3GPP TSG-T (Terminals) Meeting #8
Düsseldorf, Germany, 21 - 23 June, 2000

Tdoc TP-000098

Source: T3

Title: Change Requests to
GSM 03.19 " SIM API - stage 2 - (Java™)"
GSM 03.48 "SIM toolkit secure messaging"

Agenda item: 6.3.3

Document for: Approval

This document contains change requests to GSM 03.48 R98 and GSM 03.19 R98 agreed by T3.

T3 Doc	Spec	CR	Rv	Rel	Subject
T3-000289	03.19	A002	1	R98	Clarifications of EVENT_FORMATTED_SMS_PP_UPD, applet example
T3-000288	03.48	A011		R98	Definition of the TAR for the Card Manager

Tdoc T3-000289

CHANGE REQUEST No : **A002 rev 1** *Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.*

Technical Specification GSM : 03.19 Version: 7.1.0

Submitted to SMG					
TSG-T # *list SMG plenary meeting no. here ↑*	for approval	X	without presentation ("non-strategic")	X
	for information		with presentation ("strategic")	

PT SMG CR cover form. Filename: crf26_3.doc

Proposed change affects: SIM [X] ME [ ] Network [ ]
(at least one should be marked with an X)

Work item: SIM API

Source: 3GPP T3 **Date:** 22/05/00

Subject: clarifications of EVENT_FORMATTED_SMS_PP_UPD, applet example

Category: *(one category and one release only shall be marked with an X)*

F	Correction	X	**Release:**	
A	Corresponds to a correction in an earlier release		Phase 2	
B	Addition of feature		Release 96	
C	Functional modification of feature		Release 97	
D	Editorial modification		Release 98	X
			Release 99	
			UMTS	

Reason for change:
The purpose of this CR is to :
- clarify the content of the EnvelopeHandler in case of EVENT_UNFORMATTED_SMS_PP_UPD
- correct the applet example

Clauses affected: §6.2, Annex C

Other specs affected:

Other releases of same spec		→ List of CRs:
Other core specifications		→ List of CRs:
MS test specifications / TBRs		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

6.2 Applet Triggering

[…]

EVENT_FORMATTED_SMS_PP_UPD

This event is triggered by Update Record EFsms with an SMS TP-UD field formatted according to GSM03.48[4].

The SIM Toolkit Framework shall :

- update the EFsms file with the data received, it is than up to the receiving toolkit applet to change the SMS stored in the file (i.e. the toolkit applet need to have access to the EFsms file)
- verify the GSM03.48[4] security of the SMS TPDU ;
- convert the Update Record EFsms in a TLV List, an EnvelopeHandler ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;

The Update Record EFsms APDU shall be converted in a TLV list as defined below :

UPDATE RECORD APDU	nb bytes	Handler TLV LIST	size
CLA, INS	2	specific event	1
P1,P2	2	device Identity rec-number	1
P3 = 176	1		1
status	1	device Identity rec-status	1
TS-SCA (RP-OA)	<= ~~18~~12	Address	Y
SMS TPDU	var	SMS TPDU	Y
padding bytes	var		Y

The EnvelopeHandler provided to the applet shall:

- return *BTAG_SMS_PP_DOWNLOAD* to the *getEnvelopeTag()* method call;
- return the Simple TLV list length to the *getLength()* method call;
- contain the Simple TLV list :

~~The order of the elements in the EnvelopeHandler TLV list :~~

EnvelopeHandler TLV List
~~SMS-PP download tag~~
~~length~~
Device identities
Address
SMS TPDU

The applet should use the *findTLV()* methods to get each Simple TLV.

The Device Identity Simple TLV is used to store the information about the absolute record number in the EFsms file and the value of the EFsms record status byte, and formatted as defined below:

Device identities Simple TLV
Device identities tag
length = 02
Absolute Record Number
Record Status

With the absolute record number the toolkit applet can update EFsms in absolute mode to change the received SMS in a readable text.

