3GPP TSG-T (Terminals) Meeting #16
Marco Island, USA 4 – 6 June 2002

Tdoc TP-020120

Source: T3

Title: Change Requests to TS 43.019

Document for: Approval

This document contains several change requests as follows:

T3 Doc	Spec	CR	Rel	Cat	Subject
T3-020388	43.019	016	4	F	Clarification of MEProfile behaviour
T3-020389	43.019	017	5	F	Clarification of MEProfile behaviour
T3-020390	43.019	018	5	F	Approved CRs not correct integrated in the current version
T3-020392	43.019	019	4	F	Correction of getSecuredDataOffset() method description for SMS-CB.
T3-020393	43.019	020	5	F	Correction of getSecuredDataOffset() method description for SMS-CB.

The following CR was presented at TSG-T#15. TSG-T proposed to discuss the issue with T2 and represent the CR at T#16 for approval. Please also refer to related Liaison Statement in T3-020378.

T3 Doc	Spec	CR	Rel	Cat	Subject
T3-020085	43.019	013	5	B	Introduction of Concatenated Short Messages in SMS Point to Point.

3GPP T3 Meeting #22
Marbella, Spain, 22 - 25 Jan 2002

Tdoc T3-020085

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **013** ⌘ rev **-** ⌘ Current version: **5.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Introduction of Concatenated Short Messages in SMS Point to Point.
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	23/01/2002
Category:	⌘	**B**
Release:	⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Current specification is limited to single SMS-PP
Summary of change:	⌘	Define/modify : - Framework behaviour upon concatenated short messages receipt. - API methods allowing Toolkit Applet to access concatenated short message data.
Consequences if not approved:	⌘	No standardize solution to process Concatenated Short Messages in SMS Point to Point.

Clauses affected:	⌘	§ 2, § 6.2, § 6.8, Annex A, Annex B
Other specs Affected:	⌘	[ ] Other core specifications ⌘ [ ] Test specifications [ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

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.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.
- For this Release 1999 document, references to GSM documents are for Release 1999 versions (version 8.x.y).

[1] 3GPP TR 21.905: "Abbreviations and acronyms".

[2] 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface".

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

[4] 3GPP TS 23.048: "Security Mechanisms for the SIM application toolkit; Stage 2".

[5] ISO/IEC 7816-3 (1997) " Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols".

[6] 3GPP TS 42.019: "Subscriber Identity Module Application Programming Interface (SIM API); Service description; Stage 1".

[7] SUN Java Card Specification "Java Card 2.1 API Specification".

[8] SUN Java Card Specification "Java Card 2.1 Runtime Environment Specification".

[9] SUN Java Card Specification "Java Card 2.1 VM Architecture Specification".

SUN Java Card Specifications can be downloaded at http://java.sun.com/products/javacard

[10] ETSI TS 101 220 "Integrated Circuit Cards (ICC); ETSI numbering system for telecommunication; Application providers (AID)".

[11] 3GPP TS 23.040: “Technical realization of the Short Message Service (SMS)”

[12] ISO/IEC 7816-6 (1995): "Identification cards - Integrated circuit(s) cards with contacts, Part 6 Inter-industry data elements".

6.2 Applet Triggering

[..]

EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_UPD

A toolkit applet can be activated upon the reception of a short message.

There are two ways for a card to receive an ~~SMS :~~SMS: via the Envelope SMS-PP Data Download or the Update Record EFsms instruction.

The reception of the SMS by the toolkit applet cannot be guaranteed for the Update Record EFsms instruction.

The received SMS may ~~be :~~be:
- formatted according to TS 23.048[4] or an other protocol to identify explicitly the toolkit applet for which the message is ~~sent ;~~sent;
- unformatted or using a toolkit applet specific protocol the SIM Toolkit Framework will pass this data to all registered toolkit applets.

