3GPP TSG-T (Terminals) Meeting #17
Biarritz, France   4 – 6 September 2002

Tdoc TP-020223

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Terminals; Test Specification for 'C'-language binding to (U)SIM API (Release 6)

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Keywords

GSM, SIM, API

3GPP

Postal address

3GPP support office address

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

Internet

http://www.3gpp.org

Contents

Contents ... 3

Foreword ... 11

1 Scope ... 12

2 References ... 12

3 Definitions and Acronyms ... 13
3.1 Definitions ... 13
3.2 Acronyms ... 13

4 Test Environment ... 14
4.1 Applicability ... 14
4.2 Test Environment Description ... 14
4.3 Test Format ... 15
4.3.1 Test Area Reference ... 15
4.3.1.1 Conformance Requirements ... 15
4.3.1.2 Test Area Files ... 16
4.3.1.3 Test Procedure ... 16
4.3.1.4 Test Coverage ... 16
4.4 Initial Conditions ... 16
4.5 Test Equipment ... 16
4.5.1 APDU Tool ... 16
4.6 Testing Methodology ... 17
4.6.1 Test Interfaces and Facilities ... 17

5 Test Plan ... 17

6 API Test Plan ... 17
6.1 UICC File Store Access ... 17
6.1.1 CatSelect ... 17
6.1.1.1 Conformance Requirements ... 18
6.1.1.2 Test Suite Files ... 18
6.1.1.3 Test Procedure ... 19
6.1.1.4 Test Coverage ... 20
6.1.2 CatStatus ... 20
6.1.2.1 Conformance Requirements ... 20
6.1.2.2 Test Suite Files ... 20
6.1.2.3 Test Procedure ... 21
6.1.2.4 Test Coverage ... 21
6.1.3 CatGetCHVStatus ... 21
6.1.3.1 Conformance Requirements ... 21
6.1.3.2 Test Suite Files ... 21
6.1.3.3 Test Procedure ... 21
6.1.3.4 Test Coverage ... 21
6.1.4 CatReadBinary ... 22
6.1.4.1 Conformance Requirements ... 22
6.1.4.2 Test Suite Files ... 22
6.1.4.3 Test Procedure ... 23
6.1.4.4 Test Coverage ... 23
6.1.5 CatUpdateBinary ... 23
6.1.5.1 Conformance Requirements ... 24
6.1.5.2 Test Suite Files ... 24
6.1.5.3 Test Procedure ... 24
6.1.5.4 Test Coverage ... 25
6.1.6 CatReadRecord ... 25
6.1.6.1 Conformance Requirements ... 25
6.1.6.2 Test Suite Files ... 26
6.1.6.3 Test Procedure ... 27
6.1.6.4 Test Coverage ... 29

6.1.7 CatUpdateRecord ... 29
6.1.7.1 Conformance Requirements ... 29
6.1.7.2 Test Suite Files ... 30
6.1.7.3 Test Procedure ... 31
6.1.7.4 Test Coverage ... 33
6.1.8 CatSearch ... 33
6.1.8.1 Conformance Requirements ... 33
6.1.8.2 Test Suite Files ... 34
6.1.8.3 Test Procedure ... 34
6.1.8.4 Test Coverage ... 35
6.1.9 CatIncrease ... 35
6.1.9.1 Conformance Requirements ... 36
6.1.9.2 Test Suite Files ... 36
6.1.9.3 Test Procedure ... 36
6.1.9.4 Test Coverage ... 37
6.1.10 CatInvalidate ... 37
6.1.10.1 Conformance Requirements ... 37
6.1.10.2 Test Suite Files ... 38
6.1.10.3 Test Procedure ... 38
6.1.10.4 Test Coverage ... 38
6.1.11 CatRehabilitate ... 38
6.1.11.1 Conformance Requirements ... 38
6.1.11.2 Test Suite Files ... 39
6.1.11.3 Test Procedure ... 39
6.1.11.4 Test Coverage ... 39
6.2 Registry ... 40
6.2.1 CatSetMenuString ... 40
6.2.1.1 Conformance Requirements ... 40
6.2.1.2 Test Suite Files ... 40
6.2.1.3 Test Procedure ... 40
6.2.1.4 Test Coverage ... 40
6.2.2 CatNotifyOnFrameworkEvent ... 40
6.2.2.1 Conformance Requirements ... 41
6.2.2.2 Test Suite Files ... 41
6.2.2.3 Test Procedure ... 41
6.2.2.4 Test Coverage ... 41
6.2.3 CatNotifyOnEnvelope ... 41
6.2.3.1 Conformance Requirements ... 41
6.2.3.2 Test Suite Files ... 42
6.2.3.3 Test Procedure ... 42
6.2.3.4 Test Coverage ... 42
6.2.4 CatNotifyOnEvent ... 42
6.2.4.1 Conformance Requirements ... 42
6.2.4.2 Test Suite Files ... 42
6.2.4.3 Test Procedure ... 43
6.2.4.4 Test Coverage ... 43
6.3 Man-Machine Interface ... 43
6.3.1 CatAddItem ... 43
6.3.1.1 Conformance Requirements ... 43
6.3.1.2 Test Suite Files ... 43
6.3.1.3 Test Procedure ... 43
6.3.1.4 Test Coverage ... 44
6.3.2 CatSelectItem ... 44
6.3.2.1 Conformance Requirements ... 44
6.3.2.2 Test Suite Files ... 44
6.3.2.3 Test Procedure ... 44
6.3.2.4 Test Coverage ... 44
6.3.3 CatEndSelectItem ... 44
6.3.3.1 Conformance Requirements ... 45
6.3.3.2 Test Suite Files ... 45
6.3.3.3 Test Procedure ... 45
6.3.3.4 Test Coverage ... 45

6.3.4 CatDisplayText ... 46
6.3.4.1 Conformance Requirements ... 46
6.3.4.2 Test Suite Files ... 46
6.3.4.3 Test Procedure ... 46
6.3.4.4 Test Coverage ... 46
6.3.5 CatGetInKey ... 47
6.3.5.1 Conformance Requirements ... 47
6.3.5.2 Test Suite Files ... 47
6.3.5.3 Test Procedure ... 47
6.3.5.4 Test Coverage ... 48
6.3.6 CatGetInput ... 48
6.3.6.1 Conformance Requirements ... 48
6.3.6.2 Test Suite Files ... 48
6.3.6.3 Test Procedure ... 49
6.3.6.4 Test Coverage ... 49
6.3.7 CatSetupIdleModeText ... 49
6.3.7.1 Conformance Requirements ... 50
6.3.7.2 Test Suite Files ... 50
6.3.7.3 Test Procedure ... 50
6.3.7.4 Test Coverage ... 50
6.3.8 CatPlayTone ... 50
6.3.8.1 Conformance Requirements ... 51
6.3.8.2 Test Suite Files ... 51
6.3.8.3 Test Procedure ... 51
6.3.8.4 Test Coverage ... 51
6.4 Timers ... 51
6.4.1 CatGetTimer ... 51
6.4.1.1 Conformance Requirements ... 52
6.4.1.2 Test Suite Files ... 52
6.4.1.3 Test Procedure ... 52
6.4.1.4 Test Coverage ... 52
6.4.2 CatFreeTimer ... 52
6.4.2.1 Conformance Requirements ... 52
6.4.2.2 Test Suite Files ... 52
6.4.2.3 Test Procedure ... 53
6.4.2.4 Test Coverage ... 53
6.4.3 CatStartTimer ... 53
6.4.3.1 Conformance Requirements ... 53
6.4.3.2 Test Suite Files ... 53
6.4.3.3 Test Procedure ... 53
6.4.3.4 Test Coverage ... 54
6.4.4 CatGetTimerValue ... 54
6.4.4.1 Conformance Requirements ... 54
6.4.4.2 Test Suite Files ... 54
6.4.4.3 Test Procedure ... 54
6.4.4.4 Test Coverage ... 54
6.5 Supplementary Card Reader Management ... 55
6.5.1 CatPowerOnCard ... 55
6.5.1.1 Conformance Requirements ... 55
6.5.1.2 Test Suite Files ... 55
6.5.1.3 Test Procedure ... 55
6.5.1.4 Test Coverage ... 55
6.5.2 CatPowerOffCard ... 56
6.5.2.1 Conformance Requirements ... 56
6.5.2.2 Test Suite Files ... 56
6.5.2.3 Test Procedure ... 56
6.5.2.4 Test Coverage ... 56
6.5.3 CatPerformCardAPDU ... 56
6.5.3.1 Conformance Requirements ... 56
6.5.3.2 Test Suite Files ... 57
6.5.3.3 Test Procedure ... 57
6.5.3.4 Test Coverage ... 57

6.5.4 CatGetReaderStatus 57
6.5.4.1 Conformance Requirements 57
6.5.4.2 Test Suite Files 58
6.5.4.3 Test Procedure 58
6.5.4.4 Test Coverage 58
6.6 Network Services 58
6.6.1 CatGetLocationInformation 58
6.6.1.1 Conformance Requirements 58
6.6.1.2 Test Suite Files 59
6.6.1.3 Test Procedure 59
6.6.1.4 Test Coverage 59
6.6.2 CatGetTimingAdvance 59
6.6.2.1 Conformance Requirements 59
6.6.2.2 Test Suite Files 60
6.6.2.3 Test Procedure 60
6.6.2.4 Test Coverage 60
6.6.3 CatGetIMEI 60
6.6.3.1 Conformance Requirements 60
6.6.3.2 Test Suite Files 60
6.6.3.3 Test Procedure 61
6.6.3.4 Test Coverage 61
6.6.4 CatGetNetworkMeasurementResults 61
6.6.4.1 Conformance Requirements 61
6.6.4.2 Test Suite Files 61
6.6.4.3 Test Procedure 61
6.6.4.4 Test Coverage 62
6.6.5 CatGetDateTimeAndTimeZone 62
6.6.5.1 Conformance Requirements 62
6.6.5.2 Test Suite Files 62
6.6.5.3 Test Procedure 62
6.6.5.4 Test Coverage 62
6.6.6 CatGetLanguage 63
6.6.6.1 Conformance Requirements 63
6.6.6.2 Test Suite Files 63
6.6.6.3 Test Procedure 63
6.6.6.4 Test Coverage 63
6.6.7 CatSetupCall 63
6.6.7.1 Conformance Requirements 64
6.6.7.2 Test Suite Files 64
6.6.7.3 Test Procedure 64
6.6.7.4 Test Coverage 64
6.6.8 CatSendShortMessage 65
6.6.8.1 Conformance Requirements 65
6.6.8.2 Test Suite Files 65
6.6.8.3 Test Procedure 65
6.6.8.4 Test Coverage 66
6.6.9 CatSendSS 66
6.6.9.1 Conformance Requirements 66
6.6.9.2 Test Suite Files 66
6.6.9.3 Test Procedure 66
6.6.9.4 Test Coverage 67
6.6.10 CatSendUSSD 67
6.6.10.1 Conformance Requirements 67
6.6.10.2 Test Suite Files 67
6.6.10.3 Test Procedure 67
6.6.10.4 Test Coverage 68
6.6.11 CatOpenCSChannel 68
6.6.11.1 Conformance Requirements 68
6.6.11.2 Test Suite Files 69
6.6.11.3 Test Procedure 69
6.6.11.4 Test Coverage 69
6.6.12 CatOpenGPRSChannel 69

6.6.12.1 Conformance Requirements 69
6.6.12.2 Test Suite Files 70
6.6.12.3 Test Procedure 70
6.6.12.4 Test Coverage 70
6.6.13 CatCloseChannel 70
6.6.13.1 Conformance Requirements 70
6.6.13.2 Test Suite Files 70
6.6.13.3 Test Procedure 71
6.6.13.4 Test Coverage 71
6.6.14 CatReceiveData 71
6.6.14.1 Conformance Requirements 71
6.6.14.2 Test Suite Files 71
6.6.14.3 Test Procedure 72
6.6.14.4 Test Coverage 72
6.6.15 CatSendData 73
6.6.15.1 Conformance Requirements 73
6.6.15.2 Test Suite Files 73
6.6.15.3 Test Procedure 73
6.6.15.4 Test Coverage 74
6.6.16 CatGetChannelStatus 74
6.6.16.1 Conformance Requirements 74
6.6.16.2 Test Suite Files 74
6.6.16.3 Test Procedure 75
6.6.16.4 Test Coverage 75
6.6.17 CatServiceSearch 75
6.6.17.1 Conformance Requirements 75
6.6.17.2 Test Suite Files 75
6.6.17.3 Test Procedure 76
6.6.17.4 Test Coverage 76
6.6.18 CatGetServiceInformation 76
6.6.18.1 Conformance Requirements 76
6.6.18.2 Test Suite Files 76
6.6.18.3 Test Procedure 77
6.6.18.4 Test Coverage 77
6.6.19 CatDeclareService 77
6.6.19.1 Conformance Requirements 77
6.6.19.2 Test Suite Files 77
6.6.19.3 Test Procedure 78
6.6.19.4 Test Coverage 78
6.6.20 CatRunATCommand 78
6.6.20.1 Conformance Requirements 78
6.6.20.2 Test Suite Files 78
6.6.20.3 Test Procedure 79
6.6.20.4 Test Coverage 79
6.6.21 CatSendDTMFCommad 79
6.6.21.1 Conformance Requirements 79
6.6.21.2 Test Suite Files 80
6.6.21.3 Test Procedure 80
6.6.21.4 Test Coverage 80
6.7 Toolkit Application 80
6.7.1 main 80
6.7.1.1 Conformance Requirements 80
6.7.1.2 Test Suite Files 80
6.7.1.3 Test Procedure 81
6.7.1.4 Test Coverage 81
6.7.2 CatGetFrameworkEvent 81
6.7.2.1 Conformance Requirements 81
6.7.2.2 Test Suite Files 81
6.7.2.3 Test Procedure 81
6.7.2.4 Test Coverage 81
6.7.3 CatExit 82
6.7.3.1 Conformance Requirements 82

6.7.3.2 Test Suite Files ... 82
6.7.3.3 Test Procedure ... 82
6.7.3.4 Test Coverage ... 82
6.8 Miscellaneous ... 82
6.8.1 CatGetTerminalProfile ... 82
6.8.1.1 Conformance Requirements ... 82
6.8.1.2 Test Suite Files ... 83
6.8.1.3 Test Procedure ... 83
6.8.1.4 Test Coverage ... 83
6.8.2 CatMoreTime ... 83
6.8.2.1 Conformance Requirements ... 83
6.8.2.2 Test Suite Files ... 84
6.8.2.3 Test Procedure ... 84
6.8.2.4 Test Coverage ... 84
6.8.3 CatPollingOff ... 84
6.8.3.1 Conformance Requirements ... 84
6.8.3.2 Test Suite Files ... 84
6.8.3.3 Test Procedure ... 85
6.8.3.4 Test Coverage ... 85
6.8.4 CatPollInterval ... 85
6.8.4.1 Conformance Requirements ... 85
6.8.4.2 Test Suite Files ... 85
6.8.4.3 Test Procedure ... 85
6.8.4.4 Test Coverage ... 86
6.8.5 CatRefresh ... 86
6.8.5.1 Conformance Requirements ... 86
6.8.5.2 Test Suite Files ... 86
6.8.5.3 Test Procedure ... 86
6.8.5.4 Test Coverage ... 87
6.8.6 CatLanguageNotification ... 87
6.8.6.1 Conformance Requirements ... 87
6.8.6.2 Test Suite Files ... 87
6.8.6.3 Test Procedure ... 87
6.8.6.4 Test Coverage ... 87
6.8.7 CatLaunchBrowser ... 88
6.8.7.1 Conformance Requirements ... 88
6.8.7.2 Test Suite Files ... 88
6.8.7.3 Test Procedure ... 88
6.8.7.4 Test Coverage ... 88
6.9 Low-Level Interface ... 89
6.9.1 CatResetBuffer ... 89
6.9.1.1 Conformance Requirements ... 89
6.9.1.2 Test Suite Files ... 89
6.9.1.3 Test Procedure ... 89
6.9.1.4 Test Coverage ... 89
6.9.2 CatStartProactiveCommand ... 89
6.9.2.1 Conformance Requirements ... 89
6.9.2.2 Test Suite Files ... 90
6.9.2.3 Test Procedure ... 90
6.9.2.4 Test Coverage ... 90
6.9.3 CatSendProactiveCommand ... 90
6.9.3.1 Conformance Requirements ... 90
6.9.3.2 Test Suite Files ... 90
6.9.3.3 Test Procedure ... 91
6.9.3.4 Test Coverage ... 91
6.9.4 CatOpenEnvelope ... 91
6.9.4.1 Conformance Requirements ... 91
6.9.4.2 Test Suite Files ... 91
6.9.4.3 Test Procedure ... 91
6.9.4.4 Test Coverage ... 92
6.9.5 CatSendEnvelopeResponse ... 92
6.9.5.1 Conformance Requirements ... 92

6.9.5.2 Test Suite Files 92
6.9.5.3 Test Procedure 92
6.9.5.4 Test Coverage 92
6.9.6 CatSendEnvelopeErrorResponse 92
6.9.6.1 Conformance Requirements........ 92
6.9.6.2 Test Suite Files 93
6.9.6.3 Test Procedure 93
6.9.6.4 Test Coverage 93
6.9.7 CatPutData........ 93
6.9.7.1 Conformance Requirements........ 93
6.9.7.2 Test Suite Files 93
6.9.7.3 Test Procedure 94
6.9.7.4 Test Coverage 94
6.9.8 CatPutByte........ 94
6.9.8.1 Conformance Requirements........ 94
6.9.8.2 Test Suite Files 94
6.9.8.3 Test Procedure 94
6.9.8.4 Test Coverage 95
6.9.9 CatPutTLV........ 95
6.9.9.1 Conformance Requirements........ 95
6.9.9.2 Test Suite Files 95
6.9.9.3 Test Procedure 95
6.9.9.4 Test Coverage 95
6.9.10 CatPutBytePrefixedTLV........ 96
6.9.10.1 Conformance Requirements 96
6.9.10.2 Test Suite Files........ 96
6.9.10.3 Test Procedure........ 96
6.9.10.4 Test Coverage........ 96
6.9.11 CatPutOneByteTLV........ 96
6.9.11.1 Conformance Requirements 96
6.9.11.2 Test Suite Files........ 97
6.9.11.3 Test Procedure........ 97
6.9.11.4 Test Coverage........ 97
6.9.12 CatPutTwoByteTLV 97
6.9.12.1 Conformance Requirements 97
6.9.12.2 Test Suite Files........ 97
6.9.12.3 Test Procedure........ 98
6.9.12.4 Test Coverage........ 98
6.9.13 CatGetByte 98
6.9.13.1 Conformance Requirements 98
6.9.13.2 Test Suite Files........ 98
6.9.13.3 Test Procedure........ 98
6.9.13.4 Test Coverage........ 98
6.9.14 CatGetData 99
6.9.14.1 Conformance Requirements 99
6.9.14.2 Test Suite Files........ 99
6.9.14.3 Test Procedure........ 99
6.9.14.4 Test Coverage........ 99
6.9.15 CatFindNthTLV........ 99
6.9.15.1 Conformance Requirements 99
6.9.15.2 Test Suite Files........ 100
6.9.15.3 Test Procedure........ 100
6.9.15.4 Test Coverage........ 100
6.9.16 CatFindNthTLVInUserBuffer........ 100
6.9.16.1 Conformance Requirements 100
6.9.16.2 Test Suite Files........ 101
6.9.16.3 Test Procedure........ 101
6.9.16.4 Test Coverage........ 101

Annex A Script file syntax and format description (normative) 102
A.1 Syntax description 102
A.2 Semantics 102
A.3 Example 103
A.4 Style and formatting 103

Annex B Default Prepersonalisation (normative) 105
B.1 General Default Prepersonalisation 105
B.2 File System Access Default Prepersonalisation 106
B.2.1 $DF_{SIMTEST}$ (SIM Test) 106
B.2.2 EF_{TNR} (Transparent Never Read) 106
B.2.3 EF_{TNU} (Transparent Never Update) 106
B.2.4 EF_{TARU} (Transparent Always Read and Update) 106
B.2.5 EF_{CNR} (Cyclic Never Read) 107
B.2.6 EF_{CNU} (Cyclic Never Update) 107
B.2.7 EF_{CNIC} (Cyclic Never Increase) 107
B.2.8 EF_{CNIV} (Cyclic Never Invalidate) 108
B.2.9 EF_{CNRH} (Cyclic Never Rehabilitate) 108
B.2.10 EF_{CARU} (Cyclic Always Read and Update) 108
B.2.11 EF_{LNR} (Linear Fixed Never Read) 109
B.2.12 EF_{LNU} (Linear Fixed Never Update) 109
B.2.13 EF_{LARU} (Linear Fixed Always Read and Update) 109
B.2.14 $EF_{\mathbf{CINA}}$ (Cyclic Increase Not Allowed) 110
B.2.15 $EF_{\mathbf{TRAC}}$ (Transparent Read Access Condition CHV2) 110
B.2.16 $EF_{\mathbf{TIAC}}$ (Transparent Invalidate Access Condition CHV1) 110
B.2.17 $EF_{\mathbf{CIAC}}$ (Cyclic Increase Access Condition CHV2) 111
B.2.18 $EF_{\mathbf{CIAA}}$ (Cyclic Increase Access Condition ADM) 111
B.2.19 $EF_{\mathbf{CNRI}}$ (Cyclic Never Rehabilitate Invalidated) 111

Annex C (informative): Change history 111

Foreword

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

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document covers the minimum characteristics considered necessary in order to provide compliance to 3GPP 31.131 "'C'-language binding to (U)SIM API" [11].

