter Errors

- n/a.

6.1.5.8.1.3 Context errors

- CRRC1: if the flush method was called previously and the selected transfer mode is "Fixed transfer mode", a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.
- CRRC2: if the `sendError` method was called previously and the selected transfer mode is "Fixed transfer mode", a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.

6.1.5.8.2 Test area files

Specific triggering: None:

- Test Source: `Test_Api_1_Hrs_Grrs.java`.
- Test Applet: `Api_1_Hrs_Grrs_1.java`.
- Cap File: `Api_1_Hrs_Grrs.cap`.

6.1.5.8.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	3
N3	1
N4	2
C1	4
C2	5

6.1.5.8.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Remaining content data</p> <pre>writeStatusCode(200) finalizeHeader() remaining = getRemainingResponseBufferSize(); appendContent(data = [aaa...] offset = 0 length = remaining) flush()</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= remaining responseBody = "AaA[...]")</pre>
2	<p>Remaining + Chunk on</p> <pre>writeStatusCode(200) enableChunkMode(); finalizeHeader(); remaining = getRemainingResponseBufferSize(); appendContent([OK], 0, 2); flush();</pre>	<pre>remaining = -1</pre>	<pre>HTTPResponse (statusCode= "200" Header("Transfer-Encoding")= "chunked" responseBody= "OK")</pre>
3	<p>Remaining header data</p> <pre>writeStatusCode(200) remainingA = getRemainingResponseBufferSize(); appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 7, valueLength= 7) remainingB = getRemainingResponseBufferSize(); remainingC = getRemainingResponseBufferSize(); finalizeHeader(); appendContent([OK], 0, 2); flush();</pre>	<pre>remainingB < remainingA remainingB = remainingC</pre>	<pre>HTTPResponse (statusCode= "200" Header("Accept")= "header2" Header("Content-Length")= "2" responseBody= "OK")</pre>
4	<p>flush called previously</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent([OK], 0, 2); flush(); getRemainingResponseBufferSize();</pre>	<pre>Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" responseBody= "OK")</pre>
5	<p>sendError called previously</p> <pre>sendError(500) getRemainingResponseBufferSize();</pre>	<pre>Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "500")</pre>

6.1.5.9 Method reset

Test Area Reference: Api_1_Hrs_Rset.

6.1.5.9.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void reset() throws ScwsException
```

6.1.5.9.1.1 Normal execution

- CRRN1: Clears the Response Buffer content.
- CRRN2: Reset the state of the `HttpResponse` Object to its initial state.
- CRRN3: All the header variables and content that was appended to the `HttpResponse` until the invocation of the `reset` method is lost.

6.1.5.9.1.2 Parameter Errors

- n/a.

6.1.5.9.1.3 Context errors

- CRR1: if the flush method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.
- CRR2: if the `sendError` method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.

6.1.5.9.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Rset.java.

Test Applet: Api_1_Hrs_Rset_1.java.

Cap File: Api_1_Hrs_Rset.cap.

6.1.5.9.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	1, 2, 3
N3	1, 2, 3
C1	6
C2	7

6.1.5.9.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Reset StatusCode</p> <pre>writeStatusCode(204) reset(); writeStatusCode(200); finalizeHeader(); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" contentType = "OK")</pre>
2	<p>Reset Header</p> <pre>writeStatusCode(500); appendHeaderVariable(8, [L1], 0, 2); reset(); appendHeaderVariable(8, [L2], 0, 2); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Length")= "2" Header("Content-Language")= "L2" contentType = "OK")</pre>
3	<p>Reset ContentType</p> <pre>setContentTypes(4); reset(); writeStatusCode(200); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" Header("Content-Type") = "text/plain" (default value). contentType = "OK")</pre>
4	<p>Reset Header Finalized</p> <pre>writeStatusCode(200); appendHeaderVariable(8, [L1], 0, 2); finalizeHeader(); reset(); appendHeaderVariable(8, [L2], 0, 2); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" Header("Content-Language")= "L2" contentType = "OK")</pre>
5	<p>Reset Data</p> <pre>writeStatusCode(200); appendHeaderVariable(8, [L1], 0, 2); finalizeHeader(); appendContent([NOT OK], 0, 6); reset(); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" HeaderNotPresent("Content-Language") contentType = "OK")</pre>
6	<p>flush() called previously</p> <pre>writeStatusCode(200); finalizeHeader(); appendContent([OK], 0, 2); flush(); reset();</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= "200" Header("Content-Length")= "2" contentType = "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
7	sendError called previously sendError (500) ; reset () ;	} uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.	HTTPResponse (statusCode= "500")

6.1.5.10 Method sendError

Test Area Reference: Api_1_Hrs_Sene.

6.1.5.10.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void sendError(short errorCode)
```

6.1.5.10.1.1 Normal execution

- CRRN1: Sends an error code to the SCWS.
- CRRN2: The HTTP response is sent immediately by the SCWS no other data is send by the SCWS.
- CRRN3: Missing header variables are added by the SCWS with their default values.
- CRRN4: Invoking this method implicitly finalizes the header of the HttpResponseMessage.