The Short Message may be received as Concatenated Short Messages as defined in TS 23.040[11]. It is the responsibility of the SIM Toolkit Framework to link single Short Messages together to re-assemble the original message before any further processing. The original Short Message shall be placed in one SMS TPDU TLV (with TP-UDL field coded on one octet) included in the EnvelopeHandler. The concatenation control headers used to re-assemble the short messages in the correct order shall not be present in the SMS TPDU. The TP-elements of the SMS TPDU and the Address (TS-Service-Centre-Address) shall correspond to the ones in the last received Short Message (independently of the Sequence number of Information-Element-Data).

The minimum requirement for the SIM Toolkit Framework is to process a concatenated short message with the following properties:
- the Information Element Identifier is equal to the 8-bit reference number.
- it contains uncompressed 8 bit data or uncompressed UCS2 data.

EVENT_FORMATTED_SMS_PP_ENV

~~This event is triggered by an envelope APDU containing an SMS_DATADOWNLOAD BER TLV with an SMS_TPDU simple TLV according to TS 23.048[4].~~

This event is generated when a Short Message Point to Point (Single or Concatenated) is received by Envelope SMS-PP download APDU(s) and is formatted according to TS 23.048[4].

The SIM Toolkit Framework shall:
- verify the~~ TS 23.048[4]~~ security of the Short Message as per TS 23.048[4]~~SMS TPDU~~ ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;
- take the optional Application Data posted by the triggered toolkit applet if present;
- secure and send the response packet.

~~The toolkit applet will only be triggered if the TAR is known and the security verified, application data will also be deciphered.~~

When the toolkit applet is triggered, data shall be provided deciphered.

EVENT_UNFORMATTED_SMS_PP_ENV

This event is generated when a Short Message Point to Point (Single or Concatenated) is received by Envelope SMS-PP download APDU(s) and is unformatted.

The registered toolkit applets will be triggered by this event and get the data transmitted in the Envelope APDU(s).~~APDU envelope SMS_DATADOWNLOAD~~.

But only the first toolkit applet triggered will be able to send back a response as defined by the rules in chapter 6.6.

EVENT_FORMATTED_SMS_PP_UPD

This event is generated when a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is formatted according to TS 23.048[4].

~~This event is triggered by Update Record EFsms with an SMS TP-UD field formatted according to TS 23.048[4].~~

The SIM Toolkit Framework ~~shall :~~shall:

- update the EFsms file with the data received, it is then 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 ~~TS 23.048[4]~~ security of the Short Message as per TS 23.048[4] ~~SMS TPDU~~ ;
- convert the Update Record EFsms in the EnvelopeHandler a ~~-~~ TLV List~~, an EnvelopeHandler~~ ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;

When the toolkit applet is triggered, data shall be provided deciphered.

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

UPDATE RECORD APDU	Nb bytes	Handler TLV LIST	size
CLA, INS	2	~~Specific event~~	1
P1,P2	2	device Identity Absolute Record Number ~~rec-number~~	1
P3 = 176	1		1
Status	1	device Identity Record Status~~rec-status~~	1
TS-SCA (RP-OA)	<= 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 ~~method call~~;
- contain the Simple TLV list :

EnvelopeHandler TLV List
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.

For Concatenated Short Message the Absolute Record Number and the Record Status will correspond to the last Update Record EFsms APDU received.

EVENT_UNFORMATTED_SMS_PP_UPD

This event is generated when a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is unformatted.

The SIM Toolkit Framework will first update the EFsms file, convert the received APDU as described above, and then trigger all the registered toolkit applets. All of them may modify the content of EFsms (i.e. the toolkit applets need to have access to the EFsms file).

6.8 Usage of ViewHandler and EditHandler

The ViewHandler and EditHandler classes have been defined to group the properties of the system handler, and may be used in the future to provide a simple mechanism to the toolkit applet to handle TLV lists. The length of simple TLV present in a Handler TLV List shall be coded according to ISO/IEC 7816-6 [12] (e.g. coded onto 1,or 2 or 3 bytes).