The present document describes the technical characteristics and methods of test for testing the SIM API for the C programming language [11] implemented in the subscriber identity modules for GSM and 3G networks. 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 may be used.

2 References

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

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

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 31.111: "3rd Generation Partnership Project; Technical Specification Group; USIM Application Toolkit (USAT)".

[3] 3GPP TS 23.048: "3rd Generation Partnership Project; Technical Specification Group Terminals; Security Mechanisms for the SIM application toolkit".

[4] 3GPP TS 42.019: "3rd Generation Partnership Project; Technical Specification Group Terminals; Subscriber Identity Module Application Programming Interface (SIM API); Stage 1".

[5] ISO 639 (1988): "Code for the representation of names of languages".

[6] 3GPP TS 23.038: "Alphabets and language-specific information".

[7] ISO/IEC 9899 Second Edition 1999-12-01: "Programming Languages -- C".

[8] 3GPP TS 11.14: "Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface" version 4.0.0 Release 4.

[9] Tool Interface Standard (TIS) Executable and Linking Format Specification Version 1.2

[10] SYSTEM V Application Binary Interface, Edition 4.1

[11] 3GPP TS 31.131 V1.1.2: "'C'-language binding to (U)SIM API".

[12] GSM 11.10-1: "Digital cellular telecommunication system (Phase 2+); Mobile Station (MS) conformance specification; Part 1: Conformance specification".

[13] 3GPP TS 51.011: "Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) interface".

[14] ETSI TS 102.221: "UICC-Terminal interface; Physical and logical characteristics".

[15] ETSI TS 102.226: "Remote APDU Structure for UICC based Applications."

3 Definitions and Acronyms

3.1 Definitions

The definitions specified in GSM 11.10-1 [12] shall apply, unless otherwise specified in the present clause.

Application: A computer program that defines and implements a useful domain-specific functionality. The term may apply to the functionality itself, to the representation of the functionality in a programming language, or to the realization of the functionality as executable code.

Application Executable: The representation of an application as collection of executable codes.

Application Program: The representation of an application in a programming language such as assembly language, C, Java, WML or XHTML.

Application Programming Interface: A collection of entry points and data structures that an application program can access when translated into an application executable.

Byte Code: A processor-independent representation of a basic computer operation such as "increment by one" that is executed by computer program called a byte code interpreter.

Data Structure: A memory address that can be accessed by an application executable in order to read or write data.

Entry Point: A memory address that can be branched to by an application executable in order to access functionality defined by an application-programming interface. Depending on the software technology, an entry point is also called a subroutine, a function or a method.

Executable Code: The generic term for either byte code or native code.

Framework : A framework defines a set of Application Programming Interface (API) functions for developing applications and for providing system services to those applications.

Native Code: A processor-dependent representation of a basic computer operation such as "increment by one" that is executed by the hardware circuitry of a computer's central processing unit.

Null Operation: A computer operation that accomplishes nothing. Abbreviated as NOP and pronounced "No Op".

Toolkit Application: An application that uses the commands described in [2].

3.2 Acronyms

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

AC	Application Code
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
CAD	Card Acceptance Device
CPDU	Command Protocol Data Unit

CAT	Card Application Toolkit
DF	Dedicated File
DTMF	Dual Tone Multiple Frequency
EF	Elementary File
FID	File Identifier
GSM	Global System for Mobile communications
IFD	Interface Device
NOP	Null OPeration
ME	Mobile Equipment
NVM	Non-Volatile Memory
ROM	Read-Only Memory
SE	Sending Entity
SIM	Subscriber Identity Module
SMS	Short Message Service
STK	SIM ToolKit
TLV	Tag, Length, Value
TPDU	Transport Protocol Data Unit
UICC	(not an acronym)
URL	Uniform Resource Locator
USIM	Universal Subscri ber Interface Module

4 Test Environment

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

4.1 Applicability

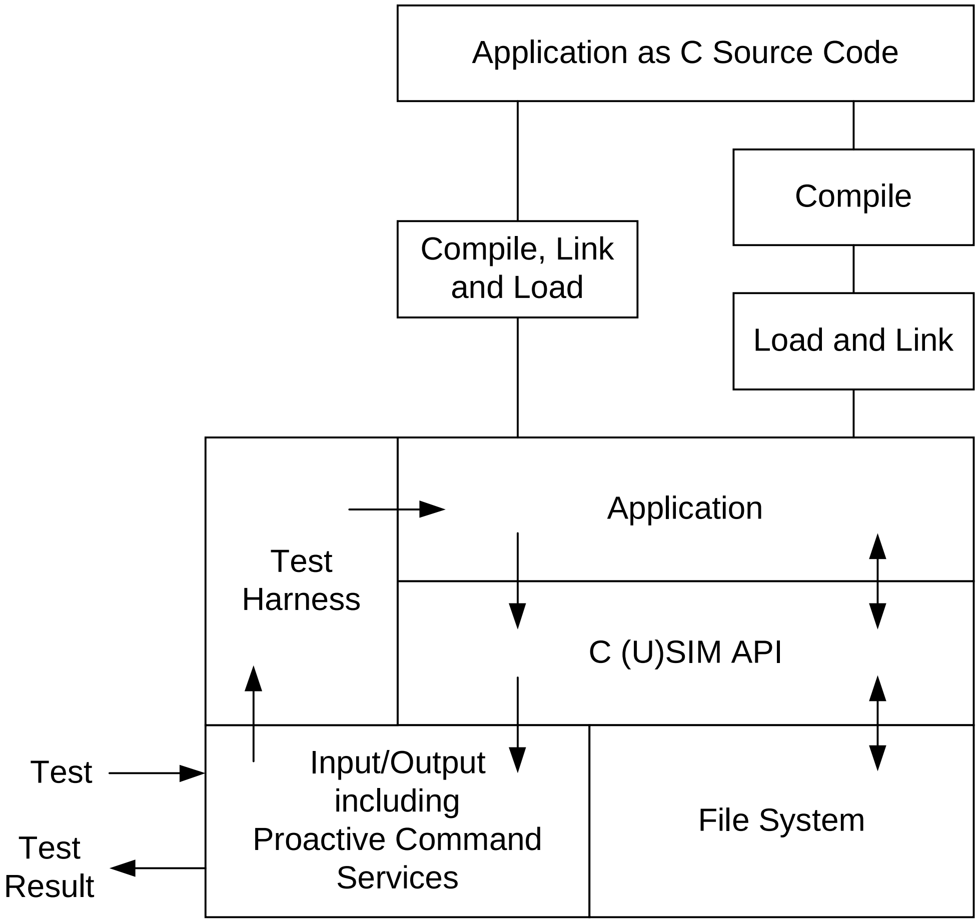
The tests defined in this specification shall be performed taking into account the services supported by the card as specified in the EF_{SST} file.

The tests can be performed with either C source code files (.c) or with ELF loadfiles (.elf).

4.2 Test Environment Description

The C SIM API test specification covers applications that are linked into the smart card mask and stored in read-only memory as well as applications that are loaded into read/write memory after it has been manufactured. The C SIM API test specification also covers applications whose executable form is either as byte codes or as native codes.

The general architecture for the test environment is given in the following diagram:



4.3 Test Format

4.3.1 Test Area Reference

Each test area is referenced as follows:

API: API Testing: 'API_[entry point name]' where

entry point name:

API entry point defined in 3GPP TS 31.131

4.3.1.1 Conformance Requirements

The conformance requirements are expressed in the following way:

- Entry point prototype as listed in 3GPP TS 31.131 [11] specification.
- 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 entry point is used in, each referenced as a Conformance Requirement Reference Context Error (CRRC)

4.3.1.2 Test Area Files

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

- Test Script: [Test Area Reference]_[Test Runtime]_[Test Number].tst
- Test Application: [Test Area Reference]_[Test Number].c

The test numbers start from '1'. If there is only one test application/script pair for an entry point, the test number is not used.

The test scripts use the syntax in Annex A. For readability purposes, the test script may be passed through a C pre-processor [7] in order to be rendered in the Annex A syntax.

If the test application is not already resident in the test environment, the test script loads it using the application loading and installation procedures of [15]. In this case, the loadfile in the test script may be specific to a particular runtime environment such as Java Card™ or Multos™ or a particular processor architecture such as a Fujitsu FR or an Infineon SLE66 so this runtime dependency is included in the name of the test script.

Test scripts for each area are run sequentially in the order of the test number. It is possible that the test with test number i+1 assumes the state of the card left by a successful execution of the test with test number i.

4.3.1.3 Test Procedure

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

Test Case			
Id	**Description**	**API Expectation**	**APDU Expectation**
	Test Case detailed description	*API expected behavior.*	*Expected response at APDU level.*

4.3.1.4 Test Coverage

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

4.4 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 this document, the following rules apply to the Initial Conditions:

- unless otherwise stated, the file system and the files' content shall fulfill the requirements described in the "Default Prepersonalisation" paragraph at the time of running the test with test number 1;
- unless otherwise stated, tests with test number 1 through i have been successfully executed before the test with test number i+1 is executed.

When both statements apply, a test procedure is said to be in the "Default Initial Conditions" state.

4.5 Test Equipment

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

4.5.1 APDU Tool

This test tool shall meet the following requirements:

- be able to send commands to the card TPDU;
- be able to check none, only a part, or all of the data returned;
- be able to check none, only part, or all of the status returned;
- be able to accept all valid status codes returned;
- be able to support CAD commands;
- be able to generate a log file for each test execution.
- if more data is returned than defined in the test specification the event shall be noted in the log and the tool shall continue;
- if less data is returned than defined in the test specification the event shall be noted in the log and the tool shall continue;
- if there is an error in data or status returned the event shall be noted in the log and the tool shall continue.

The log file produced by the test tool shall include the following information:

- all commands issued;
- all data returned;
- all status returned;
- all errors codes;
- expected data and status in case of error;
- comments from the scripts;
- a log message to report success or failure of the test;
- transmission error events including anomalies in returned data;

4.6 Testing Methodology

4.6.1 Test Interfaces and Facilities

The SIM-ME interface provides the main transport interface for the purpose of performing conformance tests.

The SIM API interface provides the main test interface for the purpose of performing conformance tests.

5 Test Plan

The test plan is divided according to the entry point sections of 3GPP TS 31.131 although the sections of the test plan do not appear in the same order as the corresponding sections in TS 31.131.

6 API Test Plan

6.1 UICC File Store Access

6.1.1 CatSelect

Test Area Reference: API_CatSelect

6.1.1.1 Conformance Requirements

The entry point with the following signature shall be compliant to its definition in the API.

```
UINT16 CatSelect(CatFID FileIdentifier, CatFileStatus *Status)
```

6.1.1.1.1 Normal execution

CRRN1: If the desired file is selected, the proper status information has been returned in *Status.

CRRN2: After selecting a DF/MF no EF is selected.

CRRN3: After selecting a linear fixed EF no record is selected.

CRRN4: After selecting a cyclic EF the first record which is the last updated record is selected.

CRRN5: The file with a file identifier that matches fid shall be found according to the following selection rules:

1) An immediate child EF or DF of the current MF/DF can be selected,
2) A sibling DF of the current DF can be selected,
3) The current MF/DF it self can be selected,
4) The parent MF/DF of the current DF can be selected,
5) The MF can always be selected.

CRRN6: If status is NULL the invocation is a NOP.

6.1.1.1.2 Parameter errors

6.1.1.1.3 Context errors

CRRC1: If the file with a file identitifier which matches FileIdentitifer could not be found according to the selection rules listed in CRRN5 then the FILE_NOT_FOUND status word shall be returned.

CRRC2: If the entry point call causes a memory problem (e.g. memory access error), the MEMORY_PROBLEM status word shall be returned.

CRRC3: If the entry point call causes an error to occur that is not expected and thus not handled, the INTERNAL_ERROR status word shall be returned.

6.1.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatSelect.tst
Test Application:	API_CatSelect.c

6.1.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Select EFICCID in MF (Transparent EF)** `CatSelect(FID_EF_ICCID, &status);`	Return value: ‘9000’	
2	**Select EF$_{ICCID}$ in MF (Transparent EF)** `CatSelect(FID_EF_ICCID, &status);`	Return value: ‘9000’	
3	**Select DF$_{GSM}$ in MF** `CatSelect(FID_DF_GSM, &status);`	Return value: ‘9000’	
3	**Select DF$_{GSM}$ in MF** `CatSelect(FID_DF_GSM, &status)`	Return value: ‘9000’	
4	**Select EF$_{ACM}$ in DF$_{GSM}$ (CyclicEF)** `CatSelect(FID_EF_ACM, &status)`	Return value: ‘9000’	
5	**Select MF** `CatSelect(FID_MF, &status)`	Return value: ‘9000’	
6	**Select DF$_{TELECOM}$ in MF** `CatSelect(FID_DF_TELECOM, &status)`	Return value: ‘9000’	
7	**Select EF$_{FDN}$ in DF$_{TELECOM}$ (Linear FixedEF)** `CatSelect(FID_EF_FDN, &status)`	Return value: ‘9000’	
8	**Status is null** `CatSelect(FID_EF_FDN, NULL)`	Return value: ‘9000’	

13	**Selection possibilities**
	`1 - CatSelect(FID_MF, &status)`
	`2 – CatSelect(FID_DF_TELECOM, &status)`
	`3 - CatSelect(FID_DF_GRAPHICS, &status)`
	`4 - CatSelect(FID_DF_TELECOM, &status)`
	`5 - CatSelect(FID_DF_GRAPHICS, &status)`
	`6 - CatSelect(FID_MF, &status)`
	`7 - CatSelect(FID_DF_GSM, &status)`
	`8 - CatSelect(FID_DF_TELECOM, &status)`
	`9 - CatSelect(FID_DF_TELECOM, &status)`
		1 – Return value: ‘9000’
		2 – Return value: ‘9000’
		3 – Return value: ‘9000’
		4 – Return value: ‘9000’.
		5 – Return value: ‘9000’
		6 – Return value: ‘9000’
		7 – Return value: ‘9000’
		8 – Return value: ‘9000’
		9 – Return value: ‘9000’	

14	**EF not selected after MF/DF selection**
	`1 - CatSelect(FID_MF, &status)`
	`CatSelect(FID_EF_ICCID, &status)`
	`2 - CatSelect(FID_MF)`
	`CatReadBinary()`
		1 - Return value: ‘9000’.
		2 – Return value: NO_EF_SELECTED	

15	**No selection of non-reachable file**
	`1 - CatSelect(FID_MF, &status)`
	`2 - CatSelect(FID_EF_ACM, &status)`
		1 – Return value: ‘9000’
		2 – Return value: FILE_NOT_FOUND.	

16	**No record is selected after selecting linear fixed EF**
	`1 - CatSelect(FID_MF, &status)`
	`2 - CatSelect(FID_DF_SIMTEST, &status)`
	`3 - CatSelect(FID_EF_LARU, &status)`
	`4 – CatReadRecord(0, CURRENT)`
		1 – Return value: ‘9000’.
		2 – Return value: ‘9000’.
		3 – Return value: ‘9000’.
		4 – Return value: RECORD_ _NOT_FOUND	

Id	Description	API Expectation	APDU Expectation
17	**Record pointer in selected cyclic EF** 1 – CatSelect(FID_MF, &status) 2 - CatSelect(FID_DF_SIMTEST, &status) 3 - CatSelect(FID_EF_CARU, &status) 4 – unsigned char data1[] = {1,2,3}, data2[3]; CatUpdateRecord(data1, PREVIOUS) 5 - CatSelect(FID_EF_CARU, &status) CatReadRecord(data2) compare data1 to data2	1 - Return value: ‘9000’. 2 - Return value: ‘9000’. 3 - Return value: ‘9000’. 4 - Return value: ‘9000’. 5 - The contents of data1 and data2 shall be identical.	

6.1.1.4 Test Coverage

CRR Number	Test Case Number
N1	2, 3
N2	4
N3	5
N4	6
N5	7
N6	8
C1	13
C2, C3	Not Tested

6.1.2 CatStatus

Test Area Reference: API_CatStatus

6.1.2.1 Conformance Requirements

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

```
UINT16 CatStatus(CatFileStatus *Status)
```

6.1.2.1.1 Normal execution

CRRN1: The status information of the current EF is returned.

CRRN2: If Status is NULL the invocation is a NOP.

6.1.2.1.2 Parameter errors

6.1.2.1.3 Context errors

6.1.2.2 Test Suite Files

Additional requirements for the U(SIM) personalisation:

Test Script:	API_ CatStatus.tst
Test Application:	API_ CatStatus.c

6.1.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	UICC **Initialisation**	Responses ignored.	

1	**Status of MF**
	`1 – CatSelect(FID_MF, &status1)`
	`2 - CatSelect(&status2)`
		1 – Return value ‘9000’
		2 – Return value ‘9000’
		status1 shall equal status2	

2	**Status after select EF$_{ICCID}$ in MF**
	`1 – CatSelect(FID_DF_GSM, &status1)`
	`2 - CatSelect(&status2)`
		1 – Return value ‘9000’
		2 – Return value ‘9000’
		status1 shall equal status2	

3	**Status of DF$_{Telecom}$**
	`1 – CatSelect(FID_DF_TELECOM, &status1)`
	`2 - CatSelect(&status2)`
		1 – Return value ‘9000’
		2 – Return value ‘9000’
		status1 shall equal status2	

6.1.2.4 Test Coverage

CRR Number	Test Case Number
N1	1, 2, 3

6.1.3 CatGetCHVStatus

Test Area Reference: API_CatGetCHVStatus

6.1.3.1 Conformance Requirements

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

```
UINT16 CatGetCHVStatus(BYTE *CHV)
```

6.1.3.1.1 Normal execution

CRRN1: The current CHV status information is returned.

6.1.3.1.2 Parameter errors

6.1.3.1.3 Context errors

6.1.3.2 Test Suite Files

Additional requirements for the UICC personalisation:

Test Script: API_ CatGetCHVStatus.tst

Test Application: API_ CatGetCHVStatus.c

6.1.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Get Current CHV Status**
	`BYTE CHV[4];`
	`1- CatGetCHVStatus(CHV);`
		1 - Return value: ‘9000’
		The correct CHV status information is returned in CHV.	

6.1.3.4 Test Coverage

CRR Number	Test Case Number
N1	1

6.1.4 CatReadBinary

Test Area Reference: API_CatReadBinary

6.1.4.1 Conformance Requirements

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

```
UINT16 CatReadBinary(DWORD Offset, DWORD *Bytes, void *Buffer);
```

6.1.4.1.1 Normal execution

CRRN1: If data can be accessed at the specified offset in the currently selected transparent file, an attempt is made to read the number of bytes in Bytes. The bytes read are returned in Buffer and the actual number of bytes read returned in Bytes.

CRRN2: If Bytes is non-null and Buffer is null, then the number of bytes that could be read is returned in Bytes.

CRRN3: If Bytes is null or *Bytes is zero the invocation is a NOP

6.1.4.1.2 Parameter errors

CRRP1: If Offset exceeds the length of the file, the status word OUT_OF_FILE_BOUNDARIES shall be returned.

6.1.4.1.3 Context errors

CRRC1: If there is no currently selected EF, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is not transparent, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling application does not fulfil the access condition, READ, to perform this function, the status word SECURITY_CONDITION_NOT_SATISFIED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for the reading of an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), an instance the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, an instance of the status word INTERNAL_ERROR shall be returned.

6.1.4.2 Test Suite Files

Additional requirements for the UICC personalisation: none.

Test Script:	API_CatReadBinary.tst
Test Application:	API_ CatReadBinary.c

6.1.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored	

1	**Read from EFICCID in MF**
	`1 - CatSelect(FID_EF_ICCID, &status);`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(0, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		bytes contains 5
		buffer contains 0F FF FF FF FF 0 0 0 0 0	

2	**Offset < EOF < Offset+Bytes**
	`1 - CatSelect(FID_EF_ICCID, &status);`
	`2- DWORD bytes=10; BYTE buffer[10]={0};`
	`CatReadBinary(5, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		bytes contains 5
		buffer contains FF FF FF FF FF 0 0 0 0 0	

3	**EOF < Offset**
	`1 - CatSelect(FID_EF_ICCID, &status);`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(100, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		bytes contains 0
		buffer contains 0 0 0 0 0 0 0 0 0 0	

4	**Buffer is null**
	`1 - CatSelect(FID_EF_ICCID, &status);`
	`2- DWORD bytes=6; BYTE buffer[10]={0};`
	`CatReadBinary(10, &bytes, NULL);`
		1- Return value: '9000'
		2- Return value: '9000'
		bytes contains 4	

5	**EF is not Transparent**
	`1 - CatSelect(FID_DF_SIMTEST, &status);`
	`2 - CatSelect(FID_EF_LARU, &status)`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(10, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: FILE_INCONSISTENT	

6	**Access condition not fulfilled**
	`1 - CatSelect(FID_DF_SIMTEST, &status);`
	`2 - CatSelect(DFSIMTTEST, &status)`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(10, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: SECURITY_CONDITION_NOT_SATISFIED	

7	**EF is invalidated**
	`1 - CatSelect(EFTNU, &status);`
	`2- CatInvalidate();`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(10, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: INVALIDATION_STATUS_CONTRADICTION	

8	**No EF selected**
	`1 – CatSelect(FID_EF_ICCID, &status)`
	`2- DWORD bytes=5; BYTE buffer[10]={0};`
	`CatReadBinary(10, &bytes, &buffer);`
		1- Return value: '9000'
		2- Return value: NO_EF_SELECTED	

6.1.4.4 Test Coverage

CRR Number	Test Case Number
N1	1,2
N2	4
N3	Not tested
P1	3
C1	8
C2	5
C3	6
C4	7
C5, C6	Not tested

6.1.5 CatUpdateBinary

Test Area Reference: API_CatUpdateBinary

6.1.5.1 Conformance Requirements

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

```
UINT16 CatUpdateBinary(DWORD Offset, DWORD Bytes, void *Buffer);
```

6.1.5.1.1 Normal execution

CRRN1: The currently selected transparent file is updated starting at Offset with the sequence of Bytes bytes in Buffer.