Annex C (informative): Toolkit applet example

```
/**
 * Example of Toolkit Applet
 */

package ToolkitAppletExample;

import sim.toolkit.*;
import sim.access.*;
import javacard.framework.*;


public class MyToolkitApplet extends javacard.framework.Applet implements ToolkitInterface,
ToolkitConstants{

    public static final byte MY_INSTRUCTION      = (byte)0x46;
    public static final byte SERVER_OPERATION    = (byte)0x0F;
    public static final byte CMD_QUALIFIER       = (byte)0x80;
    public static final byte EXIT_REQUESTED_BY_USER  = (byte)0x10;
    private byte[] menuEntry =    {(byte)'S',(byte)'e',(byte)'r',(byte)'v',(byte)'i',(byte)'c',
                                   (byte)'e', (byte)'1'};
     private byte[] menuTitle=    {(byte)'M',(byte)'y',(byte)'M',(byte)'e',(byte)'n' ,(byte)'u'};
    private byte[] item1 =        {(byte)'I',(byte)'T',(byte)'E',(byte)'M',(byte)'1' };
    private byte[] item2 =        {(byte)'I',(byte)'T',(byte)'E',(byte)'M',(byte)'2' };
    private byte[] item3 =        {(byte)'I',(byte)'T',(byte)'E',(byte)'M',(byte)'3' };
    private byte[] item4 =        {(byte)'I',(byte)'T',(byte)'E',(byte)'M',(byte)'4' };
    private Object[] ItemList =   { item1, item2, item3, item4 };
    private byte[] textDText =    {(byte)'H',(byte)'e',(byte)'l',(byte)'l',(byte)'o',(byte)' ',
                                   (byte)'w',(byte)'o',(byte)'r',(byte)'l',(byte)'d',(byte)'2'};
    private byte[] textGInput =    {(byte)'Y',(byte)'o',(byte)'u',(byte)'r',(byte)' ',(byte)'n',
                                   (byte)'a',(byte)'m',(byte)'e',(byte)'?'};

    private byte[]  baGSMAID =
{(byte)0xA0,(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x09,(byte)0x00,(byte)0x01};
    private ToolkitRegistry reg;
    private SIMView gsmFile;
    private byte buffer[] = new byte[10];
    private byte itemId;
    private byte result;
    private boolean repeat;

    /**
     * Constructor of the applet
     */
    public MyToolkitApplet() {

        // get the GSM application reference
        gsmFile = SIMSystem.getTheSIMView();

        // register to the SIM Toolkit Framework
        reg = ToolkitRegistry.getEntry();

        // Define the applet Menu Entry and register to the EVENT_MENU_SELECTION
        itemId = reg.initMenuEntry(menuEntry, (short)0x0000, (short)menuEntry.length,
                                   PRO_CMD_DISPLAY_TEXT, false, (byte) 0x00, (short) 0x0000);
        // register to the EVENT_UNFORMATTED_SMS_PP_ENV
        reg.setEvent(EVENT_UNFORMATTED_SMS_PP_ENV);
    }

    /**
     * Method called by the JCRE at the installation of the applet
     */
    public static void install(byte bArray[], short bOffset, byte bLength) {
        MyToolkitApplet MyApplet = new MyToolkitApplet ();
        MyApplet.register();
    }

    /**
     * Method called by the GSM Framework
     */
    public Shareable getShareableInterfaceObject ( AID clientAID, byte parameter)
    {
        if ( clientAID.partialEquals(baGSMAID, (byte) 0x00, (byte) baGSMAID.length) == true )}
                return ((Shareable) this);
        else
                return(null);
```

```
    }

     /**
      * Method called by the SIM Toolkit Framework
      */
     public void processToolkit(byte event) {

        // get the handler references
        EnvelopeHandler          envHdlr = EnvelopeHandler.getTheHandler();
        ProactiveHandler         proHdlr = ProactiveHandler.getTheHandler();
        ProactiveResponseHandler rspHdlr = ProactiveResponseHandler.getTheHandler();

         switch(event) {
            case EVENT_MENU_SELECTION:
                // Prepare the Select Item proactive command
                proHdlr.init(PRO_CMD_SELECT_ITEM,(byte)0x00,DEV_ID_ME);
                // Append the Menu Title
                proHdlr.appendTLV((byte) (TAG_ALPHA_IDENTIFIER | TAG_SET_CR),
                                  menuTitle,(short)0x0000,(short)menuTitle.length);
                // add all the Item
                for (short i=(short) 0x0000; i<(short) 0x0004; i++) {
                    proHdlr.appendTLV((byte) (TAG_ITEM | TAG_SET_CR),(byte) (i+1),
                                      (byte[])ItemList[i],(short) 0x0000,
                                      (short)((byte[])ItemList[i]).length);
                }
                // ask the SIM Toolkit Framework to send the proactive command and check the result
                if ((result = proHdlr.send()) == RES_CMD_PERF){
                    rspHdlr = ProactiveResponseHandler.getTheHandler();
                    // SelectItem response handling
                    switch (rspHdlr.getItemIdentifier()) {
                        case 1:
                        case 2:
                        case 3: // DisplayText
                            proHdlr.init(PRO_CMD_DISPLAY_TEXT, CMD_QUALIFIER,
                                         DEV_ID_DISPLAY);
                            proHdlr.appendTLV((byte)(TAG_TEXT_STRING| TAG_SET_CR), DCS_8_BIT_DATA,
                                         textDText,(short)0x0000, (short)textDText.length);
                            proHdlr.send();
                            break;
                        case 4: // Ask the user to enter data and display it
                            do {
                                repeat = false;
                                try {
                                // GetInput followed by a DisplayText of the entered text
                                    // GetInput asking the users name
                                    proHdlr.initGetInput((byte)0x01, DCS_8_BIT_DATA,
textGInput,(byte)0x00,
                                        (short)textGInput.length,(short)0x0001,(short)0x0002);
                                    proHdlr.send();

                                    // display the entered text
                                    rspHdlr.copyTextString(textDText,(short)0x0000);
                                    proHdlr.initDisplayText((byte)0x00,DCS_8_BIT_DATA, textDText,
                                        (short)0x0001,(short) textDText.length);
                                    proHdlr.send();
                                }
                                catch (ToolkitException MyException) {
                                    if (MyException.getReason() ==
ToolkitException.UNAVAILABLE_ELEMENT) {
                                        if (rspHdlr.getGeneralResult() != EXIT_REQUESTED_BY_USER)
                                            repeat = true;
                                        break;
                                    }
                                }
                            }
                            while (repeat);
                            break;
                    }
                }
                break;

            case EVENT_UNFORMATTED_SMS_PP_ENV:
                // get the offset of the instruction in the TP-UD field offset
                short TPUDOffset = (short) (envHdlr.getTPUDLOffset() + SERVER_OPERATION);

                // start the action requested by the server
                switch (envHdlr.getValueByte((short)TPUDOffset) ) {
                case 0x41 : // Update of a gsm file
                    // get the data from the received SMS
                    envHdlr.copyValue((short)TPUDOffset+1,buffer, (short)0x0000,(short)0x0003);
                    // write these data in the EFpuct
                    gsmFile.select(SIMView.FID_DF_GSM);
                    gsmFile.select(SIMView.FID_EF_PUCT);
```

```
                    gsmFile.updateBinary((short)0x0000,buffer,(short)0x0000,(short)0x0003);

                    break;

                case 0x36 : // change the MenuTitle for the SelectItem
                    envHdlr.copyValue((short)TPUDOffset+1, menuTitle,(short)0x0000,(short)0x0006);
                    break;
                }
                break;
            }

    }

     /**
      * Method called by the JCRE, once selected
      */
     public void process(APDU apdu) {
         // Handle the Select AID apdu
         if (selectingApplet()) return;

         switch(apdu.getBuffer()[1]) {
            // specific APDU for this applet to configure the MenuTitle from SelectItem
            case (byte)MY_INSTRUCTION:
                if (apdu.setIncomingAndReceive()>(short)0) {
                    Util.arrayCopy(apdu.getBuffer(),(short)0x0005,menuTitle,(short)0x0000,
                                   (short)0x0006);
                }
                break;
            default:
                ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
         }
     }
}
```

