3GPP2 X.S0004-600-E
Version 1.0
Date: July 2005

Introduction to Procedures

Revision History

Revision		Date
Rev. 1	Initial Publication	July 2005

PART 600 INTRODUCTION

1 OBJECTIVE

This document presents the recommendation by which a roaming cellular subscriber may be provided with "Automatic Roaming," defined to comprise the following mechanisms:

1. Making the identity of the current serving, or visited system known to the home system.
2. Establishing financial responsibility for the roaming subscriber.
3. Establishing a valid roamer service profile in a visited system.
4. Providing for voice features and Short Message Services to the subscriber and providing of those feature and services while the subscriber is roaming.

The transactions are automatic in the sense that they operate in a manner that requires minimal intervention on the part of both the cellular subscriber and parties attempting to place a call to a cellular subscriber.

1.1 SCOPE

This document describes the signaling and call processing procedures required to perform automatic roaming features and services. The messages required to perform the automatic roaming are defined in Part 540.

Conformance with this document shall mean that a system's externally visible behavior is the same as the externally visible behavior of the abstract system described here.

1.2 ORGANIZATION

The 6XX Series are organized as follows:

1. This part, Part 600 provides an introduction to the 6xx Series.
2. Part 630, entitled “Basic Call Processing,” describes the basic call processing states, transitions, actions and the detection points for feature processing.
3. Part 640, entitled “Intersystem Procedures,” describes the procedures used between systems for handoff, mobility management, authentication, voice services and short message services.
4. Part 650, entitled “Common Voice Feature Procedures,” describes common procedures for voice feature control.
5. Part 651, entitled “Voice Feature Procedures,” describes the procedures for individual voice feature control.
6. Part 690, entitled “Operation Timer Values,” summarizes the operation timers used.
7. Part 691, entitled “Annexes” describes annex A, B, C, D and E of the 6xx Series.

1.3 DOCUMENTATION CONVENTIONS

The start of a task or procedure is always the first line of text.

<Comments> are statements enclosed in parentheses.

<label>: alone on line identifies a point in a task where control may be connected.

<Processing steps> are expressed as simple statements.

CASE <expression> OF…<value1>:…<value2>:… DEFAULT:… ENDCASE allows multiple branching with a single test in a task or procedure. The result of the expression is compared against the values in the value labels. If a match is found, the subordinate statements for that value are executed up to the next label, default statement or the end of the case, whichever comes first. When the last subordinate (indented) statement is executed, execution continues with the next statement after the ENDCASE statement at the same indentation level. Some license can be taken in specifying the value for the CASE labels. Multiple values that receive the same processing may be formatted as a comma separated list. For example:

1 CASE the color of roses OF:
2 *Red*:
2-1 A1.
2-2 A2.
3 *White, Grey, Purple*:
3-1 B1.
3-2 B2.
4 *DEFAULT*:
4-1 C1.
4-2 C2.
5 ENDCASE.
6 D1.

If roses are red, then A1, A2, and D1 are executed.

If roses are white, grey, or purple then B1, B2, and D1 are executed.

If roses are not red and not white and not grey and not purple then C1, C2, and D1 are executed.

Clear <parameter or variable> is used to reset the value of a message or state variable. If a binary value is not specified it may be assumed to use the value 0.

Discard <parameter> removes a received parameter to prevent it from being relayed.

Execute <procedure> causes a self contained procedure to be executed.

Exit causes the current task to terminate its own operation. The task and all associated variables cease to exist.

FOR <condition> <statements> ENDFOR is used to process multiple iterations of a particular sequence of subordinate processing steps.

GOTO <label> is used to transfer execution control to a specified label or task name.

IF <conditional>…ELSEIF <conditional>… ELSEIF <conditional>… ELSE… ENDIF allows branching in a task or procedure. When the condition is true, the subordinate statements are executed, otherwise execution proceeds with the next statement at the same indentation as the current statement. When the last subordinate (indented) statement is executed, execution continues with the statement after the ENDIF statement at the same indentation level. **ELSEIF** allows for testing of multiple related conditions. **ELSE** allows for processing when the condition(s) are false. For example:

7 IF roses are red then:

7-1 A1.

7-2 A2.

8 ELSEIF roses are white then:

8-1 B1.

8-2 B2.

9 ELSE:

9-1 C1.

9-2 C2.

10 ENDIF.

11 D1.

If roses are red then A1, A2, and D1 are executed.

If roses are not red, but are white then B1, B2, and D1 are executed.

If roses are not red and not white, then C1, C2, and D1 are executed.