6.1.5.10.1.2 Parameter Errors

- CRRP1: if parameter errorCode is unknown with respect to the HTTP 1.1 protocol, a uicc.scws.ScwsException with reason codes HTTP_CODE_UNKNOWN is thrown.

6.1.5.10.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown.
- CRRC2: if the flush method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.
- CRRC3: if the sendError method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.

6.1.5.10.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Sene.java.

Test Applet: Api_1_Hrs_Sene_1.java.

Cap File: Api_1_Hrs_Sene.cap.

6.1.5.10.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6
N2	1, 2, 3, 4, 5, 6
N3	1, 2, 3, 4, 5, 6
N4	1, 2, 3, 4, 5, 6
P1	7
C1	8, 9
C2	10
C3	11

6.1.5.10.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	Normal use case : Return an Informational code (1xx)		
a	// SC_CONTINUE sendError(100)		Not testable
b	// SC_SWITCHING_PROTOCOLS sendError(101)		Not testable
2	Normal use case : Return a Successful code (2xx)		
a	// SC_OK sendError(200)		HTTPResponse(statusCode= 200)
b	// SC_CREATED sendError(201)		HTTPResponse(statusCode= 201)
c	// SC_ACCEPTED sendError(202)		HTTPResponse(statusCode= 202)
d	// SC_NON_AUTHORITATIVE_INFORMATION sendError(203)		HTTPResponse(statusCode= 203)
e	// SC_NO_CONTENT sendError(204)		HTTPResponse(statusCode= 204)
f	// SC_RESET_CONTENT sendError(205)		HTTPResponse(statusCode= 205)
	// SC_PARTIAL_CONTENT sendError(206)		HTTPResponse(statusCode= 206)
3	Normal use case : Return a Redirection code (3xx)		
a	// SC_MULTIPLE_CHOICES sendError(300)		HTTPResponse(statusCode= 300)
b	// SC_MOVED_PERMANENTLY sendError(301)		HTTPResponse(statusCode= 301)
c	// SC_FOUND sendError(302)		HTTPResponse(statusCode= 302)
d	// SC_SEE_OTHER sendError(303)		HTTPResponse(statusCode= 303)
e	// SC_NOT_MODIFIED sendError(304)		HTTPResponse(statusCode= 304)
f	// SC_USE_PROXY sendError(305)		HTTPResponse(statusCode= 305)
g	// SC_TEMPORARY_REDIRECT sendError(307)		HTTPResponse(statusCode= 307)
4	Normal use case : Return a client error code (4xx)		
a	// SC_BAD_REQUEST sendError(400)		HTTPResponse(statusCode= 400)
b	// SC_UNAUTHORIZED sendError(401)		HTTPResponse(statusCode= 401)
c	// SC_PAYMENT_REQUIRED sendError(402)		HTTPResponse(statusCode= 402)
d	// SC_FORBIDDEN sendError(403)		HTTPResponse(statusCode= 403)
e	// SC_NOT_FOUND sendError(404)		HTTPResponse(statusCode= 404)

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
f	// SC_METHOD_NOT_ALLOWED sendError(405)		HTTPResponse(statusCode= 405)
g	// SC_NOT_ACCEPTABLE sendError(406)		HTTPResponse(statusCode= 406)
h	// SC_PROXY_AUTHENTICATION_REQUIRED sendError(407)		HTTPResponse(statusCode= 407)
i	// SC_REQUEST_TIMEOUT sendError(408)		HTTPResponse(statusCode= 408)
j	// SC_CONFLICT sendError(409)		HTTPResponse(statusCode= 409)
k	// SC_GONE sendError(410)		HTTPResponse(statusCode= 410)
l	// SC_LENGTH_REQUIRED sendError(411)		HTTPResponse(statusCode= 411)
m	// SC_PRECONDITION_FAILED sendError(412)		HTTPResponse(statusCode= 412)
n	// SC_REQUEST_ENTITY_TOO_LARGE sendError(413)		HTTPResponse(statusCode= 413)
o	// SC_REQUEST_URI_TOO_LONG sendError(414)		HTTPResponse(statusCode= 414)
p	// SC_UNSUPPORTED_MEDIA_TYPE sendError(415)		HTTPResponse(statusCode= 415)
q	// SC_REQUESTED_RANGE_NOT_SATISFIABLE sendError(416)		HTTPResponse(statusCode= 416)
r	// SC_EXPECTATION_FAILED sendError(417)		HTTPResponse(statusCode= 417)
5	Normal use case : Return a Server Error 5xx code		
a	// SC_INTERNAL_SERVER_ERROR sendError(500)		HTTPResponse(statusCode= 500)
b	// SC_NOT_IMPLEMENTED sendError(501)		HTTPResponse(statusCode= 501)
c	// SC_BAD_GATEWAY sendError(502)		HTTPResponse(statusCode= 502)
d	// SC_SERVICE_UNAVAILABLE sendError(503)		HTTPResponse(statusCode= 503)
e	// SC_GATEWAY_TIMEOUT sendError(504)		HTTPResponse(statusCode= 504)
f	// SC_HTTP_VERSION_NOT_SUPPORTED sendError(505)		HTTPResponse(statusCode= 505)
6	After writeStatusCode writeStatusCode(200); sendError(403);		HTTPResponse(statusCode= 403)
7	HTTP_CODE_UNKNOWN sendError(0) sendError(-1) sendError(1000) sendError(99) sendError(102) sendError(199) sendError(207)	uicc.scws.ScwsException with reason codes HTTP_CODE_UNKNOWN is thrown. Catch the exception and build a page with status code = 204	HTTPResponse(statusCode = "204")