6.1.5.1.2 Parameter errors

CRRP1: If Bytes is positive and Buffer is null, then INCORRECT_PARAMETERS shall be returned.

6.1.5.1.3 Context errors

CRRC1: If there is no currently selected EF, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is not transparent, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling application does not fulfil the access condition, READ, to perform this function, the status word SECURITY_CONDITION_NOT_SATISFIED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for the reading of an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), an instance the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, an instance of the status word INTERNAL_ERROR shall be returned.

6.1.5.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_ CatUpdateBinary.tst

Test Application: API_ CatUpdateBinary.c

6.1.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Update Transparent EF**
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, 5, &buffer)`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		bytes contains 5	

2	**Offset < EOF, Offset+Bytes > EOF**
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2- bytes = 5`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		bytes contains 5	

3	`Offset >EOF`
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		bytes contains 5	

4	**Buffer is null**
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, NULL)`
		1- Return value: ‘9000’
		2- Return value:
		INCORRECT_PARAMETERS	

Id	Description	API Expectation	APDU Expectation

5	**No EF selected**
	`1 - CatSelect(FID_MF, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: '9000'
		2- Return value: NO_EF_SELECTED
		bytes contains 5	

6	**EF is not Transparent**
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: '9000'
		2- Return value: FILE_INCONSISTENT
		bytes contains 5	

7	**Access condition not fulfilled**
	`1 - CatSelect(FID_EF_ICCID, &status)`
	`2 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: '9000'
		2- Return value: SECURITY_CONDITION_NOT_SATISFIED
		bytes contains 5	

8	**EF is invalidated**
	`1 - CatSelect(EFTNU, &status)`
	`2 - CatInvalidate()`
	`3 - DWORD bytes=5;`
	`BYTE buffer[10]={1,2,3,4,5,6,7,8,9,0};`
	`CatUpdateBinary(0, &bytes, &buffer)`
		1- Return value: '9000'
		2- Return value: INVALIDATION_STATUS_CONTRADICTION
		bytes contains 5	

6.1.5.4 Test Coverage

CRR Number	Test Case Number
N1	1,2,3
P1	4
C1	5
C2	6
C3	7
C4	8
C5, C6	Not Tested

6.1.6 CatReadRecord

Test Area Reference: API_CatReadRecord

6.1.6.1 Conformance Requirements

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

```
UINT16 CatReadRecord(DWORD RecordNumber, CatRecordAccessModes Mode,
                     DWORD Offset, DWORD *NumBytes, void *Buffer);
```

6.1.6.1.1 Normal execution

CRRN1: NumBytes bytes starting at Offset from the record specified by Mode and RecordNumber of the currently selected linear fixed or cyclic EF are read into Buffer.

CRRN2: If the access mode is ABSOLUTE or CURRENT:

- if RecordNumber is not 0, the record addressed by RecordNumber will be read;
- if RecordNumber is 0 the current selected record will be read; and
- the current record pointer shall not change.

CRRN3: If the access mode is NEXT:

- the next record relative to the current selected record will be selected and read;
- if no current record is selected, the first record will be selected and read;
- if the current record pointer is set to the last record for a cyclic EF the record pointer is set to the first record and the record is read;

- the current record pointer of any other Application shall not be changed.

CRRN4: If the access mode is PREVIOUS:

- the previous record relative to the current selected record will be selected and read;
- if no current record is selected, the last record will be selected and read;
- if the current record pointer is set to the first record, for a linear fixed EF the entry point responses with an error exception and for a cyclic EF the record pointer is set to the last record and the record is read;
- the current record pointer of any other Application shall not be changed.

6.1.6.1.2 Parameter errors

CRRP1: If the currently selected EF is linear fixed and the access mode is ABSOLUTE and RecordNumber is greater than records available, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned

CRRP2: If the currently selected EF is linear fixed and the access mode is CURRENT, RecordNumber is 0 and there is no current record selected, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

CRRP3: If the currently selected EF is linear fixed and the access mode is NEXT and the current record pointer is set to the last record, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

CRRP4: If the currently selected EF is linear fixed and the access mode is PREVIOUS and the current record pointer is set to the first record, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

6.1.6.1.3 Context errors

CRRC1: If the calling Application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling Application does not fulfil the access condition, READ, to perform this function, the status word AC_NOT_FULFILLED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned.

6.1.6.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_ CatReadRecord.tst
Test Application:	API_ CatReadRecord.c

6.1.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Read Absolute Linear Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=1, off=0, bytes=4;`
	`CatRecordAccessMode mode = ABSOLUTE;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {55, 55, 55, 55}	

2	**Read Current Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = CURRENT;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {55, 55, 55, 55}	

3	**Read Next from Linear Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = NEXT;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {AA, AA, AA, AA}	

4	**Read Next from Linear Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = NEXT;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: RECORD_NUMBER_NOT_AVAILABLE.	

5	**Read Previous from Linear Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = PREVIOUS;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {55, 55, 55, 55}	

6	**Read Previous from Linear Fixed EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = PREVIOUS;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: RECORD_NUMBER_NOT_AVAILABLE	

7	**Read Absolute and Current from Cyclic EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFCARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = CURRENT;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {55, 55, 55, 55}	

8	**Read Next from Cyclic EF**
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFCARU, NULL)`
	`DWORD rn=0, off=0, bytes=4;`
	`CatRecordAccessMode mode = NEXT;`
	`BYTE buffer[4];`
	`3 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		buffer = {AA, AA, AA, AA}	

Id	Description	API Expectation	APDU Expectation

9	**Read Next from Cyclic EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFCARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = NEXT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		buffer = {55, 55, 55, 55}	

10	**Read Previous from Cyclic EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFCARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = PREVIOUS;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		buffer = {AA, AA, AA, AA}	

11	**Read Previous from Cyclic EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFCARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = PREVIOUS;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		buffer = {55, 55, 55, 55}	

12	**NumBytes > Record Length**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: OUT_OF_RECORD_BOUNDARIES	

13	**No current record in linear fixed EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: RECORD_NUMBER_NOT_AVAILABLE	

14	**No EF selected**
	1 - CatSelect(DFSIMTEST, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: NO_EF_SELECTED	

15	**EF is neither Cyclic nor Linear Fixed**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFTNR, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: FILE_INCONSISTENT.	

16	**Access condition not fulfilled**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFCNR, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	3 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: SECURITY_CONDITION_NOT_SATISFIED.	

17	**EF is invalidated**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	3 – CatInvalidate();
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4];
	4 - CatReadRecord(rn, mode, off, &bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		4- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.6.4 Test Coverage

CRR Number	Test Case Number
N1	1,2,3,5,7,8,9,10,11
N2	1,7
N3	3,4,8,9
N4	5,6,10,11
P1	Not Tested
P2	Not Tested
P3	4
P4	6
C1	14
C2	15
C3	16
C4	17
C5, C6	Not Tested

6.1.7 CatUpdateRecord

Test Area Reference: API_CatUpdateRecord

6.1.7.1 Conformance Requirements

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

```
UINT16 CatUpdateRecord(DWORD RecordNumber, CatRecordAccessModes Mode,
                       DWORD Offset, DWORD NumBytes, void *Buffer);
```

6.1.7.1.1 Normal execution

CRRN1: NumBytes of data starting at Offset in the record specified by Mode and RecordNumber of the current selected linear fixed or cyclic EF are overwritten by the data in Buffer.

CRRN2: If the access mode is ABSOLUTE or CURRENT and the file is a linear fixed EF:

- the record addressed by RecordNumber will be updated;
- if RecordNumber is 0 the current selected record will be updated; and
- the current record pointer shall not change.

CRRN3: If the access mode is NEXT and the file is a linear fixed EF:

- the next record relative to the current selected record will be selected and updated;
- if no current record is selected, the first record will be selected and updated;
- the current record pointer of any other application shall not be changed.

CRRN4: If the access mode is PREVIOUS:

- the previous record relative to the current selected record will be selected and updated;
- if no current record is selected, the last record will be selected and updated;
- if a cyclic EF is updated, the oldest record will be updated independent of the current record pointer and this record becomes record number 1 and the current record;
- the current record pointer of any other application shall not be changed in case of a linear fixed EF.

6.1.7.1.2 Parameter errors

CRRP1: If the currently selected EF is linear fixed and the access mode is ABSOLUTE and RecordNumber is greater than records available, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned

CRRP2: If the currently selected EF is linear fixed and the access mode is CURRENT, RecordNumber is 0 and there is no current record selected, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

CRRP3: If the currently selected EF is linear fixed and the access mode is NEXT and the current record pointer is set to the last record, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

CRRP4: If the currently selected EF is linear fixed and the access mode is PREVIOUS and the current record pointer is set to the first record, the status word RECORD_NUMBER_NOT_AVAILABLE shall be returned.

CRRP5: If the currently selected EF is linear fixed and Offset plus NumBytes is greater than the record length, then the status word OUT_OF_FILE_BOUNDARIES shall be returned.

6.1.7.1.3 Context errors

CRRC1: If the calling Application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling Application does not fulfil the access condition, UPDATE, to perform this function, the status word AC_NOT_FULFILLED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned.

6.1.7.2 Test Suite Files

Additional requirements for the UICC personalisation: This test is based on the assumption that the contents of the EFs in $DF_{SIMTEST}$ are identical to those defined in the default pre-personalisation and the current record pointers have not been altered.

Test Script:	API_CatUpdateRecord.tst
Test Application:	API_ CatUpdateRecord.c

6.1.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Update Absolute Record in Linear Fixed EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=1, off=0, bytes=4;
	CatRecordAccessMode mode = ABSOLUTE;
	BYTE buffer[4]={1,2,3,4};
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: '9000'
		buffer = {1,2,3,4}	

2	**Update Current Record in Linear Fixed EF**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = CURRENT;
	BYTE buffer[4]={1,2,3,4};
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: '9000'
		buffer = {1,2,3,4}	

3	**Update Next from Linear Fixed EF, no record pointer set**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=0, off=0, bytes=4;
	CatRecordAccessMode mode = NEXT;
	BYTE buffer[4]={1,2,3,4};
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: '9000'
		buffer = {1,2,3,4}	

4	**Update Next from Linear Fixed EF, record pointer set**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=1, off=0, bytes=4;
	CatRecordAccessMode mode = NEXT;
	BYTE buffer[4]={1,2,3,4};
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: '9000'
		buffer = {1,2,3,4}	

5	**Update Next from Linear Fixed EF, no more records** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=1, off=0, bytes=4; CatRecordAccessMode mode = NEXT; BYTE buffer[4]; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: '9000' 2- Return value: '9000' 3- Return value: RECORD_NUMBER_NOT_AVAILABLE.	

6	**Update Previous from Linear Fixed EF, no record pointer set**
	1 - CatSelect(DFSIMTEST, NULL)
	2 - CatSelect(EFLARU, NULL)
	DWORD rn=1, off=0, bytes=4;
	CatRecordAccessMode mode = PREVIOUS;
	BYTE buffer[4];
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: '9000'
		buffer = {1,2,3,4}	

7	**Update Previous from Linear Fixed EF, record pointer set** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL)	1- Return value: '9000' 2- Return value: '9000' 3- Return value: '9000' 4- Return value: '9000'	

Id	Description	API Expectation	APDU Expectation

	DWORD rn=1, off=0, bytes=4;
	CatRecordAccessMode mode = PREVIOUS;
	BYTE buffer[4];
	3 - CatUpdateRecord(rn, mode, off, bytes, buffer);
	4 – CatReadRecord(0, REC_ACC_MODE_ABSOLUTE_CURRENT, 0, &bytes, buffer);
		buffer = {1,2,3,4}	

8	**Update Previous from Linear Fixed EF , no more records** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=1, off=0, bytes=4; CatRecordAccessMode mode = PREVIOUS; BYTE buffer[4]; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: RECORD_NUMBER_NOT_AVAILABLE.	
9	**Update Previous from Cyclic EF** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=1, off=0, bytes=4; CatRecordAccessMode mode = PREVIOUS; BYTE buffer[4]={1,2,3,4}; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: ‘9000’ 4- Return value: ‘9000’ buffer = {1,2,3,4}	

10	**No EF selected**
	1 - CatSelect(DFSIMTEST, NULL)
	DWORD rn=1, off=0, bytes=4;
	CatRecordAccessMode mode = ABSOLUTE;
	BYTE buffer[4];
	2 - CatUpdateRecord(rn, mode, off, bytes, buffer);
		1- Return value: ‘9000’
		2- Return value: NO_EF_SELECTED	

11	**Update Absolute from Linear Fixed EF beyond Records** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=3, off=0, bytes=4; CatRecordAccessMode mode = ABSOLUTE; BYTE buffer[4]={1,2,3,4}; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: RECORD_NUMBER_NOT_AVAILABLE.	
12	**No current record in linear fixed EF, update current** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=0, off=0, bytes=4; CatRecordAccessMode mode = CURRENT; BYTE buffer[4]={1,2,3,4}; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: RECORD_NUMBER_NOT_AVAILABLE.	
13	**Record Length < Offset + NumBytes** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFLARU, NULL) DWORD rn=1, off=2, bytes=6; CatRecordAccessMode mode = ABSOLUTE; BYTE buffer[4]={1,2,3,4};; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: OUT_OF_RECORD_BOUNDARIES.	
14	**EF is neither Cyclic nor Linear Fixed** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFTNR, NULL) DWORD rn=1, off=0, bytes=4; CatRecordAccessMode mode = ABSOLUTE; BYTE buffer[4]={1,2,3,4};; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: FILE_INCONSISTENT.	
15	**Access condition not fulfilled** 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFCNU, NULL) DWORD rn=1, off=0, bytes=4; CatRecordAccessMode mode = ABSOLUTE; BYTE buffer[4]={1,2,3,4}; 3 - CatUpdateRecord(rn, mode, off, bytes, buffer);	1- Return value: ‘9000’ 2- Return value: ‘9000’ 3- Return value: SECURITY_CONDITION_NOT_SATISFIED.	
15	**EF is invalidated**	1- Return value: ‘9000’	

Id	Description	API Expectation	APDU Expectation

	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFLARU, NULL);`
	`3 – CatInvalidate();`
	`DWORD rn=1, off=0, bytes=4;`
	`CatRecordAccessMode mode = ABSOLUTE;`
	`BYTE buffer[4]={1,2,3,4};;`
	`4 - CatUpdateRecord(rn, mode, off, bytes, buffer);`
		2- Return value: '9000'
		3- Return value: '9000'
		4- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.7.4 Test Coverage

CRR Number	Test Case Number
N1	1,2,3,4,6,7,9
N2	2
N3	3,4
N4	6,7,9
P1	11
P2	12
P3	5
P4	8
P5	13
C1	10
C2	14
C3	15
C4	16
C5, C6	Not Tested

6.1.8 CatSearch

Test Area Reference: API_CatSearch

6.1.8.1 Conformance Requirements

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

```
UINT16 CatSearch(CatSeekModes Mode, DWORD Offset, DWORD PatternLength, const void *Pattern);
```

6.1.8.1.1 Normal execution

CRRN1: If the pattern in Pattern with the length PatternLength is found in the record being specified by mode starting at Offset, the current record pointer is set to that record and the record number is returned.

CRRN2: If mode is BEGINNING_FORWARD, the search starts with the first record forward towards the end of the file.

CRRN3: If mode is END_BACKWARD, the search starts with the last record backward towards the beginning of the file.

CRRN4: If mode is NEXT_FORWARD, the search starts from the next record after the current record pointer forward towards the end of file. If no current record pointer is selected, the search starts with the first record.

CRRN5: If mode is PREVIOUS_BACKWARD, the search starts from the previous record before the current record pointer backward towards the beginning of the file. If no current record pointer is selected the search starts with the last record.

CRRN6: If pattern in patt is not found, the status word PATTERN_NOT_FOUND shall be returned.

CRRN7: If mode is NEXT_FORWARD and the record pointer is at the last record, the status word PATTERN_NOT_FOUND shall be returned.

CRRN8: If mode is PREVIOUS_BACKWARD and the record pointer is at the first record, the status word PATTERN_NOT_FOUND shall be returned.

6.1.8.1.2 Parameter errors

CRRP1: If PatternLength is greater than the size of the record of the currently selected EF, the status word OUT_OF_RECORD_BOUNDARIES shall be returned.

CRRP2: If Offset plus PatternLength is greater than the length of the pattern array patt.length, the status word OUT_OF_RECORD_BOUNDARIES shall be returned.

6.1.8.1.3 Context errors

CRRC1: If the calling Application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling Application does not fulfil the access condition, READ, to perform this function, the status word AC_NOT_FULFILLED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned.

6.1.8.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSearch.tst

Test Application: API_ CatSearch.c

6.1.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Pattern Not Found**
	`BYTE pat[] = {0xDA, 0xDA, 0xDA}, *p;`
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFLARU, NULL)`
	`3 - CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1 – Return ‘9000’
		1 – Return ‘9000’
		3- Return PATTERN_NOT_FOUND	

2	**Search from Beginning Forward**
	`BYTE pat[] = {0x55,0x55,0x55}, *p;`
	`1- CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1 – Return 1	

3	**Search from End Backward**
	`BYTE pat[] = {0x55,0x55,0x55}, *p;`
	`1- CatSearch(END_BACKWARD, 0, sizeof(pat), pat);`
		1 – Return 1	

4	**Search from Next Forward**
	`BYTE pat[] = {0xAA, 0xAA, 0xAA }, *p;`
	`1- CatSearch(NEXT_FORWARD, 0, sizeof(pat), pat);`
		1 – Return 2	

5	**Last Record, Search from Next Forward**
	`BYTE pat[] = {55,55,55}, *p;`
	`1- CatSearch(NEXT_FORWARD, 0, sizeof(pat), pat);`
		1- Return PATTERN_NOT_FOUND	

6	**Search from Previous Backward**
	`BYTE pat[] = {55,55,55}, *p;`
	`1- CatSearch(PREVIOUS_BACKWARD, 0, sizeof(pat), pat);;`
		1 – Return 1	

7	**First Record, Search from Previous Backward** `BYTE pat[] = {55,55,55}, *p;`	1- Return PATTERN_NOT_FOUND.	

Id	Description	API Expectation	APDU Expectation
	`1- CatSearch(PREVIOUS_BACKWARD, 0, sizeof(pat), pat);`		

8	**Pattern not Found (out of reach)**
	`BYTE pat[] = {55,55,55}, *p;`
	`1- CatSearch(NEXT_FORWARD, 0, sizeof(pat), pat);`
		1- Return PATTERN_NOT_FOUND.	

9	**Record Length < PatternLength**
	`BYTE pat[] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}, *p;`
	`1- CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1 Return value: OUT_OF_RECORD_BOUNDARIES.	

10	**Offset < Record Length < Offset + PatternLength**
	`BYTE pat[] = {55,55,55}, *p;`
	`1- CatSearch(BEGINNING_FORWARD, 1, sizeof(pat), pat);`
		1 - Return value: OUT_OF_RECORD_BOUNDARIES.	

11	**EF is not Linear Fixed**
	`BYTE pat[] = {55,55,55}, *p;`
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSelect(EFTNR, NULL)`
	`3- CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1 – Return value: ‘9000’
		2 – Return value: ‘9000’
		3- Return value: FILE_INCONSISTENT.	

12	**No EF selected**
	`BYTE pat[] = {55,55,55}, *p;`
	`1 - CatSelect(DFSIMTEST, NULL)`
	`2 - CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1- Return value: ‘9000’
		2- Return value: NO_EF_SELECTED	

13	**Access condition not fulfilled**
	`BYTE pat[] = {55,55,55}, *p;`
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFCNU, NULL);`
	`3- CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: SECURITY_CONDITION_NOT_SATISFIED	

14	**EF is invalidated**
	`BYTE pat[] = {55,55,55}, *p;`
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFCNU, NULL);`
	`3 – CatInvalidate();`
	`4- CatSearch(BEGINNING_FORWARD, 0, sizeof(pat), pat);`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		4- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.8.4 Test Coverage

CRR Number	Test Case Number
N1	2
N2	3
N3	4
N4	6
N5	8
N6	1
N7	5
N8	7
P1	9
P2	10
C1	12
C2	11
C3	13
C4	14
C5, C6	Not tested

6.1.9 CatIncrease

Test Area Reference: API_CatIncrease

6.1.9.1 Conformance Requirements

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

```
UINT16 CatIncrease(DWORD Increment, DWORD *Value);
```

6.1.9.1.1 Normal execution

CRRN1: The value Increment is added to the value of the last increased / updated record in the currently selected cyclic EF. The result is stored in the oldest record and returned in Value if Value is non-null. The updated record becomes record number 1 and is selected as current record.

6.1.9.1.2 Parameter errors

CRRP1: If the result of the addition is greater than the maximum value a DWORD, the status word MAX_VALUE_REACHED shall be returned.

6.1.9.1.3 Context errors

CRRC1: If the calling Application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, the status word FILE_INCONSISTENT shall be returned.

CRRC3: If the calling Application does not fulfil the access condition, READ, to perform this function, the status word AC_NOT_FULFILLED shall be returned.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading an invalidated file, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC5: If the entry point call causes a memory problem (e.g. memory access error), the status word MEMORY_PROBLEM shall be returned.