Include <parameter> is used to include the particular parameter in an outgoing message.

Order is used to indicate the communication of a command to a mobile station (MS) using local defined procedures and procedures specific to the air interface being used by the mobile station.

Pass <message> to <destination> causes the named message to be transferred from the current task to the named destination task. The destination task must be on the same functional entity as the sender.

Process is used to indicate the execution of locally defined procedures that are currently not subject to standardization.

Relay <parameter> is used to include one or more received parameter(s) in the next outgoing message.

Remain in this state transfers execution control to the containing **WAIT** statement.

Replace <parameter> is used to change the values of an incoming parameter for use with the next outgoing message.

Return causes the current procedure to exit and return control to the calling task or procedure.

Send <message> to <destination> causes the named message to be transferred from the current task to the named destination task. The destination may be on any functional entity. RETURN RESULTs, RETURN ERRORs, or REJECTs are understood to be sent to the functional entity sending a corresponding INVOKE.

Set <parameter or variable> is used to change the value of a message parameter or an internal state variable. If a binary value is not specified it may be assumed to use the value 1.

Spawn <task> creates an instance of the named task which executes independently of the current task.

Start <timer> causes the named timer to be started from zero.

Stop <timer> causes the named timer to be stopped.

WAIT <state name or main event>…**WHEN** <event>…**WHEN** <event>…**ENDWAIT** identifies a state where a task waits for the occurrence of one or more possible events. Each WHEN identifies an event causing a state transition, such as, the arrival of a message or the expiry of a timer. All statements subordinate (indented) to the WHEN statement are executed. When the last subordinate (indented) statement is executed, execution continues with the statement after the ENDWAIT statement at the same indentation level. For example:

12 WAIT for message to arrive:

13 WHEN message arrives:

13-1 A1.

13-2 A2.

14 WHEN error response arrives:

14-1 B1.

14-2 B2.

15 WHEN timer expires:

15-1 C1.

15-2 C2.

16 ENDWAIT.

17 D1.

If the message arrives, then A1, A2, and D1 are executed.

If an error response arrives, then B1, B2, and D1 are executed.

If the timer expires, then C1, C2, and D1 are executed.

WHILE <condition> ENDWHILE is used to process zero or more iterations of a particular sequence of subordinate processing steps while the condition remains true. All subordinate (indented) statements are executed while the condition is true. When the condition is false, execution begins with the statement following the ENDWHILE statement.

WITH <**data reference**>: <**statement list**> **ENDWITH** is used to specify the data that is to be operated on by the enclosed statement list.

1.4 INTERSYSTEM HANDOFF PROCEDURES

All intersystem handoff messages and associated parameters are defined in Parts 540 and 550. Part 640 provides the transaction procedures defined for the Intersystem Handoff. Timer values are specified in Part 690.

Intersystem handoff uses dedicated voice and data facilities between the MSCs. Future versions may use other facilities. Handoffs are allowed to traverse from system to system with the total number of inter-MSC facilities in the sequence limited to the value of a parameter (MAXHANDOFF) which is programmed by the service providers. Path minimization is used to keep the number of MSCs involved in a call to a minimum. Handoff Back and Handoff-To-Third are actions taken to perform path minimization. The number of systems that can be involved in path minimization is limited to the value of a parameter (TANDEMDEPTH) which is programmed by the service providers.

This feature is manufacturer and system independent. This protocol does not preclude handoff between bands A and B within the same Cellular Geographic Serving Area (CGSA).

An off-hook MS is a unit that is currently involved in a call. Only off-hook MSs may handoff.

The procedures are based on the assumption that intersystem handoff relies upon dedicated intersystem trunks. This is required since intersystem handoff is a tightly controlled activity of the cellular systems involved. Intersystem handoff cannot be considered any differently than an inter-cell handoff. The handoff procedures specified in the following sections can be applied to consecutive inter-MSC handoffs for the same MS. A non-Anchor Serving MSC that has handed off a call becomes a Tandem MSC. The initial Serving MSC retains its role as an Anchor MSC.

1.5 SHORT MESSAGE SERVICE PROCEDURES

Short message procedures defined in Part 691are written in such a way as to allow the invoking of other procedures, especially at the MSC and the MC. Air interface messages and procedures are defined in informative Part 691 Annex D, so that all air interfaces can have common characteristics.

At the MSC, MS originated messages may be received over the air interface, from a Tandem MSC or from the Serving MSC as shown in the following figure. These MS originated messages may then be routed either toward the Anchor MSC or, at the Anchor MSC, toward the message's destination. Messages can either be initiated by an SME, an SME collocated with an HLR or MSC, an MS-based SME, or an MC.