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
	<pre> sendError(299) sendError(308) sendError(399) sendError(418) sendError(499) sendError(506) sendError(600) sendError(700) sendError(800) sendError(900) </pre>		
8	<p>finalizeHeader called previously</p> <pre> writeStatusCode(200) finalizeHeader(); sendError(403); appendContent([OK], 0, 2); flush(); </pre>	<pre> } Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure. </pre>	<pre> HTTPResponse (statusCode= 200 Header("Content-Length")= "2" contentBody= "OK") </pre>
9	<p>appendContent called previously</p> <pre> writeStatusCode(200) appendContent([OK], 0, 2); sendError(403); flush(); </pre>	<pre> } Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure. </pre>	<pre> HTTPResponse (statusCode= 200 Header("Content-Length")= "2" contentBody= "OK") </pre>
10	<p>flush called previously</p> <pre> writeStatusCode(200) appendContent([OK], 0, 2); flush(); sendError(403); </pre>	<pre> } uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure. </pre>	<pre> HTTPResponse (statusCode= 200 Header("Content-Length")= "2" contentBody= "OK") </pre>
11	<p>sendError called previously</p> <pre> sendError(500); sendError(403); </pre>	<pre> } uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure. </pre>	<pre> HTTPResponse (statusCode= 500) </pre>

6.1.5.11 Method setContentType

Test Area Reference: Api_1_Hrs_Scot.