CRRC6: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned.

CRRC 7: If Increase not allowed due to an entry in the FCP them the status word FILE_INCONSISTENT shall be returned.

6.1.9.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatIncrease.tst

Test Application: API_ CatIncrease.c

6.1.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Increase , verify response**
	`BYTE incr[] = {0,0,1,0}, value[4];`
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFCARU, NULL);`
	`3 - CatIncrease(incr, value);`
		1 – Return value: ‘9000’
		2 – Return value: ‘9000’
		3 – value = {0,0,1,0}	

2	**Increase, verify file**
	`DWORD rn=1, off=0, bytes=4;`
	`CatRecordAccessMode mode = ABSOLUTE;`
	`BYTE buffer[4];`
	`1 - CatReadRecord(rn, mode, off, &bytes, buffer);`
		1 – buffer = {0,0,1,0}	

3	**Reach Maximum Value** `BYTE incr[] = {0xFF, 0xFF, 0xFF, 0xFF }, value[4];`	1 – Return value: ‘9000’ 2 – Return value: ‘9000’	

Id	Description	API Expectation	APDU Expectation

	1 - CatSelect(DFSIMTEST, NULL);
	2 - CatSelect(EFCARU, NULL);
	3 - CatIncrease(incr, value);
		3 - Return MAX_VALUE_REACHED.	

4	**No EF selected**
	BYTE incr[]={0xFF,0xFF,0xFF,0xFF }, value[4];
	1 - CatSelect(DFSIMTEST, NULL);
	2 - CatIncrease(incr, value);
		1- Return value: ‘9000’
		2- Return value: NO_EF_SELECTED	

5	**EF is not Cyclic** BYTE incr[] = {0,0,0,1}, value[4]; 1 - CatSelect(DFSIMTEST, NULL) 2 - CatSelect(EFTNR, NULL) 3- CatIncrease(incr, value);	1 – Return value: ‘9000’ 2 – Return value: ‘9000’ 3- Return value: FILE_INCONSISTENT.	

6	**Access condition not fulfilled**
	BYTE incr[] = {0,0,0,1}, value[4];
	1 – CatSelect(DFSIMTEST, NULL);
	2 - CatSelect(EFCNU, NULL);
	3- CatIncrease(incr, value);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: SECURITY_CONDITION_NOT_SATISFIED	

7	**EF is invalidated**
	BYTE incr[] = {0,0,0,1}, value[4];
	1 - CatSelect(DFSIMTEST, NULL);
	2 - CatSelect(EFCARU, NULL);
	3 – CatInvalidate();
	4- CatIncrease(incr, value);
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: ‘9000’
		4- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.9.4 Test Coverage

CRR Number	Test Case Number
N1	1,2
P1	3
C1	4
C2	5
C3	6
C4	7
C5	Not tested
C6, C7	Not tested

6.1.10 CatInvalidate

Test Area Reference: API_CatInvalidate

6.1.10.1 Conformance Requirements

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

```
UINT16 CatInvalidate(void)
```

6.1.10.1.1 Normal execution

CRRN1: The currently selected EF of the calling application shall be invalidated, i.e. the flag in the EF file status shall be changed accordingly.

6.1.10.1.2 Parameter errors

6.1.10.1.3 Context errors

CRRC1: If the calling application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the calling application does not fulfill the access condition, INVALIDATE, the status word AC_NOT_FULFILLED shall be returned.

CRRC3: If the currently selected EF is already invalidated, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC4: If the entry point call causes a memory problem (e.g. memory access error), an instance of the status word MEMORY_PROBLEM shall be returned.

CRRC5: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned.

6.1.10.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatInvalidate.tst

Test Application: API_ CatInvalidate.c

6.1.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Invalidate EF**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFTNR, NULL);`
	`3 - CatInvalidate();`
		1 – Return value: ‘9000’
		2 – Return value: ‘9000’
		3 – Return value: ‘9000’	

2	**No EF is selected**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatInvalidate();`
		1- Return value: ‘9000’
		2- Return value: NO_EF_SELECTED	

3	**Access condition not fulfilled**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFCNIV, NULL);`
	`3 - CatInvalidate();`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: SECURITY_CONDITION_NOT_SATISFIED	

4	**EF is already invalidated**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFTNR, NULL);`
	`3 - CatInvalidate();;`
		1- Return value: ‘9000’
		2- Return value: ‘9000’
		3- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.10.4 Test Coverage

CRR number	Test Case Number
N1	1
C1	2
C2	3
C3	4
C4, C5	Not Tested

6.1.11 CatRehabilitate

Test Area Reference: API_CatRehabilitate

6.1.11.1 Conformance Requirements

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

```
UINT16 CatRehabilitate(void)
```

6.1.11.1.1 Normal execution

CRRN1: The currently selected EF of the calling application shall be rehabilitated, i.e. the flag in the EF file status shall be changed accordingly.

6.1.11.1.2 Parameter errors

6.1.11.1.3 Context errors

CRRC1: If the calling application has currently no EF selected, the status word NO_EF_SELECTED shall be returned.

CRRC2: If the calling application does not fulfill the access condition, INVALIDATE, the status word AC_NOT_FULFILLED shall be returned.

CRRC3: If the currently selected EF is already invalidated, the status word INVALIDATION_STATUS_CONTRADICTION shall be returned.

CRRC4: If the entry point call causes a memory problem (e.g. memory access error), an instance of the status word MEMORY_PROBLEM shall be returned.

CRRC5: If the entry point call causes an error to occur that is not expected and thus not handled, the status word INTERNAL_ERROR shall be returned..

6.1.11.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatRehabilitate.tst
Test Application:	API_ CatRehabilitate.c

6.1.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Rehabilitate invalidated File**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFTNR, NULL);`
	`3 - CatRehabilitate();`
		1 – No return
		1 – Return value: '9000'
		2 – Return value: '9000'
		3 – Return value: '9000'	

2	**No EF is selected**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatRehabilitate();`
		1- Return value: '9000'
		2- Return value: NO_EF_SELECTED	

3	**Access condition not fulfilled**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFCNIV, NULL);`
	`3 - CatRehabilitate ();`
		1 – No return
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: SECURITY_CONDITION_NOT_SATISFIED	

4	**Rehabilitate validated File**
	`1 - CatSelect(DFSIMTEST, NULL);`
	`2 - CatSelect(EFTNR, NULL);`
	`3 - CatRehabilitate();`
		1- Return value: '9000'
		2- Return value: '9000'
		3- Return value: INVALIDATION_STATUS_CONTRADICTION.	

6.1.11.4 Test Coverage

CRR number	Test Case Number
N1	1
C1	2
C2	3
C3	4
C4, C5	Not Tested

6.2 Registry

6.2.1 CatSetMenuString

Test Area Reference: API_CatSetMenuString

6.2.1.1 Conformance Requirements

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

```
void CatSetMenuString (BYTE MenuID, BYTE MenuStringLength, const void *MenuString,
                       const CatIconIdentifier *IconIdentifier,
                       BYTE HelpAvailable, BYTE NextAction);
```

6.2.1.1.1 Normal execution

CRRN1: Menu string with optional modifiers is added to the main menu.

CRNN2: If MenuString is null, the invocation is a NOP.

6.2.1.1.2 Parameter errors

6.2.1.1.3 Context errors

6.2.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatSetMenuString.tst
Test Application:	API_ CatSetMenuString.c

6.2.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Set Non-Null Menu String**
	`char *str = “Pick this one”;`
	`1- CatSetMenuString (1, strlen(str), str, NULL,0, 0);`
		1 – No return	1 - SETUP MENU proactive command

2	**Set Null Menu String** `1- CatSetMenuString (1, 0), NULL, NULL,0, 0);`	1 – No return	

6.2.1.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.2.2 CatNotifyOnFrameworkEvent

Test Area Reference: API_CatNotifyOnFrameworkEvent

6.2.2.1 Conformance Requirements

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

```
void CatNotifyOnFrameworkEvent(CatFrameworkEventType Event, BYTE Enabled);
```

6.2.2.1.1 Normal execution

CRRN1: Enable or disable monitoring of the framework event indicated by Event depending on the setting of Enabled.

6.2.2.1.2 Parameter errors

6.2.2.1.3 Context errors

6.2.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatNotifyOnFrameworkEvent.tst
Test Application:	API_ CatNotifyOnFrameworkEvent.c

6.2.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Enable Monitoring of an Event** `1-CatNotifyOnFrameworkEvent(EVENT_UNRECOGNIZED_ENVELOPE, 1);`	1 – No return	
2	**Disable Monitoring of an Event** `1 – CatNotifyOnFrameworkEvent(EVENT_UNRECOGNIZED_ENVELOPE, 0);`	1 – No return	

6.2.2.4 Test Coverage

CRR number	Test Case Number
N1	1,2

6.2.3 CatNotifyOnEnvelope

Test Area Reference: API_CatNotifyOnEnvelope

6.2.3.1 Conformance Requirements

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

```
void CatNotifyOnEnvelope(CatEnvelopeTagType Tag, BYTE Enabled);
```

6.2.3.1.1 Normal execution

CRRN1: Enable or disable monitoring of the envelope event indicated by Tag depending on the setting of Enabled.

6.2.3.1.2 Parameter errors

6.2.3.1.3 Context errors

6.2.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatNotifyOnEnvelope.tst

Test Application: API_ CatNotifyOnEnvelope.c

6.2.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Enable Monitoring of an Event** `1 - CatNotifyOnEnvelope(SMS_PP_DOWNLOAD_TAG, 1);`	1 – No return	
2	**Disable Monitoring of an Event** `1 - CatNotifyOnEnvelope(SMS_PP_DOWNLOAD_TAG, 0);`	1 – No return	

6.2.3.4 Test Coverage

CRR number	Test Case Number
N1	1, 2

6.2.4 CatNotifyOnEvent

Test Area Reference: API_CatNotifyOnEvent

6.2.4.1 Conformance Requirements

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

```
void CatNotifyOnEvent(CatEventType EventType, BYTE Enabled);
```

6.2.4.1.1 Normal execution

CRRN1: Enable or disable monitoring of the terminal event indicated by Event depending on the setting of Enabled.

6.2.4.1.2 Parameter errors

6.2.4.1.3 Context errors

6.2.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatNotifyOnEvent.tst

Test Application: API_ CatNotifyOnEvent.c

6.2.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Enable Monitoring of an Event** `1 - CatNotifyOnEvent(CARD_READER_STATUS, 1);`	1 – No return	
2	**Disable Monitoring of an Event** `1 - CatNotifyOnEvent(CARD_READER_STATUS, 0);`	1 – No return	

6.2.4.4 Test Coverage

CRR number	Test Case Number
N1	1, 2

6.3 Man-Machine Interface

6.3.1 CatAddItem

Test Area Reference: API_CatAddItem

6.3.1.1 Conformance Requirements

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

```
void CatAddItem(BYTE ItemTextLength, const void *ItemText, BYTE ItemIdentifier);
```

6.3.1.1.1 Normal execution

CRRN1: Add the provided item with the provided item identifier to the list of items in the current select list.

CRRN2: If ItemTextLength is zero or ItemText is null, the invocation is a NOP.

6.3.1.1.2 Parameter errors

6.3.1.1.3 Context errors

6.3.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatAddItem.tst
Test Application:	API_ CatAddItem.c

6.3.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Add a Non-Null Item**
	`char *item = “Item 1”;`
	`1 - CatAddItem(strlen(item), item, 1);`
		1 – No return	

2	**Add a Null Item** `1 - CatAddItem(0 NULL, 1);`	1 – No return	

6.3.1.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.3.2 CatSelectItem

Test Area Reference: API_CatSelectItem

6.3.2.1 Conformance Requirements

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

```
void CatSelectItem(BYTE TitleLength, const void *Title, CatSelectItemOptions Options);
```

6.3.2.1.1 Normal execution

CRRN1: Create a new select list with the provided title and options.

CRRN2: If TitleLength is zero or Title is null, the select list has no title.

6.3.2.1.2 Parameter errors

6.3.2.1.3 Context errors

6.3.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSelectItem.tst

Test Application: API_ CatSelectItem.c

6.3.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Initilize a Select List with a Title** `1 - CatSelectItem(5, “Title”, DEFAULT_STYLE_NO_HELP);`	1 – No return	
2	**Initilize a Select List without a Title** `1 - CatSelectItem(0, NULL, DEFAULT_STYLE_NO_HELP);`	1 – No return	

6.3.2.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.3.3 CatEndSelectItem

Test Area Reference: API_CatEndSelectItem

6.3.3.1 Conformance Requirements

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

```
CatGeneralResult CatEndSelectItem(BYTE *SelectedItem, const CatIconIdentifier *IconIdentifier);
```

6.3.3.1.1 Normal execution

CRRN1: Send a SELECT ITEM proactive command to the terminal using the current select list and the indicated icon; return the item picked in SelectedItem.

CRNN2: If SelectedItem is null, the invocation is for the return value only.

CRNN3: If there is no current select list or if no items have been added to the current select list, the invocation does not send a proactive command to the terminal but rather returns CAT_SUCCESS and sets *SelelectedItem to zero if SelelectedItem is non-null.

6.3.3.1.2 Parameter errors

6.3.3.1.3 Context errors

6.3.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatEndSelectItem.tst

Test Application: API_ CatEndSelectItem.c

6.3.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Non-null Select List and non-null SelectItem**
	`char *item = “Item 1”;`
	`CatGeneralResult rv;`
	`BYTE pick;`
	`1 - rv = CatSelectItem(5, “Title”, DEFAULT_STYLE_NO_HELP);`
	`2 - CatAddItem(strlen(item), item, 1);`
	`3 – rv = CatEndSelectItem(&pick, NULL);`
		1 – No return
		2 – No return
		3 – rv = CAT_SUCCESS and pick = 1
			1 - SELECT ITEM proactive command

2	**Null SelectItem**
	`char *item = “Item 1”;`
	`CatGeneralResult rv;`
	`BYTE pick;`
	`1 - rv = CatSelectItem(5, “Title”, DEFAULT_STYLE_NO_HELP);`
	`2 - CatAddItem(strlen(item), item, 1);`
	`3 – rv = CatEndSelectItem(NULL, NULL);`
		1 – No return
		2 – No return
		3 – rv = CAT_SUCCESS
			1 - SELECT ITEM proactive command

3	**Null Select List**
	`CatGeneralResult rv;`
	`BYTE pick;`
	`1 - rv = CatSelectItem(5, “Title”, DEFAULT_STYLE_NO_HELP);`
	`2 – rv = CatEndSelectItem(&pick, NULL);`
		1 – No return
		2 – rv = CAT_SUCCESS and pick = 0	

6.3.3.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3

6.3.4 CatDisplayText

Test Area Reference: API_CatDisplayText

6.3.4.1 Conformance Requirements

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

```
CatGeneralResult CatDisplayText(CatDCSValue TextDCS, BYTE TextLength, const void *Text,
            CatDisplayTextOptions Options, const CatIconIdentifier *IconIdentifier,
            BYTE ImmediateResponse);
```

6.3.4.1.1 Normal execution

CRRN1: Send a DISPLAY TEXT proactive command to the terminal with the provided text with the provided options and icon.

CRRN2: If TextLength is zero or Text is null, the invocation is a NOP that returns CAT_SUCCESS.

6.3.4.1.2 Parameter errors

6.3.4.1.3 Context errors

6.3.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_Cat CatDisplayText.tst

Test Application: API_ Cat CatDisplayText.c

6.3.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Non-zero TextLength and non-null Text**
	`CatGeneralResult rv;`
	`char *str = “Hello, world”;`
	`1 – rv = CatDisplayText(DCS_SMS_UNPACKED, strlen(str), str, NORMAL_PRIORITY_AUTO_CLEAR, NULL,0);`
		1 – rv = CAT_SUCCESS	1 - DISPLAY TEXT proactive command

2	**Zero TextLength**
	`CatGeneralResult rv;`
	`char *str = “Hello, world”;`
	`1 - rv = CatDisplayText(DCS_SMS_UNPACKED, 0 str, NORMAL_PRIORITY_AUTO_CLEAR, NULL,0);;`
		1 – rv = CAT_SUCCESS	

3	**Null Text**
	`CatGeneralResult rv;`
	`char *str = “Hello, world”;`
	`1 - rv = CatDisplayText(DCS_SMS_UNPACKED, 0, NULL, NORMAL_PRIORITY_AUTO_CLEAR, NULL,0);`
		1 – rv = CAT_SUCCESS	

6.3.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2, 3

6.3.5 CatGetInKey

Test Area Reference: API_CatGetInKey

6.3.5.1 Conformance Requirements

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

```
CatGeneralResult CatGetInKey(CatDCSValue TitleDCS, BYTE TitleLength, const void *Title,
         CatGetInKeyOptions Options, const CatIconIdentifier *IconIdentifier,
         CatDCSValue *DCSOut, void *KeyOut);
```

6.3.5.1.1 Normal execution

CRRN1: Send a GET IN KEY proactive command to the terminal with the text in Title and according to the provided Options; return a key value in *KeyOut and the alphabet indicator of the key value in *DCSOut.

CRRN2: If TitleLength is zero or Title is null, then no title is included in the proactive command.

CRRN3: If KeyOut is null, then no key value is returned.

CRRN4: If DCSOut is null, then no key value alphabet indicator is returned.

6.3.5.1.2 Parameter errors

6.3.5.1.3 Context errors

6.3.5.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetInKey.tst

Test Application: API_ CatGetInKey.c

6.3.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All pointer arguments non-null**
	`CatGeneralResult rv;`
	`Char *tit = “Title”;`
	`CatDCSValue strfmt;`
	`BYTE str[4];`
	`1 – rv = CatGetInKey(DCS_SMS_UNPACKED, strlen(tit), tit, YES_NO_OPTION_NO_HELP, NULL, &strfmt, str);`
		1 – rv = CAT_SUCCESS	1 - GET IN KEY proactive command

2	**Text null**
	`CatGeneralResult rv;`
	`Char *tit = “Title”;`
	`CatDCSValue strfmt;`
	`BYTE str[4];`
	`1 – rv = CatGetInKey(DCS_SMS_UNPACKED, strlen(tit), NULL, YES_NO_OPTION_NO_HELP, NULL, &strfmt, str);`
		1 – rv = CAT_SUCCESS	1 - GET IN KEY proactive command

3	**KeyOut null**
	`CatGeneralResult rv;`
	`Char *tit = “Title”;`
	`CatDCSValue strfmt;`
	`BYTE str[4];`
	`1 – rv = CatGetInKey(DCS_SMS_UNPACKED, strlen(tit), tit, YES_NO_OPTION_NO_HELP, NULL, &strfmt, NULL);`
		1 – rv = CAT_SUCCESS	1 - GET IN KEY proactive command

4	**DCSOut null**
	`CatGeneralResult rv;`
	`Char *tit = “Title”;`
	`CatDCSValue strfmt;`
		1 – rv = CAT_SUCCESS	1 - GET IN KEY proactive command

Id	Description	API Expectation	APDU Expectation

	BYTE str[4];
	1 – rv = CatGetInKey(DCS_SMS_UNPACKED, strlen(tit), tit, YES_NO_OPTION_NO_HELP, NULL, NULL, str);		

6.3.5.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3
N4	4

6.3.6 CatGetInput

Test Area Reference: API_CatGetInput

6.3.6.1 Conformance Requirements

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

```
CatGeneralResult CatGetInput(CatDCSValue TitleDCS, BYTE TitleLength, const void *Title,
        CatGetInputOptions Options, CatDCSValue DefaultReplyDCS,
        BYTE DefaultReplyLength, const void *DefaultReply,
        BYTE MinimumResponseLength, BYTE MaximumResponseLength,
        const CatIconIdentifier *IconIdentifier,
        CatDCSValue *MsgOutDCS, BYTE *MsgOutLength, void *MsgOut);
```

6.3.6.1.1 Normal execution

CRRN1: Send a GET INPUT proactive command to the terminal with the provided Title and acquire a sequence of key hits according to Options, MinimumResponseLength and MaximumResponseLength; return the MsgOutLength length sequence in MsgOut and the alphabet indicator of the returned values in MsgOutDCS.

CRRN2: If TitleLength is zero or Title is null then no title is displayed.

CRRN3: If MsgOut is null, then the sequence is not returned.

CRRN4: If MsgOutDCS is null, then no alphabet indicator is returned.

CRRN5: If MinimumResponseLength is greater than MaximumResponseLength then it is taken to be equal to MaximumResponseLength.