Annex A (normative): Java Card SIM API

The attached files "Annex_A_java.zip" and "Annex_A_HTML.zip" contains source files for the Java Card SIM API.

[The HTML and JAVA source files will be included]

Annex B (normative): Java Card SIM API identifiers

The attached file "Annex_B_Export_files.zip" contains source files for the Java Card SIM API identifiers.

[The export files will be included]

NOTE: The export files in this annex have been generated with the following steps and tools :

- Compilation from the API java source file (.java) to the API class files (.class) with the Java compiler from the Java Development Kit version 1.2.2.
- Convertion from the API class files (.class) to the API export files (.exp) with the Java Card 2.1.2 Class File Converter (version 1.2) and the Java Development Kit 1.2.2.

List of changes to the API html and java source files

Class sim.toolkit.EnvelopeHandler

Modify the methods:

```
/**
    * Looks for the TP-UDL field in the first TPDU TLV element in the Envelope
    * data field. This method can be used on the events EVENT_FORMATTED_SMS_PP_ENV,
    * EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_ENV,
EVENT_UNFORMATTED_SMS_PP_UPD.
    * If the element is available it becomes the TLV selected.
    *
    * @return TPUDL offset in the first TPDU TLV element if TPUDL exists.
    *     The value retrieved by using getValueByte is meaningless when the message is
    Concatenated.
    *     To recover the TP-User-Data-Length the method getUserDataLength() shall be used.The
    TPUD length can be recovered by using the getValueByte method in Handler class.
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TPDU TLV element or if
the TPUDL field does not exist</ul>
    */
   public short getTPUDLOffset() throws ToolkitException {
    return 0;
   }

   /**
    * Looks for the Secured Data from the Command Packet in the first SMS TPDU
    * or Cell Broadcast Page Simple TLV contained in the Envelope handler. This can
    * be used on the events:
    * - EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, if the SMS TP-
UD is formatted
    * according to GSM03.48 Single or Concatenated Short Message.
    * - EVENT_FORMATTED_SMS_CB, if the Cell Broadcast Page is formatted according to GSM
03.48.
    * If the element is available it becomes the TLV selected.
    *
    * @return the offset of the Secured Data first byte in the first SMS TPDU or Cell Broadcast Page
TLV element. If the Secured Data length is zero the value returned shall be the SMS TPDU TLV length.
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable SMS TPDU or Cell
Broadcast Page TLV element or wrong data format </ul>
    */

    public short getSecuredDataOffset() throws ToolkitException {
    return 0;
   }

   /**
    * Looks for the length of the Secured Data from the Command Packet in the first SMS TPDU
    * or Cell Broadcast Page Simple TLV contained in the Envelope handler. This can be used
    * on the events:
    * - EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, if the SMS TP-
UD
    * is formatted according to GSM03.48 Single or Concatenated Short Message.
    * - EVENT_FORMATTED_SMS_CB, if the Cell Broadcast Page is formatted according to GSM
03.48.
    * If the element is available it becomes the TLV selected.
    *
```

```
        * @return the length of the Secured Data contained in the first SMS TPDU or Cell Broadcast Page
TLV element (without padding bytes). If the Secured Data length is zero, no exception shall be thrown.
        *
        * @exception ToolkitException with the following reason codes: <ul>
        *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable SMS TPDU or Cell
Broadcast Page TLV element or wrong data format </ul>
        */

        public short getSecuredDataLength() throws ToolkitException {
        return 0;
       }
```

Add the method:

- getUserDataLength()

```
/**
    * Looks for the TP-User Data field in the first TPDU TLV element contained in the Envelope
Handler
    * This method can be used on the events EVENT_FORMATTED_SMS_PP_ENV,
    * EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_ENV,
EVENT_UNFORMATTED_SMS_PP_UPD.
    * If the element is available it becomes the TLV selected.
    *
    * @return the length of the User Data contained in the first SMS TPDU TLV element.
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TPDU TLV element or
wrong data format </ul>
    */

public short getUserDataLength() throws ToolkitException {
    return 0;
}
```