6.1.5.11.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void setContentType(short type) throws ScwsException
```

6.1.5.11.1.1 Normal execution

- CRRN1: Sets the content type of the HTTP response.

6.1.5.11.1.2 Parameter Errors

- CRRP1: if parameter type is not correct according to SCWSConstants, a uicc.scws.ScwsException with reason codes UNKNOWN_KEYWORD_ID is thrown.

6.1.5.11.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown.
- CRRC2: if appending the header variable would cause an overflow of the response buffer, a uicc.scws.ScwsException with reason codes BUFFER_OVERFLOW is thrown.
- CRRC3: if the flush method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.
- CRRC4: if the sendError method was called previously, a uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown.

6.1.5.11.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Scot.java.

Test Applet: Api_1_Hrs_Scot_1.java.

Cap File: Api_1_Hrs_Scot.cap.

6.1.5.11.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5
P1	6
C1	7, 8
C2	-
C3	9
C4	10

6.1.5.11.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	All success type		
a	<pre>// CONTENT_TYPE_IMAGE_GIF writeStatusCode(200) setContentType(4); finalizeHeader(); appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse (statusCode= "200" Header("Content-Type")= "image/gif" Header("Content-Length")= "2" responseBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
b	<pre>// CONTENT_TYPE_IMAGE_JPEG writeStatusCode(200) setContentType(3); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "image/jpeg" Header("Content-Length")= "2" responseBody= "OK")</pre>
c	<pre>// CONTENT_TYPE_IMAGE_PNG writeStatusCode(200) setContentType(5); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "image/png" Header("Content-Length")= "2" responseBody= "OK")</pre>
d	<pre>// CONTENT_TYPE_TEXT_HTML writeStatusCode(200) setContentType(1); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/html" Header("Content-Length")= "2" responseBody= "OK")</pre>
e	<pre>// CONTENT_TYPE_TEXT_PLAIN writeStatusCode(200) setContentType(2); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>
2	Call in first <pre>setContentType(2); writeStatusCode(200) finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>
3	Call after status line <pre>writeStatusCode(200) setContentType(2); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>
4	Call after appendHeaderVariable <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 1, value= [header1header2], valueOffset= 0, valueLength= 7) setContentType(2); finalizeHeader(); appendContent([OK],0,2); flush();</pre>		<pre>HTTPResponse (statusCode= 200 Header("Accept")= "header1" Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>
5	Multiple call <pre>writeStatusCode(200) setContentType(5); setContentType(4); setContentType(3); setContentType(2); setContentType(1); finalizeHeader(); appendContent([OK],0,2);</pre>		<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/html" Header("Content-Length")= "2" responseBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
	flush();		
6	<p>UNKNOWN_KEYWORD_ID</p> <pre>writeStatusCode(200) setContentType(0); setContentType(-1); setContentType(6); setContentType(-2); setContentType(2); finalizeHeader(); appendContent([OK],0,2); flush();</pre>	<pre>}uicc.scws.ScwsException with reason codes UNKNOWN_KEYWORD_ID is thrown. }uicc.scws.ScwsException with reason codes UNKNOWN_KEYWORD_ID is thrown. }uicc.scws.ScwsException with reason codes UNKNOWN_KEYWORD_ID is thrown. }uicc.scws.ScwsException with reason codes UNKNOWN_KEYWORD_ID is thrown. Catch all exceptions and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>
7	<p>finalizeHeader called previously</p> <pre>writeStatusCode(200) finalizeHeader(); setContentType(2); appendContent([OK],0,2); flush();</pre>	<pre>}Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= 200 Header("Content-Length")= "2" responseBody= "OK")</pre>
8	<p>appendContent called previously</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent([OK],0,2); setContentType(2); flush();</pre>	<pre>}Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= 200 Header("Content-Length")= "2" responseBody= "OK")</pre>
9	<p>flush called previously</p> <pre>writeStatusCode(200) setContentType(2); appendContent([OK],0,2); flush(); setContentType(1);</pre>	<pre>}uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= 200 Header("Content-Type")= "text/plain" Header("Content-Length")= "2" responseBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
10	sendError called previously <pre>sendError (500) setContentType (2);</pre>	<pre>uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</pre>	<pre>HTTPResponse (statusCode= 500)</pre>

6.1.5.12 Method writeStatusCode

Test Area Reference: Api_1_Hrs_Wstc.

6.1.5.12.1 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
void writeStatusCode(short code)
    throws ScwsException
```

6.1.5.12.1.1 Normal execution

- CRRN1: Write an HTTP status code into the outgoing buffer.

6.1.5.12.1.2 Parameter Errors

- CRRP1: if parameter `code` is unknown with respect to the HTTP 1.1 protocol, a `uicc.scws.ScwsException` with reason codes `HTTP_CODE_UNKNOWN` is thrown.

6.1.5.12.1.3 Context errors

- CRRC1: if the HTTP header was already finalized, a `uicc.scws.ScwsException` with reason codes `HEADER_ALREADY_FINALIZED` is thrown.
- CRRC2: if the `writeStatusCode` has already been invoked, a `uicc.scws.ScwsException` with reason codes `STATUS_LINE_ALREADY_SET` is thrown.
- CRRC3: if the `appendHeaderVariable` has already been invoked, a `uicc.scws.ScwsException` with reason codes `STATUS_LINE_ALREADY_SET` is thrown.
- CRRC4: if the `flush` method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.
- CRRC5: if the `sendError` method was called previously, a `uicc.scws.ScwsException` with reason codes `HTTP_RESPONSE_ALREADY_SENT` is thrown.

6.1.5.12.2 Test area files

Specific triggering: None:

Test Source: Test_Api_1_Hrs_Wstc.java.

Test Applet: Api_1_Hrs_Wstc_1.java.

Cap File: Api_1_Hrs_Wstc.cap.

6.1.5.12.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 12
P1	6
C1	7
C2	8
C3	9
C4	10
C5	11