6.3.6.1.2 Parameter errors

6.3.6.1.3 Context errors

6.3.6.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetInput.tst

Test Application: API_ CatGetInput.c

6.3.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All pointer arguments non-null**
	`CatGeneralResult rv;`
	`char *who = “Who are you?”;`
	`CatDCSValue strfmt;`
	`BYTE strlgth, str[9];`
	`1 - rv = CatGetInput(DCS_SMS_UNPACKED, strlen(who), who, UNPACKED_SMS_ALPHABET_NO_HELP, DCS_SMS_UNPACKED, 8, “Sam”, 1, 8, NULL, &strfmt, &strlgth, str);`
		1 – rv = CAT_SUCCESS	1 - GET INPUT proactive command

2	**Text null**
	`CatGeneralResult rv;`
	`char *who = “Who are you?”;`
	`CatDCSValue strfmt;`
	`BYTE strlgth, str[9];`
	`1 - rv = CatGetInput(DCS_SMS_UNPACKED, strlen(who), NULL, UNPACKED_SMS_ALPHABET_NO_HELP, DCS_SMS_UNPACKED, 8, “Sam”, 1, 8, NULL, &strfmt, &strlgth, str);;`
		1 – rv = CAT_SUCCESS	1 - GET INPUT proactive command

3	**MsgOut null**
	`CatGeneralResult rv;`
	`char *who = “Who are you?”;`
	`CatDCSValue strfmt;`
	`BYTE strlgth, str[9];`
	`1 - rv = CatGetInput(DCS_SMS_UNPACKED, strlen(who), who, UNPACKED_SMS_ALPHABET_NO_HELP, DCS_SMS_UNPACKED, 8, “Sam”, 1, 8, NULL, &strfmt, &strlgth, NULL);`
		1 – rv = CAT_SUCCESS	1 - GET INPUT proactive command

4	**MsgOutDCS null**
	`CatGeneralResult rv;`
	`char *who = “Who are you?”;`
	`CatDCSValue strfmt;`
	`BYTE strlgth, str[9];`
	`1 - rv = CatGetInput(DCS_SMS_UNPACKED, strlen(who), who, UNPACKED_SMS_ALPHABET_NO_HELP, DCS_SMS_UNPACKED, 8, “Sam”, 1, 8, NULL, &strfmt, NULL, str);`
		1 – rv = CAT_SUCCESS	1 - GET INPUT proactive command

5	**MinimumResponseLength > MaximumResponseLength**
	`CatGeneralResult rv;`
	`char *who = “Who are you?”;`
	`CatDCSValue strfmt;`
	`BYTE strlgth, str[9];`
	`1 - rv = CatGetInput(DCS_SMS_UNPACKED, strlen(who), who, UNPACKED_SMS_ALPHABET_NO_HELP, DCS_SMS_UNPACKED, 8, “Sam”, 9, 8, NULL, &strfmt, &strlgth, str);`
		1 – rv = CAT_SUCCESS	1 - GET INPUT proactive command

6.3.6.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3
N4	4
N5	5

6.3.7 CatSetupIdleModeText

Test Area Reference: API_CatSetupIdleModeText

6.3.7.1 Conformance Requirements

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

```
    CatGeneralResult CatSetupIdleModeText(CatDCSValue TextDCS, BYTE TextLength, const void *Text,
                                          const CatIconIdentifier *IconIdentifier);
```

6.3.7.1.1 Normal execution

CRRN1: Send a SETUP IDLE MODE TEXT proactive command to the terminal with Text in the indicated alphabet and with an optional icon.

CRRN2: If TextLength is zero or Text is null, then the invocation is a NOP.

6.3.7.1.2 Parameter errors

6.3.7.1.3 Context errors

6.3.7.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSetupIdleModeText.tst

Test Application: API_ CatSetupIdleModeText.c

6.3.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Non-null Text**
	`CatGeneralResult rv;`
	`char *str = “Zzzzzzzz”;`
	`1 – rv = CatSetupIdleModeText(DCS_SMS_UNPACKED, strlen(str), str, NULL);`
		1 – rv = CAT_SUCCESS	1 - SETUP IDLE MODE TEXT proactive command

2	**Null Text**
	`CatGeneralResult rv;`
	`char *str = “Zzzzzzzz”;`
	`1 – rv = CatSetupIdleModeText(DCS_SMS_UNPACKED, strlen(str), NULL, NULL);;`
		1 – rv = CAT_SUCCESS	

3	**Zero TextLength**
	`CatGeneralResult rv;`
	`char *str = “Zzzzzzzz”;`
	`1 – rv = CatSetupIdleModeText(DCS_SMS_UNPACKED, 0, str, NULL);`
		1 – rv = CAT_SUCCESS	

6.3.7.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2, 3

6.3.8 CatPlayTone

Test Area Reference: API_CatPlayTone

6.3.8.1 Conformance Requirements

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

```
    CatGeneralResult CatPlayTone(BYTE TextLength, const void *Text,
                                 CatTone Tone, CatTimeUnit Units, BYTE Duration,
                                 const CatIconIdentifier *IconIdentifier);
```

6.3.8.1.1 Normal execution

CRRN1: Send a PLAY TONE proactive command to the terminal with the provided data.

CRRN2: If Duration is zero, the invocation is a NOP that returns CAT_SUCCESS.

6.3.8.1.2 Parameter errors

6.3.8.1.3 Context errors

6.3.8.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPlayTone.tst

Test Application: API_ CatPlayTone.c

6.3.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Duration is positive, Text is non-null**
	`CatGeneralResult rv;`
	`char *str = “Ping”;`
	`1 - CatPlayTone(strlen(str), str, GENERAL_BEEP, GSM_SECONDS, 1, NULL);`
		1 – rv = CAT_SUCCESS	1 - PLAY TONE proactive command

2	**Duration is positive, Text is null**
	`CatGeneralResult rv;`
	`char *str = “Ping”;`
	`1 - CatPlayTone(strlen(str), NULL, GENERAL_BEEP, GSM_SECONDS, 1, NULL);`
		1 – rv = CAT_SUCCESS	1 - PLAY TONE proactive command

3	**Duration is zero**
	`CatGeneralResult rv;`
	`char *str = “Ping”;`
	`1 - CatPlayTone(strlen(str), str, GENERAL_BEEP, GSM_SECONDS, 0, NULL);`
		1 – rv = CAT_SUCCESS	

6.3.8.4 Test Coverage

CRR number	Test Case Number
N1	1, 2
N2	3

6.4 Timers

6.4.1 CatGetTimer

Test Area Reference: API_CatGetTimer

6.4.1.1 Conformance Requirements

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

```
    BYTE CatGetTimer(void);
```

6.4.1.1.1 Normal execution

CRRN1: Send a GET TIMER proactive command to the terminal; return a timer index.

6.4.1.1.2 Parameter errors

6.4.1.1.3 Context errors

6.4.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetTimer.tst

Test Application: API_ CatGetTimer.c

6.4.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**CatGetTimer** `1 - CatGetTimer();`	1 – Return value between 1 and 4	1 - TIMER MANAGEMENT proactive command

6.4.1.4 Test Coverage

CRR number	Test Case Number
N1	1

6.4.2 CatFreeTimer

Test Area Reference: API_CatFreeTimer

6.4.2.1 Conformance Requirements

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

```
    void CatFreeTimer(BYTE TimerID);
```

6.4.2.1.1 Normal execution

CRRN1: Send a FREE TIMER proactive command to the terminal with the provided timer index.

6.4.2.1.2 Parameter errors

6.4.2.1.3 Context errors

6.4.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatFreeTimer.tst

Test Application: API_ CatFreeTimer.c

6.4.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatFreeTimer**
	`BYTE timer;`
	`1 – timer = CatGetTimer();`
	`2 - CatFreeTimer(time);`
		1 – Return value between 1 and 4
		2 – No return
			1 - TIMER MANAGEMENT proactive command
			2 - TIMER MANAGEMENT proactive command

6.4.2.4 Test Coverage

CRR number	Test Case Number
N1	1

6.4.3 CatStartTimer

Test Area Reference: API_CatStartTimer

6.4.3.1 Conformance Requirements

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

```
void CatStartTimer(BYTE TimerID, CatTimerValue *TimerValue);
```

6.4.3.1.1 Normal execution

CRRN1: Send a START TIMER proactive command to the terminal with the provided data.

CRRN2: If TimerValue is null, the invocation is a NOP.

6.4.3.1.2 Parameter errors

6.4.3.1.3 Context errors

6.4.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatStartTimer.tst

Test Application: API_ CatStartTimer.c

6.4.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**TimerValue non-null**
	`BYTE timer;`
	`CatTimerValue duration = {0,0,5};`
	`1 – timer = CatGetTimer();`
	`2 - CatStartTimer(timer, &duration);`
		1 – Return value between 1 and 4
		2 – No return
			1 - TIMER MANAGEMENT proactive command
			2 - TIMER MANAGEMENT proactive command

2	**TimerValue null**
	`BYTE timer;`
	`CatTimerValue duration = {0,0,5};`
	`1 – timer = CatGetTimer();`
	`2 - CatStartTimer(timer, NULL);`
		1 – Return value between 1 and 4
		2 – No return
			1 - TIMER MANAGEMENT proactive command

6.4.3.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.4.4 CatGetTimerValue

Test Area Reference: API_CatGetTimerValue

6.4.4.1 Conformance Requirements

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

```
void CatGetTimerValue(BYTE TimerID, CatTimerValue *TimerValue);
```

6.4.4.1.1 Normal execution

CRRN1: Send a GET TIMER VALUE to the terminal; return a timer value in *TimerValue if TimerValue is non-null.

CRRN2: If TimerValue is null, the invocation is a NOP.

6.4.4.1.2 Parameter errors

6.4.4.1.3 Context errors

6.4.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetTimerValue.tst

Test Application: API_ CatGetTimerValue.c

6.4.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**TimerValue non-null**
	`BYTE timer;`
	`CatTimerValue duration = {0,0,5};`
	`1 – timer = CatGetTimer();`
	`2 - CatStartTimer(timer, &duration);`
	`3 - CatGetTimerValue(timer, &duration);`
		1 – Return value between 1 and 4
		2 – No return
		3 – duration is valid
			1 - TIMER MANAGEMENT proactive command
			2 - TIMER MANAGEMENT proactive command
			3 – TIMER MANGEMENT proactive command

2	**TimerValue null**
	`BYTE timer;`
	`CatTimerValue duration = {0,0,5};`
	`1 – timer = CatGetTimer();`
	`2 - CatStartTimer(timer, NULL);`
	`3 - CatGetTimerValue(timer, NULL);`
		1 – Return value between 1 and 4
		2 – No return
		3 – No return
			1 - TIMER MANAGEMENT proactive command
			2 - TIMER MANAGEMENT proactive command

6.4.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.5 Supplementary Card Reader Management

6.5.1 CatPowerOnCard

Test Area Reference: API_CatPowerOnCard

6.5.1.1 Conformance Requirements

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

```
CatGeneralResult CatPowerOnCard(CatDevice DeviceID, BYTE *ATRLength, void *ATR);
```

6.5.1.1.1 Normal execution

CRRN1: Send a POWER ON CARD proactive command to the terminal with the provided data; return the ATR of the card in the indicated card reader in ATR.

CRRN2: If either ATRLength or ATR is null, then the proactive command is issued but the ATR is not returned.

6.5.1.1.2 Parameter errors

6.5.1.1.3 Context errors

6.5.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPowerOnCard.tst

Test Application: API_ CatPowerOnCard.c

6.5.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**ATRLength and ATR non-null**
	`CatGeneralResult rv;`
	`BYTE atrlgth = sizeof(atr), atr[32];`
	`1 – rv = CatPowerOnCard(DEVICE_CARD_READER_1, &atrlgth, atr);`
		1 – rv = CAT_SUCCESS and atr contains an ATR	1 - POWER ON CARD proactive command

2	**ATRLength null**
	`CatGeneralResult rv;`
	`BYTE atrlgth = sizeof(atr), atr[32];`
	`1 – rv = CatPowerOnCard(DEVICE_CARD_READER_1, NULL, atr);`
		1 – rv = CAT_SUCCESS	1 - POWER ON CARD proactive command

3	**ATR null**
	`CatGeneralResult rv;`
	`BYTE atrlgth = sizeof(atr), atr[32];`
	`1 – rv = CatPowerOnCard(DEVICE_CARD_READER_1, &atrlgth, NULL);`
		1 – rv = CAT_SUCCESS	1 - POWER ON CARD proactive command

6.5.1.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2, 3

6.5.2 CatPowerOffCard

Test Area Reference: API_CatPowerOffCard

6.5.2.1 Conformance Requirements

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

```
CatGeneralResult CatPowerOffCard(CatDevice DeviceID);
```

6.5.2.1.1 Normal execution

CRRN1: Send a POWER OFF CARD proactive command to the terminal.

6.5.2.1.2 Parameter errors

6.5.2.1.3 Context errors

6.5.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPowerOffCard.tst

Test Application: API_ CatPowerOffCard.c

6.5.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**CatPowerOffCard** 1-CatPowerOffCard(DEVICE_CARD_READER_1);	1 – rv = CAT_SUCCESS	1 - POWER OFF CARD proactive command

6.5.2.4 Test Coverage

CRR number	Test Case Number
N1	1

6.5.3 CatPerformCardAPDU

Test Area Reference: API_CatPerformCardAPDU

6.5.3.1 Conformance Requirements

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

```
CatGeneralResult CatPerformCardAPDU(CatDevice DeviceID,
                                BYTE CAPDULength, const void *CAPDU,
                                BYTE *RAPDULength, void *RAPDU);
```

6.5.3.1.1 Normal execution

CRRN1: Send a PERFORM CARD APDU command to the terminal with the provided data; return the response of the card at RAPDU.

CRRN2: If RAPDULength or RAPDU is null, the response of the card in the indicated card reader is not returned.

6.5.3.1.2 Parameter errors

6.5.3.1.3 Context errors

6.5.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPerformCardAPDU.tst

Test Application: API_ CatPerformCardAPDU.c

6.5.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**RAPDULength and RAPDU non-null**
	`CatGeneralResult = rv;`
	`BYTE CAPDU[]={0x00,0x84,0x00,0x00,0x08};`
	`BYTE RAPDULen = sizeof(RAPDU), RAPDU[10];`
	`1 – rv =`
	`CatPerformCardAPDU(DEVICE_CARD_READER_1,`
	`sizeof(CAPDU), CAPDU,`
	`&RAPDULen, RAPDU);`
		1 – rv = CAT_SUCCESS, RAPDULen = 8 and RAPDU contains a sequence of 8 bytes	1 – PERFORM CARD APDU proactive command

2	**RAPDULength null**
	`CatGeneralResult = rv;`
	`BYTE CAPDU[]={0x00,0x84,0x00,0x00,0x08};`
	`BYTE RAPDULen = sizeof(RAPDU), RAPDU[10];`
	`1 – rv =`
	`CatPerformCardAPDU(DEVICE_CARD_READER_1,`
	`sizeof(CAPDU), CAPDU,`
	`NULL, RAPDU);`
		1 – rv = CAT_SUCCESS	1 – PERFORM CARD APDU proactive command

3	**RAPDU null**
	`CatGeneralResult = rv;`
	`BYTE CAPDU[]={0x00,0x84,0x00,0x00,0x08};`
	`BYTE RAPDULen = sizeof(RAPDU), RAPDU[10];`
	`1 – rv =`
	`CatPerformCardAPDU(DEVICE_CARD_READER_1,`
	`sizeof(CAPDU), CAPDU,`
	`&RAPDULen, NULL);`
		1 – rv = CAT_SUCCESS	1 – PERFORM CARD APDU proactive command

6.5.3.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2, 3

6.5.4 CatGetReaderStatus

Test Area Reference: API_CatGetReaderStatus

6.5.4.1 Conformance Requirements

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

```
CatGeneralResult CatGetReaderStatus(CatDevice DeviceID, CatReaderStatusOptions Options,
                                    BYTE *Status);
```

6.5.4.1.1 Normal execution

CRRN1: Send a GET READER STATUS proactive command with the provided data to the terminal; return the status of the indicated device at Status.

CRRN2: If Status is null then the invocation is a NOP that returns CAT_SUCCESS.

6.5.4.1.2 Parameter errors

6.5.4.1.3 Context errors

6.5.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetReaderStatus.tst

Test Application: API_ CatGetReaderStatus.c

6.5.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Status non-null**
	`CatGeneralResult rv;`
	`BYTE status;`
	`CatGetReaderStatus(DEVICE_CARD_READER_1, CARD_READER_STATUS, &status);`
		1 – rv = CAT_SUCCESS and status contains a reader status value	1 - GET READER STATUS proactive command

2	**Status null**
	`CatGeneralResult rv;`
	`BYTE status;`
	`CatGetReaderStatus(DEVICE_CARD_READER_1, CARD_READER_STATUS, NULL);`
		1 – rv = CAT_SUCCESS	

6.5.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6 Network Services

6.6.1 CatGetLocationInformation

Test Area Reference: API_CatGetLocationInformation

6.6.1.1 Conformance Requirements

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

```
CatGeneralResult CatGetLocationInformation(CatLocationInformation *LocationInformation);
```

6.6.1.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the terminal; return the retrieved local information in LocationInformation.

CRRN2: If LocationInformation is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.1.1.2 Parameter errors

6.6.1.1.3 Context errors

6.6.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetLocationInformation.tst

Test Application: API_ CatGetLocationInformation.c

6.6.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**LocationInformation non-null**
	`CatGeneralResult rv;`
	`CatLocationInformation linfo;`
	`1 - rv = CatGetLocationInformation(&linfo);`
		1 – rv = CAT_SUCCESS and linfo contains local information.	1 – PROVIDE LOCAL INFORMATION proactive command

2	**LocationInformation null**
	`CatGeneralResult rv;`
	`1 - rv = CatGetLocationInformation(NULL);`
		1 – rv = CAT_SUCCESS	

6.6.1.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.2 CatGetTimingAdvance

Test Area Reference: API_CatGetTimingAdvance

6.6.2.1 Conformance Requirements

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

```
CatGeneralResult CatGetTimingAdvance(CatTimingAdvance *TimingAdvance);
```

6.6.2.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the terminal to retrieve the timing advance information; return the retrieved timing advance information at TimingAdvance.

CRRN2: If TimingAdvance is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.2.1.2 Parameter errors

6.6.2.1.3 Context errors

6.6.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetTimingAdvance.tst

Test Application: API_ CatGetTimingAdvance.c

6.6.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**TimingAdvnace non-null**
	`CatGeneralResult rv;`
	`CatTimingAdvance tinfo;`
	`1 – rv = CatGetTimingAdvance(&tinfo);`
		1 – rv = CAT_SUCCESS and tinfo contains timing information	1 – PROVIDE LOCAL INFORMATION proactive command

2	**TimingAdvnace null**
	`CatGeneralResult rv;`
	`1 – rv = CatGetTimingAdvance(NULL);`
		1 – rv = CAT_SUCCESS	

6.6.2.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.3 CatGetIMEI

Test Area Reference: API_CatGetIMEI

6.6.3.1 Conformance Requirements

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

```
CatGeneralResult CatGetIMEI(BYTE IMEI[8]);
```

6.6.3.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the termina to retrieve the IMEI; return the retrieved IMEI in IMEI.

CRRN2: If IMEI is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.3.1.2 Parameter errors

6.6.3.1.3 Context errors

6.6.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetIMEI.tst

Test Application: API_ CatGetIMEI.c

6.6.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**IMEI non-null**
	`CatGeneralResult rv;`
	`BYTE IMEI[8];`
	`1 – rv = CatGetIMEI(IMEI);`
		1 – rv = CAT_SUCCESS and IMEI contains IMEI	1 – PROVIDE LOCAL INFORMATION proactive command

2	**IMEI null**
	`CatGeneralResult rv;`
	`1 – rv = CatGetIMEI(NULL);`
		1 – rv = CAT_SUCCESS	

6.6.3.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.4 CatGetNetworkMeasurementResults

Test Area Reference: API_CatGetNetworkMeasurementResults

6.6.4.1 Conformance Requirements

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

```
CatGeneralResult CatGetNetworkMeasurementResults(BYTE MeasurementResults[10]);
```

6.6.4.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the terminal to retrieve the measurement results; return the retrieved measurement results in MeasurementResults.

CRRN2: If MeasurementResults is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.4.1.2 Parameter errors

6.6.4.1.3 Context errors

6.6.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetNetworkMeasurementResults.tst

Test Application: API_ CatGetNetworkMeasurementResults.c

6.6.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**MeasurementResult non-null**
	`CatGeneralResult rv;`
	`BYTE NMR[10];`
	`1 – rv = CatGetNetworkMeasurementResults`
	`(NMR);`
		1 – rv = CAT_SUCCESS and NMR contains network measurement results	1 – PROVIDE LOCAL INFORMATION proactive command

2	**MeasurementResult null** `CatGeneralResult rv;`	1 – rv = CAT_SUCCESS	

Id	Description	API Expectation	APDU Expectation
	`1 – rv = CatGetNetworkMeasurementResults (NULL);`		

6.6.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.5 CatGetDateTimeAndTimeZone

Test Area Reference: API_CatGetDateTimeAndTimeZone

6.6.5.1 Conformance Requirements

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

```
CatGeneralResult CatGetDateTimeAndTimeZone(BYTE DateTimeAndTimeZone[7]);
```

6.6.5.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the terminal to retrieve the date, time and time zone; return the retrieved date, time and time zone in DateTimeAndTimeZone.

CRRN2: If DateTimeAndTimeZone is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.5.1.2 Parameter errors

6.6.5.1.3 Context errors

6.6.5.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetDateTimeAndTimeZone.tst

Test Application: API_ CatGetDateTimeAndTimeZone.c

6.6.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**DateTimeAndTimeZone non-null**
	`CatGeneralResult rv;`
	`BYTE DTTZ[7];`
	`1 – rv = CatGetDateTimeAndTimeZone(DTTZ);`
		1 – rv = CAT_SUCCESS and DTTZ contains data, time and time zone information	1 – PROVIDE LOCAL INFORMATION proactive command

2	**DateTimeAndTimeZone null**
	`CatGeneralResult rv;`
	`1 – rv = CatGetDateTimeAndTimeZone(NULL);`
		1 – rv = CAT_SUCCESS	

6.6.5.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.6 CatGetLanguage

Test Area Reference: API_CatGetLanguage

6.6.6.1 Conformance Requirements

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

```
CatGeneralResult CatGetLanguage(BYTE Language[2]);
```

6.6.6.1.1 Normal execution

CRRN1: Send a PROVIDE LOCAL INFORMATION proactive command to the terminal to retrieve the current language selection; return the retrieved language selection in Language.