3GPP TSG-T3 Meeting #23
Espoo, Finland, 21- 24 May 2002

Tdoc T3-020388

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **016** ⌘ rev **-** ⌘ Current version: **4.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarification of MEProfile behaviour
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	24/05/02
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change: ⌘ Unspecified behaviour of the MEProfile class and correction of errors

Summary of change: ⌘ The behaviour of the MEProfile class is not specified in the following cases:

a) Corrects the rule when ToolkitException.BAD_INPUT_PARAMETER is thrown for the check method in case a negative index value is used

b) To allow the access to 16 bits for the getValue method

c) Define the position of the bits in the MEProfile data in the short returned value for the getValue method

d) Define the MEProfile class behaviour when the requested bits are outside of the range of the MEProfile Data for all methods: the requested features are considered as not supported.

e) Define the MEProfile class behaviour when a negative startOffset parameter is submitted to the copy method: the Toolkit Exception "BAD_INPUT_PARAMETER is thrown.

Consequences if not approved: ⌘ Different interpretation and misunderstanding of functionality of the MEProfile methods, and thus interoperability issues

Clauses affected: ⌘ Annex A

Other specs Affected: ⌘
[ ] Other core specifications ⌘
[ ] Test specifications
[ ] O&M Specifications

Other comments: ⌘

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
     * Checks a facility in the handset profile.
     *
     * @param index the number of the facility to check, according to the table above.
     *
     * @return true if the facility is supported, false otherwiseif facility is not supported,
     *         or if facility-index outside MEProfile data
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if index has a negative value</ul>
     */
    public static boolean check(byte index) throws ToolkitException {
        return false;
    }

/**
     * Checks a set of facilities in the handset profile.
     * The method checks all the facilities corresponding to bits set to 1 in
     * the mask buffer.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no check is performed.</em>
     * <li><em>If </em><code>offset+length</code><em> is greater than </em><code>mask.length</code><em>, the length
     * of the </em><code>mask</code><em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no check is performed.</em>
     * </ul>
     *
     * @param mask a byte array containing the mask to compare with the profile
     * @param offset the starting offset of the mask in the byte array
     * @param length the length of the mask (at least 1)
     *.@return true if the bitwise AND of the MEProfile data padded with 0 and the <code>mask</code> is equal to the
<code>mask</code>set of facilities is supported, false otherwise. If <code>length</code> is equal to <code>0</code>, true is
returned.

     * @exception NullPointerException if <code>mask</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if check would cause access of data outside mask array bounds
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
*/
    public static boolean check(byte[] mask,
                                short offset,
                                short length) throws NullPointerException,
                                                     ArrayIndexOutOfBoundsException,
                                                     ToolkitException {
        return false;
    }

    /**
     * Checks a facility in the handset profile.
     *
     * @param index the number of the facility to check, according to the table above.
     *
     * @return true if the facility is supported, false if facility is not supported,
     *         or if facility-index outside MEProfile dataotherwise
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if index has a negative value</ul>
     */
    public static boolean check(short index) throws ToolkitException {
        return false;
    }

   /**
     * Returns the binary value of a parameter, delimited by two indexes, from the handset profile.
     *
     * @param indexMSB index of the Most Significant Bit of the handset profile .
     * @param indexLSB index of the Lowest Significant Bit of the handset profile .
     *
     * @return binary value of the data field indicated in the handset profile.
The indexLSB bit in the MEProfile data is the Lowest Significant bit in the short returned value. If padding is necessary, the
returned value is left padded with 0. The values outside the MEProfile data available are considered to bet set to 0.