3GPP T3 Meeting #14
Visby, Sweden, 24 - 26 May, 2000

Tdoc T3-000288

CHANGE REQUEST No : A011 *Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.*

Technical Specification GSM : 03.48 Version: 8.2.0

Submitted to SMG TSG-T # *list SMG plenary meeting no. here ↑*
for approval [X] for information [ ]
without presentation ("non-strategic") [X] with presentation ("strategic") [ ]

PT SMG CR cover form. Filename: crf26_3.doc

Proposed change affects: *(at least one should be marked with an X)* SIM [X] ME [ ] Network [ ]

Work item: SIM API

Source: 3GPP T3 **Date:** 22/05/00

Subject: Definition of the TAR for the Card Manager

Category: *(one category and one release only shall be marked with an X)*

F	Correction	X
A	Corresponds to a correction in an earlier release	
B	Addition of feature	
C	Functional modification of feature	
D	Editorial modification	

Release:

Phase 2	
Release 96	
Release 97	
Release 98	
Release 99	X
UMTS	

Reason for change: The purpose of this CR is to define the TAR of the Card Manager being '000000' for GSM 03.19 compliant cards.

Clauses affected: §9.2

Other specs affected:

Other releases of same spec		→ List of CRs:
Other core specifications		→ List of CRs:
MS test specifications / TBRs		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

9.2 Commands coding

Commands are coded as for the Remote File Management procedure, each command is coded as an APDU.

The messages for the Card Manager shall have a TAR value set to '000000' in hexadecimal.

9.2.1 Input Commands

The following table extends table 10 defined in 8.2.1.

Table 14: Applet Management input commands

Operational command
DELETE
SET STATUS
INSTALL
LOAD
PUT KEY

9.2.2 Output Commands

The following table extends table 11 defined in 8.2.2.

Table 15: Applet Management output commands

Operational command
GET STATUS
GET DATA

9.3 Response Packets

The behaviour of the SIM's RE/RA with regard to PoR is the same as the one defined for Remote File Management (see subclause 8.3).