3GPP TSG_CN#7 **NP-000027**
ETSI SMG3 Plenary Meeting #7,
Madrid, Spain
13th – 15th March 2000

Agenda item: **5.4.3**

Source: **TSG_N SS ad hoc**

Title: **CRs to 3G Work Item LCS**

Introduction:

This document contains **"2"** CRs on **LCS,** that have been agreed by **TSG_N SS ad hoc,** and are forwarded to **TSG_N Plenary** meeting #7 for approval.

Tdoc	Spec	CR	Rev	Cat	Rel	Old ver	New ver	Subject
NS-000051	04.80	A016	1	F	R98	7.1.0	7.2.0	Correction to Location Notification Type and LCS-MOLR errors
NS-000052	24.080	002	1	A	R99	3.1.0	3.2.0	Correction to Location Notification Type and LCS-MOLR errors

Document NS-000051

e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

04.80	CR	A016r1	Current Version:	7.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG-CN #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic	X	

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM		ME	X	UTRAN / Radio		Core Network	X

Source: SS ad hoc **Date:** 7 Jan 2000

Subject: Correction to Location Notification Type and LCS-MOLR errors

Work item: Location Services (LCS)

Category: *(only one category shall be marked with an X)*			**Release:**	
	Correction	X	Phase 2	
	Corresponds to a correction in an earlier release		Release 96	
	Addition of feature		Release 97	
	Functional modification of feature		Release 98	X
	Editorial modification		Release 99	
			Release 00	

Reason for change:

Location Notification Type is changed to have three values: notifyLocationAllowed, notifyAndVerifyLocationAllowed and notifyAndVerifyLocationNotAllowed. It describes the intended outcome of the notification procedure more clearly, i.e. there is less room for misinterpretations regarding default outcome in case of no response from the MS. The datatype NotificationToMSUser defined in GSM TS 09.02 is reused instead of defining the type in this specification.

Following new error codes are added to LCS-MOLR operation:

SS-SubscriptionViolation
FacilityNotSupported

Clauses affected: 4.2, 4.4.2

Other specs affected:				
	Other 3G core specifications	X	→ List of CRs:	09.02 A???
	Other GSM core specifications		→ List of CRs:	
	MS test specifications		→ List of CRs:	
	BSS test specifications		→ List of CRs:	
	O&M specifications		→ List of CRs:	

Other comments:

4.2 Operation types

Table 4.1 summarizes the operations defined for supplementary services in the present document and shows which of these operations are call related and call independent. The terms "call related" and "call independent" are defined in GSM 04.10.