6.1.5.12.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	Normal use case : Return a basic page with an Informational status code (1xx)		
a	// SC_CONTINUE writeStatusCode(100)		Not testable
b	// SC_SWITCHING_PROTOCOLS writeStatusCode(101)		Not testable
2	Normal use case : Return a basic page with a Successful status code (2xx)		
a	// SC_OK writeStatusCode(200) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 200 Header("Content-Length")= 2 contentType= "OK")
b	// SC_CREATED writeStatusCode(201) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 201 Header("Content-Length")= 2 contentType= "OK")
c	// SC_ACCEPTED writeStatusCode(202) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 202 Header("Content-Length")= 2 contentType= "OK")
d	// SC_NON_AUTHORITATIVE_INFORMATION writeStatusCode(203) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 203 Header("Content-Length")= 2 contentType= "OK")
e	// SC_NO_CONTENT writeStatusCode(204)		HTTPResponse (statusCode= 204 Header("Content-Length")= null // not present in the response.)
f	// SC_RESET_CONTENT writeStatusCode(205) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 205 Header("Content-Length")= 2 contentType= "OK")
g	// SC_PARTIAL_CONTENT writeStatusCode(206) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 206 Header("Content-Length")= 2 contentType= "OK")
3	Normal use case : Return a basic page with a Redirection status code (3xx)		
a	// SC_MULTIPLE_CHOICES writeStatusCode(300) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 300 Header("Content-Length")= 2 contentType= "OK")
b	// SC_MOVED_PERMANENTLY writeStatusCode(301) appendContent([OK], 0, 2); flush();		HTTPResponse (statusCode= 301 Header("Content-Length")= 2 contentType= "OK")

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
c	// SC_FOUND writeStatusCode(302) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 302 Header("Content-Length")= 2 contentBody= "OK")
d	// SC_SEE_OTHER writeStatusCode(303) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 303 Header("Content-Length")= 2 contentBody= "OK")
e	// SC_NOT_MODIFIED writeStatusCode(304)		HTTPResponse(statusCode= 304)
f	// SC_USE_PROXY writeStatusCode(305) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 305 Header("Content-Length")= 2 contentBody= "OK")
g	// SC_TEMPORARY_REDIRECT writeStatusCode(307) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 307 Header("Content-Length")= 2 contentBody= "OK")
4	Normal use case : Return a basic page with a Client error status code (4xx)		
a	// SC_BAD_REQUEST writeStatusCode(400) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 400 Header("Content-Length")= 2 contentBody= "OK")
b	// SC_UNAUTHORIZED writeStatusCode(401) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 401 Header("Content-Length")= 2 contentBody= "OK")
c	// SC_PAYMENT_REQUIRED writeStatusCode(402) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 402 Header("Content-Length")= 2 contentBody= "OK")
d	// SC_FORBIDDEN writeStatusCode(403) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 403 Header("Content-Length")= 2 contentBody= "OK")
e	// SC_NOT_FOUND writeStatusCode(404) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 404 Header("Content-Length")= 2 contentBody= "OK")
f	// SC_METHOD_NOT_ALLOWED writeStatusCode(405) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 405 Header("Content-Length")= 2 contentBody= "OK")
g	// SC_NOT_ACCEPTABLE writeStatusCode(406) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 406 Header("Content-Length")= 2 contentBody= "OK")
h	// SC_PROXY_AUTHENTICATION_REQUIRED writeStatusCode(407) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 407 Header("Content-Length")= 2 contentBody= "OK")
i	// SC_REQUEST_TIMEOUT writeStatusCode(408) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 408 Header("Content-Length")= 2 contentBody= "OK")

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
j	// SC_CONFLICT writeStatusCode(409) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 409 Header("Content-Length")= 2 contentType= "OK")
k	// SC_GONE writeStatusCode(410) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 410 Header("Content-Length")= 2 contentType= "OK")
l	// SC_LENGTH_REQUIRED writeStatusCode(411) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 411 Header("Content-Length")= 2 contentType= "OK")
m	// SC_PRECONDITION_FAILED writeStatusCode(412) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 412 Header("Content-Length")= 2 contentType= "OK")
n	// SC_REQUEST_ENTITY_TOO_LARGE writeStatusCode(413) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 413 Header("Content-Length")= 2 contentType= "OK")
o	// SC_REQUEST_URI_TOO_LONG writeStatusCode(414) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 414 Header("Content-Length")= 2 contentType= "OK")
p	// SC_UNSUPPORTED_MEDIA_TYPE writeStatusCode(415) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 415 Header("Content-Length")= 2 contentType= "OK")
q	// SC_REQUESTED_RANGE_NOT_SATISFIABLE writeStatusCode(416) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 416 Header("Content-Length")= 2 contentType= "OK")
r	// SC_EXPECTATION_FAILED writeStatusCode(417) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 417 Header("Content-Length")= 2 contentType= "OK")
5	Normal use case : Return a basic page with a Server error status code (5xx)		
a	// SC_INTERNAL_SERVER_ERROR writeStatusCode(500) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 500 Header("Content-Length")= 2 contentType= "OK")
b	// SC_NOT_IMPLEMENTED writeStatusCode(501) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 501 Header("Content-Length")= 2 contentType= "OK")
c	// SC_BAD_GATEWAY writeStatusCode(502) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 502 Header("Content-Length")= 2 contentType= "OK")
d	// SC_SERVICE_UNAVAILABLE writeStatusCode(503) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 503 Header("Content-Length")= 2 contentType= "OK")
e	// SC_GATEWAY_TIMEOUT writeStatusCode(504) appendContent([OK], 0, 2); flush();		HTTPResponse(statusCode= 504 Header("Content-Length")= 2 contentType= "OK")