_____*         The return value is according to the following example:<ul>
     *              <li>If indexMSB=108 and indexLSB=104, the return value is the number of
     *              characters down ME display.</li>
     *              <li>If indexMSB=31 and indexLSB=16, the return value is a short built
     *              from the 4th and 3rd byte of the handset profile with the 4th byte as
     *              the Most significant byte.</li></ul>
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available
     *      <li>BAD_INPUT_PARAMETER if (indexMSB >= indexLSB +16) or (indexMSB < indexLSB) or
     *      (indexMSB < 0) or (indexLSB < 0) </ul>
     */
    public static short getValue(short indexMSB, short indexLSB) throws ToolkitException {
        return 0;
    }

   /**
     * Copies a part of the handset profile in a buffer.
       The values outside the MEProfile data available are considered to bet set to 0.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no copy is performed.</em>
     * <li><em>If </em><code>dstOffset+dstLength</code><em> is greater than </em><code>dstBuffer.length</code><em>, the length
     * of the </em><code>dstBuffer</code><em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no copy is performed.</em>
     * </ul>
```

```
*
     * @param startOffset offset of the handset profile first byte to be copied
     * @param dstBuffer destination byte array
     * @param dstOffset offset within destination byte array to start copy into
     * @param dstLength byte length to be copy
     *
     * @return dstOffset + dstLength
     *
     * @exception ArrayIndexOutOfBoundsException if copy would cause access of data outside array bounds
     * @exception NullPointerException if <code>dstBuffer<code> is null
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if the startOffset is negative</ul>
     */
    public static short copy(short  startOffset, byte[] dstBuffer, short dstOffset, short dstLength)
        throws ArrayIndexOutOfBoundsException, NullPointerException, ToolkitException {
        return 0;
    }
```

3GPP TSG-T3 Meeting #23
Espoo, Finland, 21- 24 May 2002

Tdoc T3-020389

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **017** ⌘ rev **-** ⌘ Current version: **5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarification of MEProfile behaviour
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	24/05/02
Category:	⌘	**F**
Release:	⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change: ⌘ Unspecified behaviour of the MEProfile class and correction of errors

Summary of change: ⌘ The behaviour of the MEProfile class is not specified in the following cases:

a) Corrects the rule when ToolkitException.BAD_INPUT_PARAMETER is thrown for the check method in case a negative index value is used

b) To allow the access to 16 bits for the getValue method

c) Define the position of the bits in the MEProfile data in the short returned value for the getValue method

d) Define the MEProfile class behaviour when the requested bits are outside of the range of the MEProfile Data for all methods: the requested features are considered as not supported.

e) Define the MEProfile class behaviour when a negative startOffset parameter is submitted to the copy method: the Toolkit Exception "BAD_INPUT_PARAMETER is thrown.

Consequences if not approved: ⌘ Different interpretation and misunderstanding of functionality of the MEProfile methods, and thus interoperability issues

Clauses affected: ⌘ Annex A

Other specs Affected: ⌘
[ ] Other core specifications ⌘
[ ] Test specifications
[ ] O&M Specifications

Other comments: ⌘

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
     * Checks a facility in the handset profile.
     *
     * @param index the number of the facility to check, according to the table above.
     *
     * @return true if the facility is supported, false otherwiseif facility is not supported,
     *         or if facility-index outside MEProfile data
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if index has a negative value</ul>
     */
    public static boolean check(byte index) throws ToolkitException {
        return false;
    }

/**
     * Checks a set of facilities in the handset profile.
     * The method checks all the facilities corresponding to bits set to 1 in
     * the mask buffer.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no check is performed.</em>
     * <li><em>If </em><code>offset+length</code><em> is greater than </em><code>mask.length</code><em>, the length
     * of the </em><code>mask</code><em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no check is performed.</em>
     * </ul>
     *
     * @param mask a byte array containing the mask to compare with the profile
     * @param offset the starting offset of the mask in the byte array
     * @param length the length of the mask (at least 1)