Table 4.1: Relevance of supplementary service operations

Operation name	Call related SS	Call independent SS
RegisterSS	-	+
EraseSS	-	+
ActivateSS	-	+
DeactivateSS	-	+
InterrogateSS	-	+
RegisterPassword	-	+
GetPassword	-	+
ProcessUnstructuredSS-Data	+	+
ForwardCheckSS-Indication	-	+
ProcessUnstructuredSS-Request	-	+
UnstructuredSS-Request	-	+
UnstructuredSS-Notify	-	+
ForwardChargeAdvice	+	-
NotifySS	+	-
ForwardCUG-Info	+	-
BuildMPTY	+	-
HoldMPTY	+	-
RetrieveMPTY	+	-
SplitMPTY	+	-
ExplicitCT	+	-
AccessRegisterCCEntry	+	-
EraseCCEntry	-	+
CallDeflection	+	-
UserUserService	+	-
LCS-LocationNotification	-	+
LCS-MOLR	-	+
NOTE: The ProcessUnstructuredSS-Data operation may be used call related by a GSM Phase 1 MS.		

```
SS-Operations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Operations (0) version3 (3)}

DEFINITIONS ::=

BEGIN

EXPORTS

-- exports operation types

-- operations defined in the present document
ProcessUnstructuredSS-Data, NotifySS, ForwardChargeAdvice, ForwardCUG-Info, BuildMPTY, HoldMPTY,
RetrieveMPTY, SplitMPTY, ExplicitCT, AccessRegisterCCEntry, CallDeflection, UserUserService;

IMPORTS

OPERATION FROM
TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

-- The MAP operations:
-- RegisterSS, EraseSS, ActivateSS, DeactivateSS, InterrogateSS, RegisterPassword,
-- GetPassword, ProcessUnstructuredSS-Request, UnstructuredSS-Request, UnstructuredSS-Notify
-- ForwardCheckSS-Indication
-- are imported from MAP-Operations in SS-Protocol module.

-- imports SS-data types
NotifySS-Arg,
ForwardChargeAdviceArg,
ForwardCUG-InfoArg,
```

```
SS-UserData,
AccessRegisterCCEntryArg,
CallDeflectionArg,
UserUserServiceArg
FROM SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Datatypes (2) version3 (3)}

-- imports MAP-SS-data types
RegisterCC-EntryRes
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version4 (4)}

-- imports MAP-errors
IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
SS-Incompatibility, SystemFailure, FacilityNotSupported, CallBarred,UnexpectedDataValue,
ShortTermDenial, LongTermDenial, DataMissing, ForwardingViolation, ForwardingFailed,
PositionMethodFailure
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-Errors (10) version4 (4)}

-- imports SS-Errors
ResourcesNotAvailable, MaxNumberOfMPTY-ParticipantsExceeded,DeflectionToServedSubscriber,
InvalidDeflectedToNumber, SpecialServiceCode, RejectedByUser, RejectedByNetwork
FROM SS-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version3 (3)}
;

-- operation types definition

ProcessUnstructuredSS-Data::=   OPERATION -- Timer T(PUSSD)= 15s to 30s
    ARGUMENT
    ss-UserData     SS-UserData
    RESULT
    ss-UserData     SS-UserData
-- optional
    ERRORS{
    SystemFailure,
    UnexpectedDataValue}

NotifySS ::=    OPERATION
    ARGUMENT
    notifySS-Arg    NotifySS-Arg

ForwardChargeAdvice ::= OPERATION -- Timer T(AoC)= 1s to 40s
    ARGUMENT
    forwardChargeAdviceArg  ForwardChargeAdviceArg
    RESULT

ForwardCUG-Info ::= OPERATION
    ARGUMENT
    forwardCUG-InfoArg  ForwardCUG-InfoArg

BuildMPTY ::=   OPERATION -- Timer T(BuildMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    SystemFailure,
    ResourcesNotAvailable,
    MaxNumberOfMPTY-ParticipantsExceeded}

HoldMPTY ::=    OPERATION -- Timer T(HoldMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure}

RetrieveMPTY ::=    OPERATION -- Timer T(RetrieveMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
```

```
    SystemFailure}

SplitMPTY ::=   OPERATION -- Timer T(SplitMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure}

ExplicitCT ::=  OPERATION -- Timer T(ECT)= 5s to 15s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    CallBarred}

AccessRegisterCCEntry ::= OPERATION -- Timer T(AccRegCCEntry)= 30s
    ARGUMENT
    accessRegisterCCEntryArg    AccessRegisterCCEntryArg
    RESULT
    registerCCEntryRes  RegisterCC-EntryRes
    ERRORS{
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    CallBarred,
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    ShortTermDenial,
    LongTermDenial,
    FacilityNotSupported}

-- the timer value is defined by T308, see also in GSM 04.08 for definition of timer T308

CallDeflection ::=  OPERATION -- Timer T(CD)= 30s
    ARGUMENT
    callDeflectionArg   CallDeflectionArg
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    ForwardingViolation,
    CallBarred,
    DeflectionToServedSubscriber,
    InvalidDeflectedToNumber,
    SpecialServiceCode,
    ForwardingFailed}

-- the timer value is defined by T305, see also in GSM 04.08 for definition of timer T305
-- extensionContainer shall not be used with this operation

UserUserService ::= OPERATION -- Timer T(UUS3)= 10s
    ARGUMENT
    userUserServiceArg  UserUserServiceArg
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    RejectedByNetwork,
    RejectedByUser}

-- The timer value for UUS3 is 10s; it is applicable only if UUS3 is activated by FACILITY
-- message. If UUS service (UUS1, UUS2 or UUS3) is activated by SETUP message, no timers are
-- needed. In those cases Return Result or Return Error must be received within certain call
-- control messages, see GSM 04.87.
```

```
-- extensionContainer shall not be used with this operation.


LCS-LocationNotification::= OPERATION -- Timer T(LCSN)= 10s to 20s
    ARGUMENT
    locationNotificationArg     LocationNotificationArg
    RESULT
    locationNotificationRes     LocationNotificationRes
    ERRORS{
    SystemFailure,
    UnexpectedDataValue}


LCS-MOLR::= OPERATION -- Timer T(LCSL)= 10s to 30s
    ARGUMENT
    lcs-MOLRArg     LCS-MOLRArg
    RESULT
    lcs-MOLRRes     LCS-MOLRRes
    ERRORS{
    SystemFailure,
    UnexpectedDataValue,
    DataMissing,
    FacilityNotSupported,
    SS-SubscriptionViolation,
    PositionMethodFailure}

END
```