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
f	<pre>// SC_HTTP_VERSION_NOT_SUPPORTED writeStatusCode(505) appendContent([OK], 0, 2); flush();</pre>		<pre>HTTPResponse(statusCode= 505 Header("Content-Length")= 2 contentBody= "OK")</pre>
6	<p>HTTP_CODE_UNKNOWN</p> <pre>writeStatusCode(0) writeStatusCode(-1) writeStatusCode(1000) writeStatusCode(99) writeStatusCode(102) writeStatusCode(199) writeStatusCode(207) writeStatusCode(299) writeStatusCode(308) writeStatusCode(399) writeStatusCode(418) writeStatusCode(499) writeStatusCode(506) writeStatusCode(600) writeStatusCode(700) writeStatusCode(800) writeStatusCode(900)</pre>	<p>uicc.scws.ScwsException with reason codes HTTP_CODE_UNKNOWN is thrown. Catch ALL exceptions and build a page with status code = 204</p>	<pre>HTTPResponse(statusCode = "204")</pre>
7	<p>header already finalized</p> <pre>writeStatusCode(200) finalizeHeader(); writeStatusCode(500) appendContent([OK], 0, 2); flush();</pre>	<p>Uicc.scws.ScwsException with reason codes HEADER_ALREADY_FINALIZED is thrown. Catch the exception and continue the procedure.</p>	<pre>HTTPResponse(statusCode= 200 Header("Content-Length")= 2 contentBody= "OK")</pre>
8	<p>writeStatusCode already invoked</p> <pre>writeStatusCode(200) writeStatusCode(500) finalizeHeader(); appendContent([OK], 0, 2); flush();</pre>	<p>Uicc.scws.ScwsException with reason codes STATUS_LINE_ALREADY_SET is thrown. Catch the exception and continue the procedure.</p>	<pre>HTTPResponse(statusCode= 200 Header("Content-Length")= 2 contentBody= "OK")</pre>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
9	<p>appendHeaderVariable already invoked</p> <pre>writeStatusCode(200) appendHeaderVariable(headerKeywordNameId= 10, value= [test], valueOffset= 0, valueLength= 4) writeStatusCode(500) finalizeHeader(); appendContent([OK], 0, 2); flush();</pre>	<p>Uicc.scws.ScwsException with reason codes STATUS_LINE_ALREADY_SET is thrown. Catch the exception and continue the procedure.</p>	<pre>HTTPResponse(statusCode= 200 Content-Length: 2 Content-Type: test responseBody= "OK")</pre>
10	<p>flush called previously</p> <pre>writeStatusCode(200) finalizeHeader(); appendContent([OK], 0, 2); flush(); writeStatusCode(500)</pre>	<p>Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</p>	<pre>HTTPResponse(statusCode= 200 Content-Length: 2 responseBody= "OK")</pre>
11	<p>sendError called previously</p> <pre>sendError(500) writeStatusCode(200)</pre>	<p>Uicc.scws.ScwsException with reason codes HTTP_RESPONSE_ALREADY_SENT is thrown. Catch the exception and continue the procedure.</p>	<pre>HTTPResponse(statusCode= 500)</pre>
12	<p>writeStatusCode not called</p> <pre>appendContent([OK], 0, 2); flush();</pre>	<p>Default status code is 200 .</p>	<pre>HTTPResponse(statusCode= 200 Content-Length: 2 responseBody= "OK")</pre>

6.2 SCWS Runtime Environment

6.2.1 Applet state

6.2.1.1 Invokation of applets not in state selectable

Test Area Reference: Api_2_Aps_Ivns.

6.2.1.1.1 Conformance Requirement

The SCWS Framework implementation shall be compliant to the following statement (TS 102 588 [2], clause 4.3):

- Only an Applet that is in selectable state can be invoked by the SCWS.

6.2.1.1.1.1 Normal execution

- CRRN1 : Only an applet in selectable state can be invoked by the SCWS.
- CRRN2 : If the applet is in non selectable state, then the SCWS does not invoke this one and returns an HTTP response with a error status code = SC_NOT_FOUND whatever the HTTP method.

6.2.1.1.1.2 Parameter Errors

- n/a.

6.2.1.1.1.3 Context errors

- n/a.

6.2.1.1.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_2_Aps_Ivns.java.

Test Applet: Api_2_Aps_Ivns_1.java.

Cap File: Api_2_Aps_Ivns.cap.