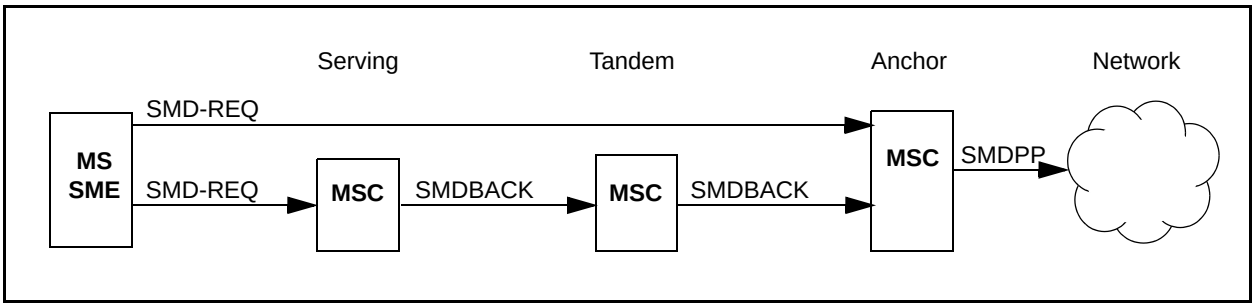

Figure 1 MS Originated SMS Delivery Point-to-Point Messages

The relationships of short message originations and the corresponding procedure is indicated in the following figure:

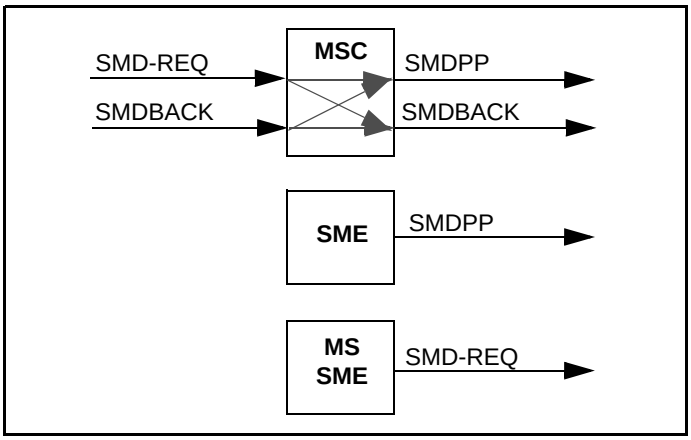

Figure 2 Originating SMS Delivery Point-to-Point Procedures

At the MSC MS terminating messages may be received from the network, from a Tandem MSC or from the Anchor MSC. These MS terminating messages may then be routed either toward the Serving MSC or, at the Serving MSC, toward the message's destination MS.

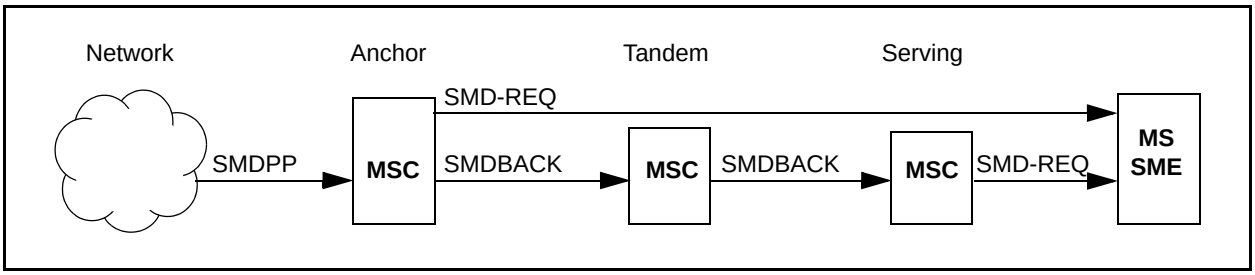

Figure 3 MS Terminated SMS Delivery Point-to-Point Messages

The relationships of short message terminations and the corresponding procedure is indicated in the following figure:

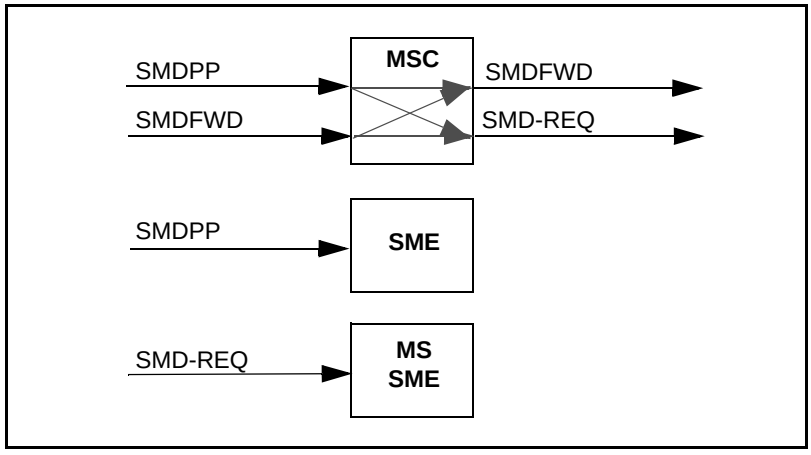

Figure 4 Terminating SMS Delivery Point-to-Point Procedures

Each of these tasks may be accepted, postponed or denied. An accepted request is one that has made it to its intended destination and that destination has responded. A postponed request is one that has been accepted by the network for later delivery to the intended destination. A denied request is one that is not accepted by the network or the intended destination. A cause is normally given with a denied request so that the sender can take appropriate action.

Messages to and from an SME may be processed by a network based Message Center (MC). The MC may provide Originating SMS Supplementary Services (OSSS) for messages originated by an SME by routing the messages to the SME's home MC (using indirect routing). Messages to an SME may be given Terminating SMS Supplementary Services (TSSS) by routing the message to the destination SME's home MC. It is possible for a message to have both originating and terminating supplementary services. These basic relationships are shown in the following figure.

There is only one home MC for each MS-based SME. This MC controls the delivery of messages to the MS-based SME as a TSSS. Notification for postponed delivery is only sent to this MC.

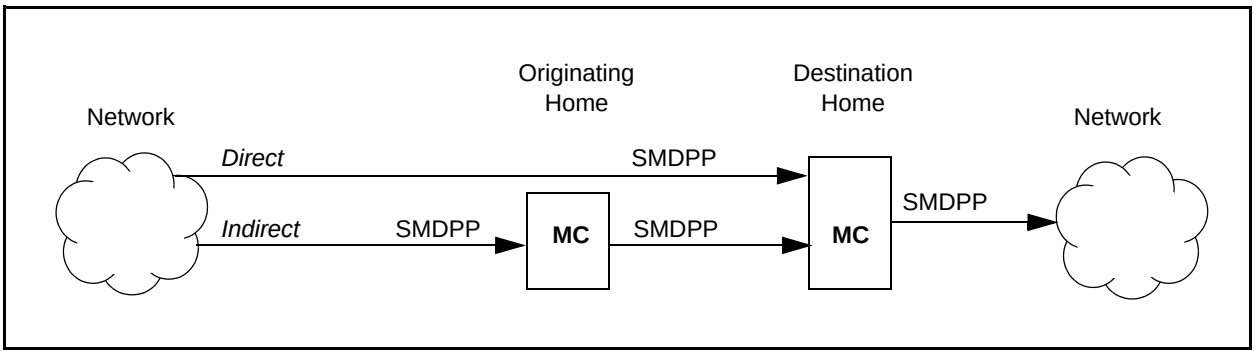

Figure 5 Message Center Processing of SMS Delivery Point-to-Point Messages

The relationships of Message Center messages and the corresponding procedure is indicated in the following figure:

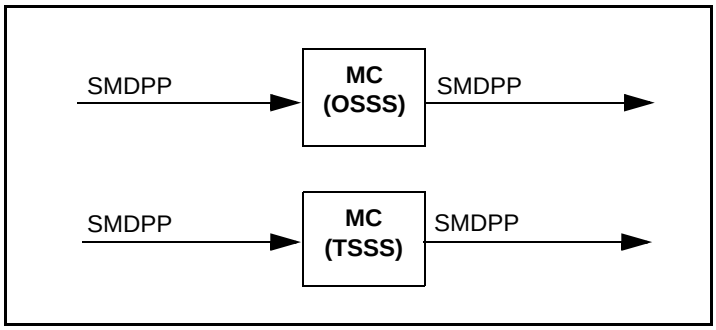

Figure 6 Message Center SMS Delivery Point-to-Point Procedures

These procedures are written to reduce the transmission of information. Destination and originating addresses may be carried by the underlying transport mechanism in some cases. However, the transport mechanism should not be used, if there is any possibility that it may change either of these addresses. The safest method is to always include the discrete address parameters.