     *.@return true if the bitwise AND of the MEProfile data padded with 0 and the <code>mask</code> is equal to the
<code>mask</code>set of facilities is supported, false otherwise. If <code>length</code> is equal to <code>0</code>, true is
returned.

     * @exception NullPointerException if <code>mask</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if check would cause access of data outside mask array bounds
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
*/
    public static boolean check(byte[] mask,
                                short offset,
                                short length) throws NullPointerException,
                                                     ArrayIndexOutOfBoundsException,
                                                     ToolkitException {
        return false;
    }

    /**
     * Checks a facility in the handset profile.
     *
     * @param index the number of the facility to check, according to the table above.
     *
     * @return true if the facility is supported, false if facility is not supported,
     *         or if facility-index outside MEProfile dataotherwise
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if index has a negative value</ul>
     */
    public static boolean check(short index) throws ToolkitException {
        return false;
    }

   /**
     * Returns the binary value of a parameter, delimited by two indexes, from the handset profile.
     *
     * @param indexMSB index of the Most Significant Bit of the handset profile .
     * @param indexLSB index of the Lowest Significant Bit of the handset profile .
     *
     * @return binary value of the data field indicated in the handset profile.
The indexLSB bit in the MEProfile data is the Lowest Significant bit in the short returned value. If padding is necessary, the
returned value is left padded with 0. The values outside the MEProfile data available are considered to bet set to 0.

_____*         The return value is according to the following example:<ul>
     *              <li>If indexMSB=108 and indexLSB=104, the return value is the number of
     *              characters down ME display.</li>
     *              <li>If indexMSB=31 and indexLSB=16, the return value is a short built
     *              from the 4th and 3rd byte of the handset profile with the 4th byte as
     *              the Most significant byte.</li></ul>
     *
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available
     *      <li>BAD_INPUT_PARAMETER if (indexMSB >= indexLSB +16) or (indexMSB < indexLSB) or
     *      (indexMSB < 0) or (indexLSB < 0) </ul>
     */
    public static short getValue(short indexMSB, short indexLSB) throws ToolkitException {
        return 0;
    }

   /**
     * Copies a part of the handset profile in a buffer.
       The values outside the MEProfile data available are considered to bet set to 0.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no copy is performed.</em>
     * <li><em>If </em><code>dstOffset+dstLength</code><em> is greater than </em><code>dstBuffer.length</code><em>, the length
     * of the </em><code>dstBuffer</code><em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no copy is performed.</em>
     * </ul>
```

```
*
     * @param startOffset offset of the handset profile first byte to be copied
     * @param dstBuffer destination byte array
     * @param dstOffset offset within destination byte array to start copy into
     * @param dstLength byte length to be copy
     *
     * @return dstOffset + dstLength
     *
     * @exception ArrayIndexOutOfBoundsException if copy would cause access of data outside array bounds
     * @exception NullPointerException if <code>dstBuffer<code> is null
     * @exception ToolkitException  with the following reason codes: <ul>
     *      <li>ME_PROFILE_NOT_AVAILABLE if Terminal Profile data are not available</ul>
     *      <li>BAD_INPUT_PARAMETER if the startOffset is negative</ul>
     */
    public static short copy(short  startOffset, byte[] dstBuffer, short dstOffset, short dstLength)
        throws ArrayIndexOutOfBoundsException, NullPointerException, ToolkitException {
        return 0;
    }
```

3GPP T3 Meeting #12
Espoo, Finland, 21 - 24 May 2002

Tdoc T3-020390

CR-Form-v3

CHANGE REQUEST

⌘ **43.019 CR 018** ⌘ rev ⌘ Current version: **5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Approved CRs not correct integrated in the current version
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	24/05/02
Category:	⌘	**F**
Release:	⌘	Rel-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	TP-020073-CR10 included in 43.019 v5.2.0 has been based on 43.019 v5.0.1 instead of v5.1.0. So the TP-010241-CR07 is no more included in the current version. TP-020073-CR15 has not been correctly integrated in the current version v5.2.0.
Summary of change:	⌘	Include again TP-010241-CR07 Integrate fully TP-020073-CR15.
Consequences if not approved:	⌘	No coherence between 43.019 v5.1.0 & v5.2.0.