6.2.1.1.3 Test coverage

CRR number	Test case number
N1, N2	1
N1, N2	2
N1, N2	3
N1, N2	4
N1, N2	5
N1, N2	6
N1, N2	7

6.2.1.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods: POST</p> <ol style="list-style-type: none"> 1. Send POST HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send POST HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send POST HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doPost()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
2	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods: GET</p> <ol style="list-style-type: none"> 1. Send GET HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send GET HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send GET HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doGet()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>
3	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods: DELETE</p> <ol style="list-style-type: none"> 1. Send DELETE HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send DELETE HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send DELETE HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doDelete()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>
4	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods : HEAD</p> <ol style="list-style-type: none"> 1. Send HEAD HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send HEAD HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send HEAD HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doHead()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
5	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods : PUT</p> <ol style="list-style-type: none"> 1. Send PUT HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send PUT HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send PUT HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doPut()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>
6	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods : OPTIONS</p> <ol style="list-style-type: none"> 1. Send OPTIONS HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send OPTIONS HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send OPTIONS HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doOptions()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>
7	<p>Invoke an applet LOCKED / UNLOCKED with HTTP methods : TRACE</p> <ol style="list-style-type: none"> 1. Send TRACE HTTP REQUEST (1) to <i>Api_2_Aps_Ivns_1</i> 2. Lock the applet <i>Api_2_Aps_Ivns_1</i> (non selectable state) 3. Send TRACE HTTP REQUEST (2) to <i>Api_2_Aps_Ivns_1</i> 4. Unlock the applet <i>Api_2_Aps_Ivns_1</i> (selectable state) 5. Send TRACE HTTP REQUEST (3) to <i>Api_2_Aps_Ivns_1</i> <p><u>Applet source code in doTrace()</u></p> <pre>response.writeStatusCode(ScwsConstants.SC_NO_CONTENT);</pre>		<p>Check the HTTP</p> <p>RESPONSE (1): HTTPResponse (statusCode = "204")</p> <p>RESPONSE (2): HTTPResponse (statusCode = "404")</p> <p>RESPONSE (3): HTTPResponse (statusCode = "204")</p>

6.2.1.2 Registration remains valid if applet is not in selectable state

This test case is FFS.

6.2.2 Response sending

All tests regarding sending of a response are implicitly covered by the test cases in clause 6.1.5.

6.2.3 Exception handling

6.2.3.1 No exception shall be propagated as HTTP error to the terminal

Test Area Reference: Api_2_Exh_Noex.

6.2.3.1.1 Conformance Requirement

The SCWS Framework implementation shall be compliant to the following statement (TS 102 588 [2], clause 4.4):

Exceptions thrown by the invoked Applet shall not be propagated to the terminal, and the SCWS shall send an error status code according to HTTP 1.1[1]. In case of "chunked mode", some response data could have already been sent by the SCWS. In this case, no error shall be sent by the SCWS. In case of "fixed buffer size mode", the status code is always sent by the SCWS.

6.2.3.1.1.1 Normal execution

- CRRN1: If an exceptions thrown by the invoked Applet the SCWS send an error status code according to HTTP 1.1[1].
- CRRN2: In case of "chunked mode", if an exceptions thrown by the invoked Applet the SCWS and if some response data could have already been sent by the SCWS. Then, no error shall be sent by the SCWS.
- CRRN3: In case of "fixed buffer size mode", if an exceptions thrown by the invoked Applet the SCWS. Then, the status code is always sent by the SCWS.

6.2.3.1.1.2 Parameter Errors

- n/a.

6.2.3.1.1.3 Context errors

- n/a.

6.2.3.1.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_2_Exh_Noex.java.

Test Applet: Api_2_Exh_Noex_1.java.

Cap File: Api_2_Exh_Noex.cap.

6.2.3.1.3 Test coverage

CRR number	Test case number
N1	1
N2	2
N3	3

6.2.3.1.4 Test procedure

Test Case			
Id	Description	API/SCWS Framework Expectation	APDU/HTTP Expectation
1	<u>Applet throws an exception</u> Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")
2	<u>In "chunked mode", applet throws an Exception AFTER writeStatusCode()</u> enableChunkMode() writeStatusCode(200) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")
3	<u>In "chunked mode", applet throws an Exception AFTER appendHeaderVariable()</u> enableChunkMode() writeStatusCode(200) appendHeaderVariable("a: B", 0, 4) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")
4	<u>In "chunked mode", applet throws an Exception AFTER appendContent()</u> enableChunkMode() writeStatusCode(200) appendHeaderVariable("A: B", 0, 4) appendContent("OK", 0, 2) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "200" Header = "A: B" Content = "OK")
5	<u>In "fixed buffer size mode", applet throws an Exception AFTER writeStatusCode()</u> writeStatusCode(200) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")
6	<u>In "fixed buffer size mode", applet throws an Exception AFTER appendHeaderVariable_ ()</u> writeStatusCode(200) appendHeaderVariable("a: B", 0, 4) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")
7	<u>In "fixed buffer size mode", applet throws an Exception AFTER appendContent()</u> writeStatusCode(200) appendHeaderVariable("A: B", 0, 4) appendContent("OK", 0, 2) Byte[] array = new byte[2] Array[3] = 0x00;	java.lang. ArrayIndexOutOfBoundsException is thrown	HTTPResponse (statusCode = "500")