CRRN2: If Language is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.6.1.2 Parameter errors

6.6.6.1.3 Context errors

6.6.6.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetLanguage.tst

Test Application: API_ CatGetLanguage.c

6.6.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Language non-null**
	`CatGeneralResult rv;`
	`BYTE language[2];`
	`1 – rv = CatGetLanguage(language);`
		1 – rv = CAT_SUCCESS and language contains the language code	1 – PROVIDE LOCAL INFORMATION proactive command

2	**Language null**
	`CatGeneralResult rv;`
	`1 – rv = CatGetLanguage(NULL);`
		1 – rv = CAT_SUCCESS	

6.6.6.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.7 CatSetupCall

Test Area Reference: API_CatSetupCall

6.6.7.1 Conformance Requirements

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

```
    CatGeneralResult CatSetupCall(BYTE UserConfirmationMessageLength,
              const void *UserConfirmationMessage,
              CatTypeOfNumberAndNumberingPlanIdentifier TONandNPI,
              BYTE DiallingNumberLength, const void *DiallingNumber,
              CatSetupCallOptions Options,
              const CatIconIdentifier *UserConfirmationIconIdentifier,
              BYTE CallSetupMessageLength, const void *CallSetupMessage,
              const CatIconIdentifier *CallSeupIconIdentifier);
```

6.6.7.1.1 Normal execution

CRRN1: Send a SETUP CALL proactive command to the terminal with the provided data.

6.6.7.1.2 Parameter errors

CRRP1: If DiallingNumberLength is zero or Dialling Number is null, the status word CAT_REQUIRED_VALUES_MISSING is returned.

6.6.7.1.3 Context errors

6.6.7.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSetupCall.tst

Test Application: API_ CatSetupCall.c

6.6.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**DialingNumber non-null**
	`CatGeneralResult rv;`
	`char *confirmMsg = “OK?”;`
	`char *setupMSG = “Go!”;`
	`BYTE number[] = {1,6,1,7,9,6,4,1,7,9,8};`
	`1 - CatGeneralResult CatSetupCall(`
	`strlen(confirmMsg), confirmMsg,`
	`TON_INTERNATIONAL_AND_NPI_TELEPHONE,`
	`sizeof(number), number,`
	`CALL_ONLY_IF_NOT_BUSY ,`
	`NULL, strlen(setupMsg), setupMSG,`
	`NULL);`
		1 – rv = CAT_SUCCESS	1 – SETUP CALL proactive command

2	**DialingNumber null**
	`CatGeneralResult rv;`
	`char *confirmMsg = “OK?”;`
	`char *setupMSG = “Go!”;`
	`BYTE number[] = {1,6,1,7,9,6,4,1,7,9,8};`
	`1 - CatGeneralResult CatSetupCall(`
	`sizeof(confirmMsg), confirmMsg,`
	`TON_INTERNATIONAL_AND_NPI_TELEPHONE,`
	`sizeof(number), NULL,`
	`CALL_ONLY_IF_NOT_BUSY ,`
	`NULL, sizeof(setupMsg), setupMSG,`
	`NULL);`
		1 – rv = CAT_REQUIRED_VALUES_MISSING	

6.6.7.4 Test Coverage

CRR number	Test Case Number
N1	1
P1	2

6.6.8 CatSendShortMessage

Test Area Reference: API_CatSendShortMessage

6.6.8.1 Conformance Requirements

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

```
CatGeneralResult CatSendShortMessage(BYTE TitleLength, const void *Title,
                 CatTypeOfNumberAndNumberingPlanIdentifier TONandNPI,
                 BYTE AddressLength, const void *Address,
                 BYTE SmsTPDULength, const void *SmsTPDU,
                 CatSendShortMessageOptions Options,
                 const CatIconIdentifier *IconIdentifier);
```

6.6.8.1.1 Normal execution

CRRN1: Send a SEND SHORT MESSAGE proactive command to the terminal with the provided data.

6.6.8.1.2 Parameter errors

CRRP1: If AddressLength is zero or Address is null, the value CAT_REQUIRED_VALUES_MISSING is returned.

CRRP2: If SmsTPDULength is zero or SmsTPDU is null, the value CAT_REQUIRED_VALUES_MISSING is returned.

6.6.8.1.3 Context errors

6.6.8.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendShortMessage.tst

Test Application: API_ CatSendShortMessage.c

6.6.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Address and TPDU Arguments non-null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE address[] = {1,6,1,7,9,6,4,1,7,9,8};`
	`BYTE SmsTPDU[] = {1,2,3,4,5,6,7,8};`
	`1 – rv = CatSendShortMessage(strlen (title), title, TON_INTERNATIONAL_AND_NPI_TELEPHONE, sizeof(address), address, sizeof(SmsTPDU), SmsTPDU, PACKING_NOT_REQUIRED, NULL);`
		1 – rv = CAT_SUCCESS	1 – SEND SMS proactive command

2	**Address Arguments null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE address[] = {1,6,1,7,9,6,4,1,7,9,8};`
	`BYTE SmsTPDU[] = {1,2,3,4,5,6,7,8};`
	`1 – rv = CatSendShortMessage(strlen (title), title, TON_INTERNATIONAL_AND_NPI_TELEPHONE,`
		1 – rv = CAT_REQUIRED_VALUES_MISSING	

Id	Description	API Expectation	APDU Expectation

	`sizeof(address), NULL,`
	`sizeof(SmsTPDU), SmsTPDU,`
	`PACKING_NOT_REQUIRED, NULL);`		

3	**TPDU arguments null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE address[] = {1,6,1,7,9,6,4,1,7,9,8};`
	`BYTE SmsTPDU[] = {1,2,3,4,5,6,7,8};`
	`1 – rv =`
	`CatSendShortMessage(strlen(title), title,`
	`TON_INTERNATIONAL_AND_NPI_TELEPHONE,`
	`sizeof(address), address,`
	`sizeof(SmsTPDU), NULL,`
	`PACKING_NOT_REQUIRED, NULL);`
		1 – rv = CAT_REQUIRED_VALUES_MISSING	

6.6.8.4 Test Coverage

CRR number	Test Case Number
N1	1
P1	2
P2	3

6.6.9 CatSendSS

Test Area Reference: API_CatSendSS

6.6.9.1 Conformance Requirements

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

```
CatGeneralResult CatSendSS(BYTE TitleLength, const void *Title,
       CatTypeOfNumberAndNumberingPlanIdentifier TONandNPI,
       BYTE SSStringLength, const void *SSString,
       const CatIconIdentifier *IconIdentifier);
```

6.6.9.1.1 Normal execution

CRRN1: Send a SEND SS proactive command to the terminal with the provided data.

CRRN2: If SSStringLength is zero or SSString is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.9.1.2 Parameter errors

6.6.9.1.3 Context errors

6.6.9.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendSS.tst

Test Application: API_ CatSendSS.c

6.6.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**String arguments non-zero** `CatGeneralResult rv;`	1 – rv = CAT_SUCCESS	1 - SEND SS proactive

Id	Description	API Expectation	APDU Expectation

	`char *title = “Title”;`
	`BYTE SS[] = {1,2,3,4,5,6,7,8};`
	`1 = rv = CatSendSS(strlen(title), title, TON_INTERNATIONAL_AND_NPI_TELEPHONE, sizeof(SS), SS, NULL);`
			command

2	**String arguments zero**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE SS[] = {1,2,3,4,5,6,7,8};`
	`1 - rv = CatSendSS(strlen(title), title, TON_INTERNATIONAL_AND_NPI_TELEPHONE, sizeof(SS), NULL, NULL);`
		1 – rv = CAT_SUCCESS	

6.6.9.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.10 CatSendUSSD

Test Area Reference: API_CatSendUSSD

6.6.10.1 Conformance Requirements

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

```
CatGeneralResult CatSendUSSD(BYTE TitleLength, const void *Title,
         CatDCSValue MessageDCS, BYTE MessageLength, const void *Message,
         CatDCSValue *MsgOutDCS, BYTE *MsgOutLength, void *MsgOut,
         const CatIconIdentifier *IconIdentifier);
```

6.6.10.1.1 Normal execution

CRRN1: Send a SEND USSD proactive command to the terminal with the provided data.

CRRN2: If MessageLength is zero or Message is null, the invocation is a NOP that returns CAT_SUCCESS.

CRRN3: If MsgOutDCS, MsgOutLength or MsgOut is null, only the CatGeneralResult return value of execution the function is returned.

6.6.10.1.2 Parameter errors

6.6.10.1.3 Context errors

6.6.10.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendUSSD.tst

Test Application: API_ CatSendUSSD.c

6.6.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**Message and Msg arguments non-zero**	1 – rv = CAT_SUCCESS	1 – SEND USSD proactive

Id	Description	API Expectation	APDU Expectation

	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE USSDmsg = {1,2,3,4,5,6,7,8};`
	`CatDCSValue responseDCS;`
	`BYTE responseMsg[36];`
	`BYTE *responseLength`
	`1 – rv = CatSendUSSD(strlen(title), title, DCS_SMS_UNPACKED, sizeof(USSDmsg), USSDmsg, &outDCS, &outLength, outMsg, NULL);`
			command

2	**Message arguments null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE USSDmsg = {1,2,3,4,5,6,7,8};`
	`CatDCSValue responseDCS;`
	`BYTE responseMsg[36];`
	`BYTE *responseLength`
	`1 – rv = CatSendUSSD(strlen(title), title, DCS_SMS_UNPACKED, sizeof(USSDmsg), NULL, &outDCS, &outLength, outMsg, NULL);`
		1 – rv = CAT_SUCCESS	1 – SEND USSD proactive command

3	**Response arguments null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE USSDmsg = {1,2,3,4,5,6,7,8};`
	`CatDCSValue responseDCS;`
	`BYTE responseMsg[36];`
	`BYTE *responseLength`
	`1 – rv = CatSendUSSD(strlen(title), title, DCS_SMS_UNPACKED, sizeof(USSDmsg), USSDmsg, &outDCS, &outLength, NULL, NULL);`
		1 – rv = CAT_SUCCESS	1 – SEND USSD proactive command

6.6.10.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3

6.6.11 CatOpenCSChannel

Test Area Reference: API_CatOpenCSChannel

6.6.11.1 Conformance Requirements

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

```
CatGeneralResult CatOpenCSChannel(CatOpenChannelOptions Options,
             BYTE UserConfirmationLength, const void *UserConfirmation,
             const CatIconIdentifier *UserConfimationIconIdentifier,
             CatTypeOfNumberAndNumberingPlanIdentifier TONandNPI,
             BYTE DiallingNumberLength, const void *DiallingNumber,
             BYTE BearerDescription[3], UINT16 *BufferSize, CatDevice *ChannelIdentifier);
```

6.6.11.1.1 Normal execution

CRRN1: Send an OPEN CHANNEL proactive command to the terminal that opens a circuit switched channel.

6.6.11.1.2 Parameter errors

CRRP1: If DiallingNumberLength is zero or Dialling Number is null, the status word CAT_REQUIRED_VALUES_MISSING is returned.

6.6.11.1.3 Context errors

6.6.11.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatOpenCSChannel.tst
Test Application:	API_ CatOpenCSChannel.c

6.6.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	CatGeneralResult rv;
	Char *userConfirm = “Opening channel?”;
	BYTE number[] = {1,6,1,7,9,6,4,1,7,9,8};
	BYTE bearer[] = {1,2,3};
	UINT16 bufferSize;
	1- rv = CatOpenCSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL, TON_INTERNATIONAL_AND_NPI_TELEPHONE, sizeof(number), number,
	bearer &bufferSize, DEVICE_CHANNEL_1);
		1 – rv = CAT_SUCCESS	1 – OPEN CHANNEL proactive command

2	**DiallingNumber arguments null**
	CatGeneralResult rv;
	Char *userConfirm = “Opening channel?”;
	BYTE number[] = {1,6,1,7,9,6,4,1,7,9,8};
	BYTE bearer[] = {1,2,3};
	UINT16 bufferSize;
	1- rv = CatOpenCSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL, TON_INTERNATIONAL_AND_NPI_TELEPHONE, sizeof(number), NULL,
	bearer &bufferSize, DEVICE_CHANNEL_1);
		1 – rv = CAT_REQUIRED_VALUES_MISSING	

6.6.11.4 Test Coverage

CRR number	Test Case Number
N1	1
P1	2

6.6.12 CatOpenGPRSChannel

Test Area Reference: API_CatOpenGPRSChannel

6.6.12.1 Conformance Requirements

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

```
CatGeneralResult CatOpenGPRSChannel(CatOpenChannelOptions Options,
               BYTE UserConfirmationLength, const void *UserConfirmation,
               const CatIconIdentifier *UserConfirmationIconIdentifier,
               BYTE BearerDescription[8], UINT16 *BufferSize, CatDevice *ChannelIdentifier);
```

6.6.12.1.1 Normal execution

CRRN1: Send an OPEN CHANNEL proactive command with the provided data to the terminal that opens a GPRS channel.

6.6.12.1.2 Parameter errors

6.6.12.1.3 Context errors

6.6.12.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatOpenGPRSChannel.tst

Test Application: API_ CatOpenGPRSChannel.c

6.6.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatOpenGPRSChannel**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`BYTE bearer[8];`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
		1 – rv = CAT_SUCCESS	1 – OPEN CHANNEL proactive command

6.6.12.4 Test Coverage

CRR number	Test Case Number
N1	1

6.6.13 CatCloseChannel

Test Area Reference: API_CatCloseChannel

6.6.13.1 Conformance Requirements

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

```
CatGeneralResult CatCloseChannel(CatDevice ChannelIdentifier,
             BYTE TitleLength, const void *Title,
             const CatIconIdentifier *IconIdentifier);
```

6.6.13.1.1 Normal execution

CRRN1: Send a CLOSE CHANNEL proactive command with the provided data to the terminal.

6.6.13.1.2 Parameter errors

6.6.13.1.3 Context errors

6.6.13.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatCloseChannel.tst

Test Application: API_ CatCloseChannel.c

6.6.13.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatCloseChannel**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Bye-bye”;`
	`BYTE bearer[8];`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatCloseChannel(channel, strlen(title), title, NULL);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – CLOSE CHANNEL proactive command

6.6.13.4 Test Coverage

CRR number	Test Case Number
N1	1

6.6.14 CatReceiveData

Test Area Reference: API_CatReceiveData

6.6.14.1 Conformance Requirements

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

```
CatGeneralResult CatReceiveData(CatDevice ChannelIdentifier,
            BYTE TitleLength, const void *Title, BYTE RequestedChannelDataLength,
            const CatIconIdentifier *IconIdentifier,
            BYTE *ChannelData, BYTE *NumChannelBytesRead, BYTE *NumChannelBytesLeft);
```

6.6.14.1.1 Normal execution

CRRN1: Send a RECEIVE DATA proactive command to the terminal with the provided data.

CRRN2: If ChannelData or NumChannelBytesRead is null, the number of bytes available for reception on the channel is returned at NumChannelBytesLeft.

CRRN3: If NumChannelBytesLeft is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.14.1.2 Parameter errors

6.6.14.1.3 Context errors

6.6.14.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatReceiveData.tst

Test Application: API_ CatReceiveData.c

6.6.14.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Incoming”;`
	`BYTE bearer[8], actualLength, remainingLength;`
	`BYTE incomingData[36];`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatReceiveData(channel, strlen(title), title, sizeof(incomingData), NULL, incomingData, &actualLength, &remainingLength);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command
			2 – RECEIVE DATA proactive comand

2	**ChannelData null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Incoming”;`
	`BYTE bearer[8], actualLength, remainingLength;`
	`BYTE incomingData[36];`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatReceiveData(channel, strlen(title), title, sizeof(incomingData), NULL, NULL, &actualLength, &remainingLength);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command

3	**NumChannelBytesLeft null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Incoming”;`
	`BYTE bearer[8], actualLength, remainingLength;`
	`BYTE incomingData[36];`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatReceiveData(channel, strlen(title), title, sizeof(incomingData), NULL, incomingData, &actualLength, NULL);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command

6.6.14.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3

6.6.15 CatSendData

Test Area Reference: API_CatSendData

6.6.15.1 Conformance Requirements

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

```
CatGeneralResult CatSendData(CatDevice ChannelIdentifier,
         CatSendDataOptions Options, BYTE TitleLength, const void *Title,
         BYTE ChannelDataLength, const void *ChannelData,
         const CatIconIdentifier *IconIdentifier, BYTE *ActualBytesSent);
```

6.6.15.1.1 Normal execution

CRRN1: Send a SEND DATA proactive command to the terminal with the provided data; if ActualBytesSent is non-null, the number of bytes actually sent is returnedin ActualBytesSent.

CRRN2: If ChannelDataLength is zero or ChannelData is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.15.1.2 Parameter errors

6.6.15.1.3 Context errors

6.6.15.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendData.tst

Test Application: API_ CatSendData.c

6.6.15.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**ActualBytesSent non-null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Outgoing”;`
	`BYTE bearer[8], actualLength, remainingLength;`
	`BYTE outgoingData[] = {1,2,3,4,5,6,7,8};`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatSendData(channel, SEND_DATA_IMMEDIATELY, strlen(title), title,sizeof(outgoingData), outgoingData, NULL, &actualLength);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command
			2 – SEND DATA proactive comand

2	**ActualBytesSent null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Outgoing”;`
	`BYTE bearer[8], actualLength, remainingLength;`
	`BYTE outgoingData[] = {1,2,3,4,5,6,7,8};`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatSendData(channel,`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command

Id	Description	API Expectation	APDU Expectation

	`SEND_DATA_IMMEDIATELY, strlen(title),`
	`title,sizeof(outgoingData),outgoingData,`
	`NULL, NULL);`		

3	**ChannelData null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`char *title = “Outgoing”;`
	`BYTE bearer[8], actualLength,`
	`remainingLength;`
	`BYTE outgoingData[] = {1,2,3,4,5,6,7,8};`
	`UINT16 bufferSize;`
	`CatDevice channel;`
	`1 – rv =`
	`CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm,`
	`NULL,  bearer, &bufferSize, &channel);`
	`2 - CatSendData(channel,`
	`SEND_DATA_IMMEDIATELY, strlen(title),`
	`title,sizeof(outgoingData), NULL, NULL,`
	`&actualLength);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command

6.6.15.4 Test Coverage

CRR number	Test Case Number
N1	1,2
N2	3

6.6.16 CatGetChannelStatus

Test Area Reference: API_CatGetChannelStatus

6.6.16.1 Conformance Requirements

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

```
CatGeneralResult CatGetChannelStatus(CatDevice ChannelIdentifier, void *ChannelStatus);
```

6.6.16.1.1 Normal execution

CRRN1: Send a GET CHANNEL STATUS proactive command to the terminal; if ChannelStatus is non-null return the channel status at ChannelStatus.

CRRN2: If ChannelStatus is null, invocation is a NOP that returns CAT_SUCCESS.

6.6.16.1.2 Parameter errors

6.6.16.1.3 Context errors

6.6.16.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetChannelStatus.tst

Test Application: API_ CatGetChannelStatus.c

6.6.16.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**ChannelStatus non-null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`BYTE bearer[8], channelStatus[16];`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatGetChannelStatus(channel, channelStatus);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command
			2 – GET CHANNEL STATUS proactive commannd

2	**ChannelStatus null**
	`CatGeneralResult rv;`
	`char *userConfirm = “Opening GPRS?”;`
	`BYTE bearer[8], channelStatus[16];`
	`CatDevice channel;`
	`1 – rv = CatOpenGPRSChannel(IMMEDIATE_LINK_ESTABLISHMENT, strlen(userConfirm), userConfirm, NULL,  bearer, &bufferSize, &channel);`
	`2 - CatGetChannelStatus(channel, NULL);`
		1 – rv = CAT_SUCCESS
		2 – rv = CAT_SUCCESS
			1 – OPEN CHANNEL proactive command

6.6.16.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.17 CatServiceSearch

Test Area Reference: API_CatServiceSearch

6.6.17.1 Conformance Requirements

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

```
CatGeneralResult CatServiceSearch(CatBearer BearerId,
            BYTE AttributeLength, void *Attributes, void *ServiceAvailability);
```

6.6.17.1.1 Normal execution

CRRN1: Send a SERVICE SEARCH proactive command to the terminal with the provided data.

CRRN2: If AttributeLength is zero, Attributes is null or ServiceAvailability is null, invocation is a NOP that returns CAT_SUCCESS.

6.6.17.1.2 Parameter errors

6.6.17.1.3 Context errors

6.6.17.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatServiceSearch.tst

Test Application: API_ CatServiceSearch.c

6.6.17.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResult rv;`
	`BYTE attr[] = {1,2,3,4}, service[16];`
	`1 – rv = CatServiceSearch(BEARER_SMS, sizeof(attr), attr, service);`
		1 – rv = CAT_SUCCESS	1 – SERVICE SEARCH proactive command

2	**Attributes null**
	`CatGeneralResult rv;`
	`BYTE attr[] = {1,2,3,4}, service[16];`
	`1 – rv = CatServiceSearch(BEARER_SMS, sizeof(attr), NULL, service);;`
		1 – rv = CAT_SUCCESS	

3	**ServiceAvailbility null**
	`CatGeneralResult rv;`
	`BYTE attr[] = {1,2,3,4}, service[16];`
	`1 – rv = CatServiceSearch(BEARER_SMS, sizeof(attr), attr, NULL);`
		1 – rv = CAT_SUCCESS	

6.6.17.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2, 3

6.6.18 CatGetServiceInformation

Test Area Reference: API_CatGetServiceInformation

6.6.18.1 Conformance Requirements

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

```
CatGeneralResult CatGetServiceInformation(BYTE TitleLength, const BYTE *Title,
            const catIconIdentifier *IconIdentifier,
            BYTE BearerId, BYTE *AttributeLength, void *Attributes,
            void *ServiceInformation);
```

6.6.18.1.1 Normal execution

CRRN1: Send a GET SERVICE INFORMATION proactive command to the terminal with the provided data.