Clauses affected:	⌘	§ 6.2, § 6.6		
Other specs affected:	⌘	[ ] Other core specifications [ ] Test specifications [ ] O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.2 Applet Triggering

[…]

~~*EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST*~~

EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST

A toolkit applet might be activated upon selection in the ME's menu by the user, or request help on this specific menu.

In order to allow the user to choose in a menu, the SIM Toolkit Framework shall have previously issued a SET UP MENU proactive command. When a toolkit applet changes a menu entry of its registry object, the SIM Toolkit Framework shall dynamically update the menu stored in the ME during the current card session. The SIM Toolkit Framework shall use the data of the EFsume file when issuing the SET UP MENU proactive command.

The positions of the toolkit applet menu entries in the item list, the requested item identifiers and the associated limits (e.g. maximum length of item text string) are defined at the loading of the toolkit applet.

If at least one Menu id of a toolkit applet registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the SIM Toolkit Framework shall indicate to the ME that help information is available unless all the menus entries that support help are disabled. ~~A toolkit applet registered for one or more menu entries, may be triggered by the event EVENT_MENU_SELECTION_HELP_REQUEST, even if it is not registered to this event. A toolkit applet registered for one or more menu entries should provide help information.~~

A toolkit applet shall be triggered by the EVENT_MENU_SELECTION_HELP_REQUEST event only if the Menu Id corresponding to the Envelope Menu Selection Help Request received by the SIM Toolkit framework was registered with the *helpSupported* value set to true.

[…]

6.6 Handler availability

The system handlers : ProactiveHandler, ProactiveResponseHandler, EnvelopeHandler and EnvelopeResponseHandler are Temporary JCRE Entry Point Object as defined in the Java Card Runtime Environment Specification [8].

The following rules define the availability of the system handlers and the lifetime of their content. They are generic rules and may vary with the event that triggers the toolkit applet.

ProactiveHandler:

- The ProactiveHandler is valid from the invocation to the termination of the processToolkit method.
- If a proactive command is pending the ProactiveHandler may not be available.
- At the processToolkit method invocation the TLV-List is cleared.
- At the call of it's init method the content is cleared and then initialised.
- After a call to ProactiveHandler.send method the handler will remain unchanged (i.e. previously send proactive command) until the ProactiveHandler.init or appendTLV methods are called.

[…]

3GPP T3 Meeting #23
Espoo, Finland, 22 - 24 May 2002

Tdoc T3-020392

CR-Form-v5.1

CHANGE REQUEST

⌘ **43.019** **CR** **019** ⌘ **rev** **-** ⌘ Current version: **4.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Correction of getSecuredDataOffset() method description for SMS-CB.
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	24/05/02
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	The 03.19 CR A010 (Tdoc T3-010064) clarified the getSecuredDataOffset() method behaviour. Unfortunately the clarification did not take into account that in R99 (and subsequent Releases REL-4 & REL-5) support for SMS-CB formatted have been integrated. Therefore the clarification is not enough for Cell Broadcast Page. Indeed the following " If the Secured Data length is zero the value returned shall be the SMS TPDU TLV length." cannot and does not apply for a Cell Broadcast Page.
Summary of change:	⌘	Re-write getSecuredDataOffset() method description, so that it behaves in the same way for both SMS-PP and SMS-CB.
Consequences if not approved:	⌘	The behaviour for SMS-CB is not described when Secured Data length is zero.