Annex A (normative): Class, methods and SCWSFramework tests acronyms

A.1 Smart Card Web Server part

ScwsExtensionRegistry	Ser
ScwsExtension	Set
ScwsExtensionService	Ses
HttpRequest	Hrq
HttpResponse	Hrs

A.1.1 Class ScwsExtensionRegistry

Method name	Acronyms
static void deregister()	Dreg
static void register()	Regi

A.1.2 Class ScwsExtensionService

Method Name	Acronyms
void doDelete()	Ddel
void doGet()	Dget
void doHead()	Dhea
void doOption()	Dopt
void doPost()	Dpos
void doPut()	Dput
void doTrace()	Dtra

A.1.3 HttpRequest interface

Method Name	Acronyms
short findAndCopyKeywordValue(byte [] headerKeywordName, short nameOffs, short nameLength, byte[] buffer, short bufferOffs, short maxLength)	Fckw_Bss_Bss
short findAndCopyKeywordValue(short keywordId, byte[] buffer, short bufferOffs, short maxLength)	Fckws_Bss
int getContentLength()	Gcle
short getContentType()	Gcty
short getRequestHttpVersion()	Grhv
short readContent()	Rcon
short readContentType()	Rcty

A.1.4 HttpResponse interface

Method Name	Acronyms
void appendContent()	Acon
void appendHeaderVariable(byte[] data, short offset, short length)	Ahva_Bss
void appendHeaderVariable(byte[] name, short nameOffset, short nameLength, byte[] value, short valueOffset, short valueLength)	Ahva_Bss_Bss
void appendHeaderVariable(short headerKeywordNameId, byte[] value, short valueOffset, short valueLength)	Ahvas_Bss

Method Name	Acronyms
void enableChunkMode()	Encm
void finalizeHeader()	Finh
void flush()	Flus
int getRemainingResponseBufferSize()	Grrs
void reset()	Rset
void sendError()	Sene
void setContentType()	Scot
void writeStatusCode()	Wstc

A.2 Acronyms for SCWS Framework tests

Method Name	Acronyms
Applet state	Aps
Exception handling	Exh

A.2.1 Applet state

Test Area within the chapter	Acronyms
Invocation of applets not in state selectable	Ivns
Registration remains valid if applet is not in selectable state	Regv

A.2.3 Exception handling

Method Name	Acronyms
No exception shall be propagated as HTTP error to the terminal	Noex

Annex B (normative): Test file description

Every test source is written in JAVA™ and shall use methods defined in Annex C interfaces to communicate with the card.

In order to be more readable, data specified as method string parameters shall be presented in 4 blocks of 4 bytes per line. Every block is separated by a space character. Every string line is appended to previous one and shall be aligned. An example is provided in annex C.

Every test file shall start with a call to reset() method.

Annex C (normative): uicc.scws.test.util package and interfaces

See attached files:

- Annex_C_ScwsTestUtil.zip.
- Annex_C_ScwsInterfaces.zip.

NOTE: These files are contained in archive ts_102835v070000p0.zip which accompanies the present document.

Annex D (normative): Test Area files

See attached file:

- Annex_D_SourceCode.zip.

NOTE: This file is contained in archive ts_102835v070000p0.zip which accompanies the present document.

Annex E (informative): HTTP-Request and HTTP-Response handling

The HttpClient V3.1 [i.2] from the Apache Software Foundation is used to handle HTTP requests and HTTP responses.

It is permissible to use another framework for HTTP handling, in this case the interface *ScwsApplicationManagementService* has to be modified accordingly.

Annex F (informative): Bibliography

- IETF RFC 791: "Internet Protocol".

NOTE: Available at <http://www.ietf.org/rfc/rfc791.txt>.

- IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

NOTE: Available at <http://www.ietf.org/rfc/rfc2616.txt>.

- IETF RFC 793: "Transmission Control Protocol".

NOTE: Available at <http://www.ietf.org/rfc/rfc793.txt>.

Annex G (informative): Change history

The table below indicates all changes that have been incorporated into the present document since it was placed under change control.

Change history								
Date	Meeting	Plenary Doc	RQ	Rev	Cat	Subject/Comment	Old	New
						Creation of the specification		7.0.0

History

Document history		
V7.0.0	April 2010	Publication