CRNN2: If AttributesLength, Attributes or ServiceInformation is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.18.1.2 Parameter errors

6.6.18.1.3 Context errors

6.6.18.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetServiceInformation.tst

Test Application: API_ CatGetServiceInformation.c

6.6.18.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResult rv;`
	`Char *title=”Title”;`
	`BYTE attr[] = {1,2,3,4}, service[16];`
	`1 – rv =`
	`CatGetServiceInformation(strlen(title),`
	`title,NULL, BEARER_SMS, sizeof(attr),`
	`attr, service);`
		1 – rv = CAT_SUCCESS	1 – SERVICE INFORMATION proactive command

2	**ServiceInformation nul**
	`CatGeneralResult rv;`
	`Char *title=”Title”;`
	`BYTE attr[] = {1,2,3,4}, service[16];`
	`1 – rv =`
	`CatGetServiceInformation(strlen(title),`
	`title,NULL, BEARER_SMS, sizeof(attr),`
	`attr, NULL);`
		1 –rv = CAT_SUCCESS	

6.6.18.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.19 CatDeclareService

Test Area Reference: API_CatDeclareService

6.6.19.1 Conformance Requirements

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

```
CatGeneralResult CatDeclareService(BYTE BearerId, BYTE ServiceId,
            CatTransportProtocol TransportProtocol,
            WORD *PortNumber, BYTE ServiceRecordLength, void *ServiceRecord);
```

6.6.19.1.1 Normal execution

CRRN1: Send a DECLARE SERVICE proactive command to the terminal with the provided data.

CRRN2: If ServiceRecordLength is zero or PortNumber or ServiceRecord is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.19.1.2 Parameter errors

6.6.19.1.3 Context errors

6.6.19.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatDeclareService.tst
Test Application:	API_ CatDeclareService.c

6.6.19.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResult rv;`
	`WORD port;`
	`BYTE service[32];`
	`1 – rv = CatDeclareService(BEARER_SMS, 1, TRANSPORT_TCP, &port, sizeof(service), service);`
		1 – rv = CAT_SUCCESS	1 – DECLARE SERVICE proactive command

2	**ServiceRecord null**
	`CatGeneralResult rv;`
	`WORD port;`
	`BYTE service[32];`
	`1 – rv = CatDeclareService(BEARER_SMS, 1, TRANSPORT_TCP, &port, sizeof(service), NULL);`
		1 – rv = CAT_SUCCESS	

6.6.19.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.6.20 CatRunATCommand

Test Area Reference: API_CatRunATCommand

6.6.20.1 Conformance Requirements

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

```
CatGeneralResult CatRunATCommand(BYTE TitleLength, const void *Title,
            BYTE CommandLength, const void *Command,
            const CatIconIdentifier *IconIdentifier,
            void *Response, BYTE *ResponseLength);
```

6.6.20.1.1 Normal execution

CRRN1: Send a RUN AT COMMAND proactive command to the terminal with the provided data; *ResponseLength bytes of the result of the command are returned in Response if both are non-null.

CRNN2: If ResponseLength or Response is null, no results are returned.

CRNN3: If CommandLength is zero or Command is null, invocation is a NOP that returns CAT_SUCCESS.

6.6.20.1.2 Parameter errors

6.6.20.1.3 Context errors

6.6.20.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatRunATCommand.tst

Test Application: API_ CatRunATCommand.c

6.6.20.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE AtCmd[] = {‘A’, ‘T’};`
	`BYTE resp[32], respLen;`
	`1 – rv = CatRunATCommand(strlen(title), title, sizeof(AtCmd), AtCmd, NULL, resp, &respLen);`
		1 – rv = CAT_SUCCESS, respLen = 2 and “OK” in resp	1 – RUN AT COMMAND proactive command

2	**Response null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE AtCmd[] = {‘A’, ‘T’};`
	`BYTE resp[32], respLen;`
	`1 – rv = CatRunATCommand(strlen(title), title, sizeof(AtCmd), AtCmd, NULL, NULL, &respLen);`
		rv = CAT_SUCCESS, respLen = 2	1 – RUN AT COMMAND proactive command

3	**Command null**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`BYTE AtCmd[] = {‘A’, ‘T’};`
	`BYTE resp[32], respLen;`
	`1 – rv = CatRunATCommand(strlen(title), title, sizeof(AtCmd), NULL, NULL, resp, &respLen);`
		1 – rv = CAT_SUCCESS	

6.6.20.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3

6.6.21 CatSendDTMFCommad

Test Area Reference: API_CatSendDTMFCommand

6.6.21.1 Conformance Requirements

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

```
CatGeneralResult CatSendDTMFCommand(BYTE TitleLength, const void *Title,
                BYTE DTMFCodeLength, const void *DTMFCode,
                const CatIconIdentifier *IconIdentifier);
```

6.6.21.1.1 Normal execution

CRRN1: Send a SEND DTMF COMMAND proactive command to the terminal with the provided data.

CRRN2: If DTMFCodeLength is zero or DTMFCode is null, the invocation is a NOP that returns CAT_SUCCESS.

6.6.21.1.2 Parameter errors

6.6.21.1.3 Context errors

6.6.21.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatSendDTMFCommand.tst
Test Application:	API_ CatSendDTMFCommand.c

6.6.21.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`CatGeneralResponse rv;`
	`char *title = “Title”;`
	`BYTE DTMF[] = {1,2,3,4};`
	`1 – rv = CatSendDTMFCommand(strlen(title), title, sizeof(DTMF), DTMF, NULL);`
		1 – rv = CAT_SUCCESS	1 – SEND DTMF proactive command

2	**DTMFCode null**
	`CatGeneralResponse rv;`
	`char *title = “Title”;`
	`BYTE DTMF[] = {1,2,3,4};`
	`1 – rv = CatSendDTMFCommand(strlen(title), title, sizeof(DTMF), NULL, NULL);`
		1 – rv = CAT_SUCCESS	

6.6.21.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.7 Toolkit Application

6.7.1 main

Test Area Reference: API_main

6.7.1.1 Conformance Requirements

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

```
void main(void)
```

6.7.1.1.1 Normal execution

CRRN1: Load and execute the null application: main(){}.

6.7.1.1.2 Parameter errors

6.7.1.1.3 Context errors

6.7.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_main.tst

Test Application: API_ main.c

6.7.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**main(){}**
	`main(){CatExit();}`
		1 – No return	

6.7.1.4 Test Coverage

CRR number	Test Case Number
N1	1

6.7.2 CatGetFrameworkEvent

Test Area Reference: API_CatGetFrameworkEvent

6.7.2.1 Conformance Requirements

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

```
CatFrameworkEventType CatGetFrameworkEvent(void);
```

6.7.2.1.1 Normal execution

CRRN1: Return the current framework event.

6.7.2.1.2 Parameter errors

6.7.2.1.3 Context errors

6.7.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetFrameworkEvent.tst

Test Application: API_ CatGetFrameworkEvent.c

6.7.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**CatGetFrameworkEvent** `1 - CatGetFrameworkEvent();`	1 –Returns a CatFrameworkEventType	

6.7.2.4 Test Coverage

CRR number	Test Case Number
N1	1

6.7.3 CatExit

Test Area Reference: API_CatExit

6.7.3.1 Conformance Requirements

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

```
void CatExit(void);
```

6.7.3.1.1 Normal execution

CRRN1: Load and execute the exit application: main(){catExit();}

6.7.3.1.2 Parameter errors

6.7.3.1.3 Context errors

6.7.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatExit.tst

Test Application: API_ CatExit.c

6.7.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**main(){CatExit();}**
	1 - main(){CatExit();}
		1 – No return.	

6.7.3.4 Test Coverage

CRR number	Test Case Number
N1	1

6.8 Miscellaneous

6.8.1 CatGetTerminalProfile

Test Area Reference: API_CatGetTerminalProfile

6.8.1.1 Conformance Requirements

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

```
void CatGetTerminalProfile(BYTE *ProfileOutLength, BYTE *Profile);
```

6.8.1.1.1 Normal execution

CRRN1: Retrieve the current terminal profile.

CRRN2: If ProfileOutLength or Profile is null, the invocation is a NOP.

6.8.1.1.2 Parameter errors

6.8.1.1.3 Context errors

6.8.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatGetTerminalProfile.tst
Test Application:	API_ CatGetTerminalProfile.c

6.8.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All arguments non-null**
	`BYTE profile[32], profileLen;`
	`1 - CatGetTerminalProfile(&profileLen, profile);`
		1 –Terminal profile returned in profile.	

2	**Profile null**
	`BYTE profile[32], profileLen;`
	`1 - CatGetTerminalProfile(&profileLen, NULL);`
		1 –No return	

6.8.1.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.8.2 CatMoreTime

Test Area Reference: API_CatMoreTime

6.8.2.1 Conformance Requirements

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

```
CatGeneralResult CatMoreTime(void);
```

6.8.2.1.1 Normal execution

CRRN1: Issue a MORE TIME request to the terminal.

6.8.2.1.2 Parameter errors

6.8.2.1.3 Context errors

6.8.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatMoreTime.tst

Test Application: API_ CatMoreTime.c

6.8.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatMoreTime**
	`CatGeneralResult rv;`
	`1 – rv = CatMoreTime();`
		1 –rv = CAT_SUCCESS	MORE TIME protocol byte

6.8.2.4 Test Coverage

CRR number	Test Case Number
N1	1

6.8.3 CatPollingOff

Test Area Reference: API_CatPollingOff

6.8.3.1 Conformance Requirements

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

```
CatGeneralResult CatPollingOff(void);
```

6.8.3.1.1 Normal execution

CRRN1: Send the POLLING OFF proactive command to the terminal.

6.8.3.1.2 Parameter errors

6.8.3.1.3 Context errors

6.8.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPollingOff.tst

Test Application: API_ CatPollingOff.c

6.8.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatPollingOff**
	`CatGeneralResult rv;`
	`1 – rv = CatPollingOff();`
		1 –rv = CAT_SUCCESS	1 – POLLING OFF proactive command

6.8.3.4 Test Coverage

CRR number	Test Case Number
N1	1

6.8.4 CatPollInterval

Test Area Reference: API_CatPollInterval

6.8.4.1 Conformance Requirements

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

```
CatGeneralResult CatPollInterval(CatTimeUnit Unit, BYTE Interval,
             CatTimeInterval *ActualIntervalOut);
```

6.8.4.1.1 Normal execution

CRRN1: Send the POLL INTERVAL proactive command to the terminal.

CRRN2: If ActualIntervalOut is null, the actual interval set by the terminal is not returned.

6.8.4.1.2 Parameter errors

6.8.4.1.3 Context errors

6.8.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPollInterval.tst

Test Application: API_ CatPollInterval.c

6.8.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**ActualIntervalOut non-null**
	`CatGeneralResult rv;`
	`CatTimeInternval interval;`
	`1 – rv = CatPollInterval (GSM_SECONDS, 1, &interval);`
		1 – rv = CAT_SUCCESS	1 – POLL INTERVAL proactive command

2	**ActualIntervalOut null**
	`CatGeneralResult rv;`
	`1 – rv = CatPollInterval (GSM_SECONDS, 1, NULL);`
		1 – rv = CAT_SUCCESS	1 – POLL INTERVAL proactive command

6.8.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.8.5 CatRefresh

Test Area Reference: API_CatRefreshWithFileList

6.8.5.1 Conformance Requirements

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

```
CatGeneralResult CatRefreshWithFileList(CatRefreshOptions Options,
                    BYTE FileListLength, const void *FileList);
```

6.8.5.1.1 Normal execution

CRRN1: Send a REFRESH proactive command to the terminal with the provided data.

CRRN2: If FileListLength is zero or FileList is null, send a REFRESH proactive command that refreshes all files.

6.8.5.1.2 Parameter errors

6.8.5.1.3 Context errors

6.8.5.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatRefresh.tst
Test Application:	API_ CatRefresh.c
Test Script:	API_CatRefreshWithFileList.tst
Test Application:	API_ CatRefreshWithFileList.c

6.8.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**All argument non-null**
	`CatGeneralResult rv;`
	`BYTE fileList[] = {0x26, 0x01};`
	`1 – rv = CatRefreshWithFileList(REFRESH_FILE_CHANGE_NOTIFICATION,sizeof(fileList), fileList);`
		1 – rv = CAT_SUCCESS	1 – REFRESH proactive command

2	**FileList null**
	`CatGeneralResult rv;`
	`1 – rv = CatRefreshWithFileList(REFRESH_FILE_CHANGE`
		1 –rv = CAT_SUCCESS	1 – REFRESH proactive command

Id	Description	API Expectation	APDU Expectation
	`_NOTIFICATION, 0, NULL);`		

6.8.5.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.8.6 CatLanguageNotification

Test Area Reference: API_CatLanguageNotification

6.8.6.1 Conformance Requirements

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

```
void CatLanguageNotification(CatLanguageNotificationOptions Options,
                             const void *Language);
```

6.8.6.1.1 Normal execution

CRRN1: Send a LANGUAGE NOTIFICATION proactive command to the terminal with the provided data.

CRRN2: If Langauge is null, the invocation is a NOP.

6.8.6.1.2 Parameter errors

6.8.6.1.3 Context errors

6.8.6.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatLanguagenotification.tst

Test Application: API_ CatLanguagenotification.c

6.8.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Language non-null**
	`WORD language = 2;`
	`1 - CatLanguageNotification(LANGUAGE_NON_SPECIFIC_NOTIFICATION, &language);`
		1 – No return	1 – LANGUAGE NOTIFICATION proactive command

2	**Language null**
	`WORD language = 2;`
	`1 - CatLanguageNotification(LANGUAGE_NON_SPECIFIC_NOTIFICATION, NULL);`
		1 – No return	

6.8.6.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.8.7 CatLaunchBrowser

Test Area Reference: API_CatLaunchBrowser

6.8.7.1 Conformance Requirements

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

```
CatGeneralResult CatLaunchBrowser(CatLaunchBrowserOptions Options,
              BYTE TitleLength, const void *Title,
              BYTE URLLength, const void *URL,
              const CatIconIdentifier *IconIdentifier);
```

6.8.7.1.1 Normal execution

CRRN1: Send a LAUNCH BROWSER proactive command to the terminal with the provided data.

6.8.7.1.2 Parameter errors

6.8.7.1.3 Context errors

6.8.7.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatLaunchBrowser.tst

Test Application: API_ CatLaunchBrowser.c

6.8.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatLaunchBrowser**
	`CatGeneralResult rv;`
	`char *title = “Title”;`
	`char *URL = http://www.etsi.org;`
	`1 – rv = CatLaunchBrowser( LAUNCH_BROWSER_IF_NOT_ALREADY_LAUNCHED, strlen(title), const void *Title, strlen(URL), URL, NULL);`
		1 – rv = CAT_SUCCESS	1 – LAUNCH BROWSER proactive command

6.8.7.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9 Low-Level Interface

6.9.1 CatResetBuffer

Test Area Reference: API_CatResetBuffer

6.9.1.1 Conformance Requirements

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

```
void CatResetBuffer(void);
```

6.9.1.1.1 Normal execution

CRRN1: Clear the buffer that is used to construct proactive commands.

6.9.1.1.2 Parameter errors

6.9.1.1.3 Context errors

6.9.1.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatResetBuffer.tst

Test Application: API_ CatResetBuffer.c

6.9.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	`BYTE *p;`
	`1 - CatResetBuffer();`
	`2 - CatPutByte(0x77);`
	`3 – p = CatGetData(1);`
		1 – No return
		2 – No return
		3 - *p = 0x77	

6.9.1.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.2 CatStartProactiveCommand

Test Area Reference: API_CatStartProactiveCommand

6.9.2.1 Conformance Requirements

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

```
void CatStartProactiveCommand(BYTE Command, BYTE Options, BYTE To);
```

6.9.2.1.1 Normal execution

CRRN1: Initialize the constructed proactive command buffer to contain the proactive command header that includes the provided data.

6.9.2.1.2 Parameter errors

6.9.2.1.3 Context errors

6.9.2.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatStartProactiveCommand.tst

Test Application: API_ CatStartProactiveCommand.c

6.9.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	
1	**CatStartProactiveCommand** `1 - CatStartProactiveCommand(0x21, 0x00, 0x81);`	1 – No return	

6.9.2.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.3 CatSendProactiveCommand

Test Area Reference: API_CatSendProactiveCommand

6.9.3.1 Conformance Requirements

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

```
CatGeneralResult CatSendProactiveCommand(BYTE *Length);
```

6.9.3.1.1 Normal execution

CRRN1: Send the proactive command in the constructed proactive command to the terminal and return the result of the terminal's execution of the command.

CRRN2: If Length is null then the invocation is for the general result only.

6.9.3.1.2 Parameter errors

6.9.3.1.3 Context errors

6.9.3.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendProactiveCommand.tst

Test Application: API_ CatSendProactiveCommand.c

6.9.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Length non-null**
	`CatGeneralResult rv;`
	`BYTE len;`
	`1 – rv = CatSendProactiveCommand(&len);`
		1 – rv = CAT_SUCCESS	1 – Constructed proactive command

2	**Length null**
	`CatGeneralResult rv;`
	`1 – rv = CatSendProactiveCommand(NULL);`
		1 – rv = CAT_SUCCESS	1 – Constructed proactive command

6.9.3.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.9.4 CatOpenEnvelope

Test Area Reference: API_CatOpenEnvelope

6.9.4.1 Conformance Requirements

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

```
CatEnvelopeTagType CatOpenEnvelope(BYTE *Length);
```

6.9.4.1.1 Normal execution

CRRN1: Return the tag and the length of the envelope command that arrived for the application.

CRRN2: If Length is null, on the tag is returned.

6.9.4.1.2 Parameter errors

6.9.4.1.3 Context errors

6.9.4.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatOpenEnvelope.tst
Test Application:	API_ CatOpenEnvelope.c

6.9.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Length non-null**
	`CatEnvelopeTagType rv;`
	`BYTE len;`
	`1 – rv = CatOpenEnvelope(&len);`
		1 – rv = EVENT_DOWNLOAD_TAG	

2	**Length null**
	`CatEnvelopeTagType rv;`
	`1 – rv = CatOpenEnvelope(NULL);`
		1 – rv = EVENT_DOWNLOAD_TAG	

6.9.4.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.9.5 CatSendEnvelopeResponse

Test Area Reference: API_CatSendEnvelopeResponse

6.9.5.1 Conformance Requirements

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

```
void CatSendEnvelopeResponse(void);
```

6.9.5.1.1 Normal execution

CRRN1: Send the contents of the constructed proactive command buffer to the terminal as a successful response to a envelope command.

6.9.5.1.2 Parameter errors

6.9.5.1.3 Context errors

6.9.5.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatSendEnvelopeResponse.tst

Test Application: API_ CatSendEnvelopeResponse.c

6.9.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatSentEnvelopeResponse**
	`1 – CatResetBuffer();`
	`2 – CatPutByte(0x77);`
	`3 - CatSendEnvelopeResponse();`
		1 – No return
		2 – No return
		3 – No return
			1 – ‘77’ ‘90’ ‘00’ response

6.9.5.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.6 CatSendEnvelopeErrorResponse

Test Area Reference: API_CatSendEnvelopeErrorResponse

6.9.6.1 Conformance Requirements

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

```
void CatSendEnvelopeErrorResponse(void);
```

6.9.6.1.1 Normal execution

CRRN1: Send the contents of the constructed proactive command buffer to the terminal as an unsuccessful response to a envelope command.

6.9.6.1.2 Parameter errors

6.9.6.1.3 Context errors

6.9.6.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script:	API_CatSendEnvelopeErrorResponse.tst
Test Application:	API_ CatSendEnvelopeErrorResponse.c

6.9.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatSendEnvelopeErrorRespose**
	`BYTE data[] = {0x6D, 0x00};`
	`1 – CatResetBuffer();`
	`2 – CatPutDate(sizeof(data), data);`
	`3 - CatSendEnvelopeErrorResponse(void);`
		1 – No return
		2 – No return
		3 – No return
			3 – ‘6D’ ‘00’ response

6.9.6.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.7 CatPutData

Test Area Reference: API_CatPutData

6.9.7.1 Conformance Requirements

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

```
void CatPutData(BYTE Length, const void *Data);
```

6.9.7.1.1 Normal execution

CRRN1: Append the provided data to the data in the constructed proactive command buffer.

CRRN2: If Length is zero or Data is null, the invocation is a NOP.

6.9.7.1.2 Parameter errors

6.9.7.1.3 Context errors

6.9.7.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutData.tst

Test Application: API_ CatPutData.c

6.9.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Both arguments non-null**
	`BYTE data[] = {1,2,3,4,5,6,7,8}, *p, *q;`
	`1 - CatResetBuffer();`
	`2 - CatPutData(sizeof(data), data);`
	`3 – q = CatGetData(4);`
	`4 – p = CatGetData(1);`
		1 – No return value.
		2 – No return value.
		3 – Return q
		4 – Return p = q+4 and *p = 5	

2	**Length zero and data null**
	`BYTE data[] = {1,2,3,4};`
	`1 – CatResetBuffer();`
	`2 – CatPutByteData(0, NULL);`
		1 – No return value.
		2 – No return value.	