Clauses affected:	⌘	Annex A (normative): Java Card SIM API
Other specs Affected:	⌘	[ ] Other core specifications ⌘ [ ] Test specifications [ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Annex A (normative): Java Card SIM API

The attached files "Annex_A_java.zip" and "Annex_A_HTML.zip" contains source files for the Java Card SIM API.

List of changes to the API html and java source files

Class sim.toolkit.EnvelopeHandler

```
/**
    * Looks for the Secured Data from the Command Packet in the first SMS TPDU
    * or Cell Broadcast Page Simple TLV contained in the Envelope handler. This can
    * be used on the events:
    * - EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, if the SMS TP-UD is formatted
    * according to TSGSM03.48 Single Short Message.
    * - EVENT_FORMATTED_SMS_CB, if the Cell Broadcast Page is formatted according to GSM 03.48.
    * If the element is available it becomes the TLV selected.
    *
    * @return the offset of the Secured Data first byte in the first SMS TPDU or Cell Broadcast Page TLV element. If the Secured Data length is zero the value returned shall be the offset of the first byte following the TS 03.48 Command Packet structure. If the Secured Data length is zero the value returned shall be the SMS TPDU TLV length.
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable SMS TPDU or Cell Broadcast Page TLV element or wrong data format </ul>
    */

    public short getSecuredDataOffset() throws ToolkitException {
        return 0;
    }
```

3GPP T3 Meeting #23
Espoo, Finland, 22 - 24 May 2002

Tdoc T3-020393

CR-Form-v5.1

CHANGE REQUEST

⌘ **43.019** **CR** **020** ⌘ **rev** **-** ⌘ Current version: **5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Correction of getSecuredDataOffset() method description for SMS-CB.
Source:	⌘	T3
Work item code:	⌘	USAT1 API Java
Date:	⌘	24/05/02
Category:	⌘	**F**
Release:	⌘	REL-5

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	The 03.19 CR A010 (Tdoc T3-010064) clarified the getSecuredDataOffset() method behaviour. Unfortunately the clarification did not take into account that in R99 (and subsequent Releases REL-4 & REL-5) support for SMS-CB formatted have been integrated. Therefore the clarification is not enough for Cell Broadcast Page. Indeed the following " If the Secured Data length is zero the value returned shall be the SMS TPDU TLV length." cannot and does not apply for a Cell Broadcast Page.
Summary of change:	⌘	Re-write getSecuredDataOffset() method description, so that it behaves in the same way for both SMS-PP and SMS-CB.
Consequences if not approved:	⌘	The behaviour for SMS-CB is not described when Secured Data length is zero.

Clauses affected:	⌘	Annex A (normative): Java Card SIM API
Other specs Affected:	⌘	[ ] Other core specifications ⌘ [ ] Test specifications [ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Annex A (normative): Java Card SIM API

The attached files "Annex_A_java.zip" and "Annex_A_HTML.zip" contains source files for the Java Card SIM API.

List of changes to the API html and java source files

Class sim.toolkit.EnvelopeHandler

```
/**
    * Looks for the Secured Data from the Command Packet in the first SMS TPDU
    * or Cell Broadcast Page Simple TLV contained in the Envelope handler. This can
    * be used on the events:
    * - EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, if the SMS TP-UD is formatted
    * according to TSGSM03.48 Single Short Message.
    * - EVENT_FORMATTED_SMS_CB, if the Cell Broadcast Page is formatted according to GSM 03.48.
    * If the element is available it becomes the TLV selected.
    *
    * @return the offset of the Secured Data first byte in the first SMS TPDU or Cell Broadcast Page TLV element. If the Secured Data length is zero the value returned shall be the offset of the first byte following the TS 03.48 Command Packet structure. If the Secured Data length is zero the value returned shall be the SMS TPDU TLV length.
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable SMS TPDU or Cell Broadcast Page TLV element or wrong data format </ul>
    */

    public short getSecuredDataOffset() throws ToolkitException {
        return 0;
    }
```