6.9.7.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.9.8 CatPutByte

Test Area Reference: API_CatPutByte

6.9.8.1 Conformance Requirements

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

```
void CatPutByte(BYTE Data)
```

6.9.8.1.1 Normal execution

CRRN1: Append the provided byte to the data in the constructed proactive command buffer.

6.9.8.1.2 Parameter errors

6.9.8.1.3 Context errors

6.9.8.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutByte.tst

Test Application: API_ CatPutByte.c

6.9.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatPutByte**
	`BYTE *p;`
	`1 - CatResetBuffer();`
	`2 - CatPutByte(0x77);`
	`3 – p = CatGetData(1);`
		1 – No return
		2 – No return
		3 - *p = 0x77	

6.9.8.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.9 CatPutTLV

Test Area Reference: API_CatPutTLV

6.9.9.1 Conformance Requirements

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

```
void CatPutTLV(BYTE Tag, BYTE Length, const void *Value);
```

6.9.9.1.1 Normal execution

CRRN1: Append the provided TLV to the data in the constructed proactive command buffer.

CRRN2: If Value is null, the invocation is a NOP.

6.9.9.1.2 Parameter errors

6.9.9.1.3 Context errors

6.9.9.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutTLV.tst

Test Application: API_ CatPutTLV.c

6.9.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Value non-null**
	`BYTE data[] = {1,2,3,4,5,6,7,8}, *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutTLV(0x83, sizeof(data), data);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - *p = 0x83 and length = 8	

2	**Value null**
	`BYTE data[] = {1,2,3,4,5,6,7,8}, *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutTLV(0x83, sizeof(data), NULL);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - p = NULL and length = 0	

6.9.9.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.9.10 CatPutBytePrefixedTLV

Test Area Reference: API_CatPutBytePrefixedTLV

6.9.10.1 Conformance Requirements

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

```
void CatPutBytePrefixedTLV(BYTE Tag, BYTE Prefix, BYTE Length, const void *Value);
```

6.9.10.1.1 Normal execution

CRRN1: Append the provided data to the data in the constructed proactive command buffer.

CRRN2: If Value is null the invocation is a NOP.

6.9.10.1.2 Parameter errors

6.9.10.1.3 Context errors

6.9.10.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutBytePrefixedTLV.tst

Test Application: API_ CatPutBytePrefixedTLV.c

6.9.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Value non-null**
	`BYTE data[] = {1,2,3,4,5,6,7,8}, *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutBytePrefixedTLV(0x83, 0x77, sizeof(data), data);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - *(p+2) = 0x77 and length = 9	

2	**Value null**
	`BYTE data[] = {1,2,3,4,5,6,7,8}, *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutBytePrefixedTLV(0x83, 0x77, sizeof(data), NULL);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - p = NULL and length = 0	

6.9.10.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2

6.9.11 CatPutOneByteTLV

Test Area Reference: API_CatPutOneByteTLV

6.9.11.1 Conformance Requirements

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

```
void CatPutOneByteTLV(BYTE Tag, BYTE Value);
```

6.9.11.1.1 Normal execution

CRRN1: Append the provided data to the data in the constructed proactive command buffer.

6.9.11.1.2 Parameter errors

6.9.11.1.3 Context errors

6.9.11.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutOneByteTLV.tst

Test Application: API_ CatPutOneByteTLV.c

6.9.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatPutOneByteTLV**
	`BYTE *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutOneByteTLV(0x83, 0x77);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - *(p+2) = 0x77 and length = 3	

6.9.11.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.12 CatPutTwoByteTLV

Test Area Reference: API_CatPutTwoByteTLV

6.9.12.1 Conformance Requirements

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

```
Void CatPutTwoByteTLV(BYTE Tag, BYTE Value1, BYTE Value2);
```

6.9.12.1.1 Normal execution

CRRN1: Append the provided data to the data in the constructed proactive command buffer.

6.9.12.1.2 Parameter errors

6.9.12.1.3 Context errors

6.9.12.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatPutTwoByteTLV.tst

Test Application: API_ CatPutTwoByteTLV.c

6.9.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatPutTwoByteTLV**
	`BYTE *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutTwoByteTLV(0x83, 0x77, 0x88);`
	`3 – p = CatFindNthTLV(0x83, 1, &length);`
		1 – No return
		2 – No return
		3 - *(p+2) = 0x77, *(p+3)= 0x88 and length = 2	

6.9.12.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.13 CatGetByte

Test Area Reference: API_CatGetByte

6.9.13.1 Conformance Requirements

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

```
BYTE CatGetByte(void);
```

6.9.13.1.1 Normal execution

CRRN1: Return the byte at the current buffer pointer and advance the pointer by one byte.

6.9.13.1.2 Parameter errors

6.9.13.1.3 Context errors

6.9.13.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetByte.tst

Test Application: API_ CatGetByte.c

6.9.13.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatGetByte**
	`1 - CatResetBuffer();`
	`2 - CatPutOneByteTLV(0x83, 0x77);`
	`3 – CatGetByte()`
	`4 – CatGetByte()`
		1 – No return
		2 – No return
		3 – Return ‘83’
		4 – Return ‘01’	

6.9.13.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.14 CatGetData

Test Area Reference: API_CatGetData

6.9.14.1 Conformance Requirements

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

```
const void *CatGetData(BYTE Length);
```

6.9.14.1.1 Normal execution

CRRN1: Return the current data pointer and then increment the pointer by Length.

6.9.14.1.2 Parameter errors

6.9.14.1.3 Context errors

6.9.14.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatGetData.tst

Test Application: API_ CatGetData.c

6.9.14.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**CatGetData**
	`BYTE *p;`
	`DWORD length = 0;`
	`1 - CatResetBuffer();`
	`2 - CatPutTwoByteTLV(0x83, 0x77, 0x88);`
	`3 – p = CatGetData(0x02);`
	`4 – p = CatGetData(0x02);`
		1 – No return
		2 – No return
		3 – *p = 0x83, *(p+1) = 0x02
		4 – *p = 0x77, *(p+1) = 0x88	

6.9.14.4 Test Coverage

CRR number	Test Case Number
N1	1

6.9.15 CatFindNthTLV

Test Area Reference: API_CatFindNthTLV

6.9.15.1 Conformance Requirements

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

```
const void *CatFindNthTLV(BYTE Tag, BYTE Occurrence, BYTE *Length);
```

6.9.15.1.1 Normal execution

CRRN1: Return a pointer to the Occurance appearance of a TLV with the provided Tag in the data buffer and the length of this TLV at Length.

CRRN2: If there are less than Occurance appearances of TLVs with the provided Tag, return null.

CRRN3: If Length is null, do not return the Length of the TLV if one is found.

6.9.15.1.2 Parameter errors

6.9.15.1.3 Context errors

6.9.15.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatFindNthTLV.tst

Test Application: API_ CatFindNthTLV.c

6.9.15.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Occurance appearances of the tag**
	`BYTE *p;`
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatResetBuffer();`
	`2 - CatPutData(sizeof(data), data);`
	`3 – p = CatFindNthTLV(0x81, 2, &length);`
		1 – No return
		2 – No return
		3 – *p = 0x81, length = 0x02	

2	**Less than occurance appearances of the tag**
	`BYTE *p;`
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatResetBuffer();`
	`2 - CatPutData(sizeof(data), data);`
	` 3 – p = CatFindNthTLV(0x81, 3, &length);`
		1 – No return
		2 – No return
		3 – p = NULL, length = 0	

3	**Length null**
	`BYTE *p;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatResetBuffer();`
	`2 - CatPutData(sizeof(data), data);`
	`3 – p = CatFindNthTLV(0x81, 2, NULL);`
		1 – No return
		2 – No return
		3 – *p = 0x81	

6.9.15.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3

6.9.16 CatFindNthTLVInUserBuffer

Test Area Reference: API_CatFindNthTLVInUserBuffer

6.9.16.1 Conformance Requirements

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

```
const void *CatFindNthTLVInUserBuffer(BYTE BufferLen,
                       const void *Buffer, BYTE Tag,
                       BYTE Occurrence, BYTE *Length);
```

6.9.16.1.1 Normal execution

CRRN1: Return a pointer to the Occurance appearance of a TLV with the provided Tag in the provided buffer and the length of this TLV at Length.

CRRN2: If there are less than Occurance appearances of TLVs with the provided Tag, return null.

CRRN3: If Length is null, do not return the Length of the TLV if one is found.

CRNN4: If BufferLen is zero or Buffer is null, the invocation is a NOP that returns null.

6.9.16.1.2 Parameter errors

6.9.16.1.3 Context errors

6.9.16.2 Test Suite Files

Additional requirements for the UICC personalisation: None

Test Script: API_CatFindNthTLVInUserBuffer.tst

Test Application: API_ CatFindNthTLVInUserBuffer.c

6.9.16.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**UICC Initialisation**	Responses ignored.	

1	**Occurance appearances of the tag**
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatFindNthTLVInUserBuffer(sizeof(buffer), buffer, 0x81, 2, &length);`
		1 – Return value buffer+6 and length = 2	

2	**Less than occurance appearances of the tag**
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatFindNthTLVInUserBuffer(sizeof(buffer), buffer, 0x81, 3, &length);`
		1 – Return value 0 and length = 0	

3	**Length null**
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatFindNthTLVInUserBuffer(sizeof(buffer), buffer, 0x81, 2, NULL);`
		1 – Return value buffer+6	

4	**Buffer null**
	`DWORD length = 0;`
	`BYTE buffer[]= {0x80, 0x01, 0x00, 0x81, 0x01, 0x01, 0x81, 0x02, 0x01, 0x02};`
	`1 - CatFindNthTLVInUserBuffer(sizeof(buffer), NULL, 0x81, 3, &length);`
		1 – Return value 0 and length = 0	

6.9.16.4 Test Coverage

CRR number	Test Case Number
N1	1
N2	2
N3	3
N4	4

Annex A Script file syntax and format description (normative)

A.1 Syntax description

Following is a syntax description in BNF.

```
<statement list> ::=   [ <statement> \n] +

<statement> ::=        <simple> | <switch> | <blank line>

<simple> ::=           <reset> | <init> | <command> | <remark>

<reset> ::=            RST

<init> ::=             INI <hexdata>

<command> ::=          CMD <hexdata> [ <response> ] ( <status> )

<response> ::=         [ <hexdata> ]

<status> ::=           ( <hexdata> )

<remark> ::=           REM <text line>

<switch> ::=           SWI { [<labelled list>] + }

<labelled list> ::=    <label> : \n <statement list>
```

Description of syntax metalanguage :

\n	represents a linebreak
[x]	means x can appear optionally
[x] +	means 1 or more appearances of x
x | y	means x or y
[]{}:	(bold) these are characters that appear literally in the script files
<text line>	any character until the end of the line
<blank line>	a line containing no text is acceptable
<hexdata>	data written in hexadecimal, each byte separated from the following by a whitespace

Each simple statement beginning with 3 characters different than the ones defined indicates another tool command, and shall be ignored by the parser if not recognised.

`' ', '\t'` : Can be used as separator
A long statement can be broken into several lines by using the character '****' at the end of each line which is not the last one in the statement.

For more details refer to the examples in A.3.

A.2 Semantics

Following is the meaning of each of the statements :

CMD : Sends an APDU Command to the card, including (optionally) the expected response data and also (optionally) the expected status words SW1, SW2.

RST : Resets and powers on the card

INI : Performs the terminal profile with the following data. Afterwards, it shall perform all the fetch and terminal response commands until there is no proactive session in progress.

REM : Used for comments

SWI : Activates a switch condition. Every labelled list represents a list of statements to be executed, if the label matches the SW resulting from the previously executed command.

Evaluation of expected response and status in the case of a CMD:

`<response>` data within **[**…**]** has to be checked, it needs to be present for an outgoing command. Bytes written as XX shall not be checked by the APDU tool.

`<status>` status contained within **(**…**)** has to be checked; when several status are valid they shall be separated by commas. Bytes written as XX shall not be checked by the APDU tool.

A.3 Example

```
REM this is an example

RST
INI FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
REM Case 1 example
CMD A0 C2 00 00 00 (91 33 , 69 XX)

REM Case 2 example
CMD A0 B6 00 00 07 \
    [XX XX XX 55 55 XX 55] \
    (91 33 , 67 XX)


CMD A0 B6 00 00 07 \
    (91 33 , 67 XX)

CMD A0 C0 00 00 1F \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 28 A4 05 \
    02 0D CC CC CC CC CC CC CC CC CC CC CC CC CC ] \
    (90 00)

REM Case 3 example
CMD A0 C2 00 00 33 \
    D1 31 82 02 83 81 06 05 80 11 22 33 44 8B 24 40 \
    08 00 24 23 85 18 41 04 51 10 10 00 00 00 00 13 \
    02 70 00 00 0E 0D 00 00 00 00 28 A4 05 00 00 00 \
    00 00 00 \
    (90 00)

REM Case 4 example with switch statement
CMD 00 A4 04 00 10 \
    A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 02 \
    (61 XX, 6A 82)

SWI {
61 XX:
CMD 00 C0 00 00 14 \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 \
    02 02 CC CC] \
    (90 00)

CMD A0 A4 00 00 02 \
    3F 00

6A 82:
RST

}
```

A.4 Style and formatting

In order to show a common appearance all the scripts shall follow those format rules:

- start always with a 'RST' followed by an 'INI' command.
- The command, data to be checked and status to be checked shall be presented in the following order:

  CMD *COMMAND* [*EXPECTED DATA*] (*EXPECTED STATUS*)
- APDU shall be presented with command (CLA INS P1 P2 P3) in one line and data (if present) in next line grouped 16 bytes per line (see example above).
- The expected data (if present) shall be presented in 16 bytes groups per line (see example above).

Annex B Default Prepersonalisation (normative)

B.1 General Default Prepersonalisation

This table shows the default prepersonalisation, the file system and the files' content, that the test SIM cards shall contain unless otherwise stated.

Name	Identifier	Default Value	Special Features
EF_{ICCID}	2FE2	`0F FF FF FF FF FF FF FF FF FF`	This value is not compliant with GSM 11.11
EF_{IMSI}	6F07	`FF FF FF FF FF FF FF FF FF`	This value is not compliant with GSM 11.11
EF_{LP}	6F05	`01 FF FF FF`	
EF_{Kc}	6F20	`FF FF FF FF FF FF FF FF 07`	
$EF_{PLMNsel}$	6F30	`FF FF FF FF FF FF FF FF FF FF FF FF` `FF FF FF FF FF FF FF FF FF FF FF FF`	
EF_{HPLMN}	6F31	`05`	
EF_{ACMmax}	6F37	`00 00 00`	Access condition UPDATE: CHV1
EF_{SST}	6F38	`FF 3F C3 03 0C 00 FF 0F 00 33`	
EF_{ACM}	6F39	`00 00 00`	Access condition UPDATE: CHV1
EF_{PUCT}	6F41	`FF FF FF 00 00`	Access condition UPDATE: CHV1

EF_{BCCH}	6F74	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`	

EF_{ACC}	6F78	`00 00`	
EF_{FPLMN}	6F7B	`FF FF FF FF FF FF FF FF FF FF FF FF`	
EF_{LOCI}	6F7E	`FF FF FF FF 00 F0 00 00 00 FF 01`	
EF_{AD}	6FAD	`00 FF FF`	
EF_{Phase}	6FAE	`03`	

EF_{FDN}	6F3B	`Default value in all the records:`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`
			Records: 5

EF_{SMSP}	6F42	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF`
			Records: 1

EF_{LND}	6F44	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`
			Records: 1

EF_{SMSS}	6F43	`FF FF`	
EF_{SMS}	6F3C	1st record: 00 FF … FF(length 176) 2nd record:00 FF … FF(length 176) 3rd record: 00 FF … FF(length 176)	Records: 3

EF_{ADN}	6F3A	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`
			Records: 1

EF_{CCP}	6F3D	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF`	

EF_{MSISDN}	6F40	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`
			Records: 1

EF_{SDN}	6F49	`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF FF FF FF FF FF FF FF FF`
		`FF FF FF FF`
			Records: 1

EF_{SUME}	6F54	`85 0C 54 4F 4F 4C 4B 49 54 20 54 45`
		`53 54 FF FF FF FF`	

EF_{CBMI}	6F45	`FF FF`	
EF_{IM}	4F20	`FF FF FF FF FF FF FF FF FF FF FF`	

The default value for the CHV1 shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'disabled' during test Applications execution.

B.2 File System Access Default Prepersonalisation

B.2.1 $DF_{SIMTEST}$ (SIM Test)

Identifier: '3113'

B.2.2 EF_{TNR} (Transparent Never Read)

Identifier: '6F01'		Structure: transparent	Mandatory	
File size: 3 bytes		Update activity: low		
Access Conditions:				
	READ	NEVER		
	UPDATE	ALWAYS		
	INVALIDATE	ALWAYS		
	REHABILITATE	ALWAYS		
Bytes	Description	Default Value	M/O	Length
1 – 3	Test Data	AA AA AA	M	3 bytes

B.2.3 EF_{TNU} (Transparent Never Update)

Identifier: '6F02'		Structure: transparent	Mandatory	
File size: 3 bytes		Update activity: low		
Access Conditions:				
	READ	ALWAYS		
	UPDATE	NEVER		
	INVALIDATE	ALWAYS		
	REHABILITATE	ALWAYS		
Bytes	Description	Default Value	M/O	Length
1 - 3	Test Data	55 55 55	M	3 bytes

B.2.4 EF_{TARU} (Transparent Always Read and Update)

Identifier: '6F03'		Structure: transparent	Mandatory	
File size: 260 bytes		Update activity: low		
Access Conditions:				
	READ	ALWAYS		
	UPDATE	ALWAYS		
	INVALIDATE	ALWAYS		
	REHABILITATE	ALWAYS		
Bytes	Description	Default Value	M/O	Length
1 - 260	Test Data	FF ... FF	M	260 bytes

B.2.5 EF$_{CNR}$ (Cyclic Never Read)

<table>
<tr><td colspan="2">Identifier: '6F04'</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ NEVER
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.6 EF$_{CNU}$ (Cyclic Never Update)

<table>
<tr><td colspan="2">Identifier: '6F05'</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE NEVER
INCREASE NEVER
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.7 EF$_{CNIC}$ (Cyclic Never Increase)

<table>
<tr><td colspan="2">Identifier: '6F06</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE NEVER
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.8 EF$_{CNIV}$ (Cyclic Never Invalidate)

<table>
<tr><td colspan="2">Identifier: '6F07</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE NEVER
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.9 EF$_{CNRH}$ (Cyclic Never Rehabilitate)

<table>
<tr><td colspan="2">Identifier: '6F08'</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE ALWAYS
REHABILITATE NEVER</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.10 EF$_{CARU}$ (Cyclic Always Read and Update)

<table>
<tr><td colspan="2">Identifier: '6F09'</td><td colspan="2">Structure: cyclic</td><td>Mandatory</td></tr>
<tr><td colspan="3">Record length: 3 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>55 55 55</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>AA AA AA</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.11 EF$_{LNR}$ (Linear Fixed Never Read)

Identifier: '6F0A'		Structure: linear fixed		Mandatory
Record length: 4 bytes		Update activity: low		
Access Conditions: READ NEVER UPDATE ALWAYS INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

B.2.12 EF$_{LNU}$ (Linear Fixed Never Update)

Identifier: '6F0B'		Structure: linear fixed		Mandatory
Record length: 4 bytes		Update activity: low		
Access Conditions: READ ALWAYS UPDATE NEVER INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

B.2.13 EF$_{LARU}$ (Linear Fixed Always Read and Update)

Identifier: '6F0C'		Structure: linear fixed		Mandatory
Record length: 4 bytes		Update activity: low		
Access Conditions: READ ALWAYS UPDATE ALWAYS INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	55 55 55 55	M	4 bytes
2	Test Data - Record 2	AA AA AA AA	M	4 bytes

B.2.14 EF_{CINA} (Cyclic Increase Not Allowed)

<table>
<tr><td colspan="2">Identifier: '6F0D'</td><td colspan="1">Structure: cyclic</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="2">Record length: 3 bytes</td><td colspan="3">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE ALWAYS (see note 1)
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td>2</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
<tr><td colspan="5">Note 1: This file will be personalised in a way such that increase is not allowed, as indicated by the FCI byte 8, bit 7 (GSM 11.11: FCI structure of an EF returned by the SELECT command)</td></tr>
</table>

B.2.15 EF_{TRAC} (Transparent Read Access Condition CHV2)

<table>
<tr><td colspan="2">Identifier: '6F0E'</td><td colspan="1">Structure: transparent</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="2">Record length: 3 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV2
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.16 EF_{TIAC} (Transparent Invalidate Access Condition CHV1)

<table>
<tr><td colspan="2">Identifier: '6F0F'</td><td colspan="1">Structure: transparent</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="2">Record length: 3 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INCREASE ALWAYS
INVALIDATE CHV1
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data</td><td>00 00 00</td><td>M</td><td>3 bytes</td></tr>
</table>

B.2.17 EF$_{CIAC}$ (Cyclic Increase Access Condition CHV2)

Identifier: '6F10'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	CHV2
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

B.2.18 EF$_{CIAA}$ (Cyclic Increase Access Condition ADM)

Identifier: '6F11'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

B.2.19 EF$_{CNRI}$ (Cyclic Never Rehabilitate Invalidated)

Identifier: '6F12'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	NEVER

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

The file status shall be invalidated as defined in [12]

Annex C (informative):
Change history

The table below indicates all changes that have been made to the present document since drafting work began.

Change history								
Date	**TSG #**	**TSG Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
2002-07	-	-				V0.1.0 presented for information to T3 SWG API #13		0.1.0
2002-08						V0.1.1 presented for information to T3 #24	0.1.0	0.1.1