4.3 Error types

4.3.1 Error types ASN.1 specification

The following ASN.1 module provides an ASN.1 specification of errors. Errors from MAP are imported in the SS-Protocol module in subclause 4.5.

```
SS-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version3 (3)}

DEFINITIONS ::=

BEGIN

IMPORTS

ERROR FROM
TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)};

-- The MAP errors
-- UnknownSubscriber, BearerServiceNotProvisioned, TeleserviceNotProvisioned,
-- IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
-- SS-Incompatibility, SystemFailure, DataMissing, UnexpectedDataValue, FacilityNotSupported,
-- PW-RegistrationFailure, NegativePW-Check, CallBarred, NumberOfPW-AttemptsViolation,
-- AbsentSubscriber, IllegalSubscriber, IllegalEquipment, USSD-Busy, UnknownAlphabet,
-- ForwardingViolation, ForwardingFailed
-- are imported from MAP-Errors in SS-Protocol module.

-- error types definition
ResourcesNotAvailable ::= ERROR
MaxNumberOfMPTY-ParticipantsExceeded ::= ERROR
InvalidDeflectedToNumber ::= ERROR
SpecialServiceCode ::= ERROR
DeflectionToServedSubscriber ::= ERROR
RejectedByNetwork ::= ERROR
RejectedByUser ::= ERROR

END
```

4.3.2 Error types description

For each error type this subclause provides a brief prose description.

4.3.2.1 UnknownSubscriber

This error is returned by the network when it is requested to perform an operation concerning an unknown subscriber.

4.3.2.2 BearerServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested bearer service group has been subscribed to.

4.3.2.3 TeleServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested teleservice group has been subscribed to.

4.3.2.4 IllegalSS-Operation

This error is returned by the network when it is requested to perform an illegal operation which is defined as not applicable for the relevant supplementary service(s) (e.g. registration request for a service which must be registered by the administration). For the definition of the allowed operations for the individual supplementary services, see GSM 04.8x and 04.9x-series of technical specifications.

4.3.2.5 SS-ErrorStatus

This error is returned by the network when it is requested to perform an operation which is not compatible with the current status of the relevant supplementary service. The current status may be given as additional information by use of the SS-parameter.

4.3.2.6 SS-NotAvailable

This error is returned by the network when it is requested to perform an operation on a supplementary service which is not available in the current location area.

4.3.2.7 SS-SubscriptionViolation

This error is returned by the network when it is requested to perform an operation on a supplementary service, transgressing the subscription restrictions. The nature of the restriction or the transgressed options may be sent as parameters.

****** NEXT MODIFIED SECTION ******

4.4 Data types and identifiers

4.4.1 General

The data types used in the SS protocol specifications are described in the ASN.1 module provided in subclause 4.4.2, while subclause 4.4.3 provides an overview of the identifiers used in SS ASN.1 specifications.

Since size constraints are subject to modifications named values have been defined in the following module for the upper boundaries of the value ranges associated to several sub-type specifications.

4.4.2 ASN.1 data types

This subclause provides an ASN.1 module defining the abstract data types in operations and errors specification. Only data types which are specific for the present document are defined. All other data types are imported from MAP together with the import of operations and errors.

```
SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-DataTypes (2) version3 (3)}

DEFINITIONS

IMPLICIT TAGS ::=

BEGIN

-- exports all data types defined in this module

IMPORTS

SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-Code (15) version3 (3)}

-- imports MAP-SS-DataTypes
SS-Status, USSD-DataCodingScheme, USSD-String, CCBS-Feature
-- USSD-DataCodingScheme, USSD-String were introduced because of CNAP.
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-DataTypes (14) version4 (4)}

CUG-Index,
NotificationToMSUser
FROM MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-MS-DataTypes (11) version53 (53)}

maxSignalInfoLength,
ISDN-AddressString,
ISDN-SubaddressString,
AlertingPattern,
LCSClientExternalID
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-CommonDataTypes (18) version54 (54)}

LocationType,
LCSClientName,
LCS-QoS,
Horizontal-Accuracy,
ResponseTime,
Ext-GeographicalInformationFROM MAP-LCS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version5 (5)}

;

-- data types definition
```

```
SS-UserData ::= IA5String (SIZE (1.. maxSignalInfoLength))

NotifySS-Arg ::= SEQUENCE{
    ss-Code                     [1]     SS-Code OPTIONAL,
    ss-Status                   [4]     SS-Status OPTIONAL,
    ss-Notification             [5]     SS-Notification OPTIONAL,
    callIsWaiting-Indicator     [14]    NULL OPTIONAL,
    callOnHold-Indicator        [15]    CallOnHold-Indicator OPTIONAL,
    mpty-Indicator              [16]    NULL OPTIONAL,
    cug-Index                   [17]    CUG-Index OPTIONAL,
    clirSuppressionRejected     [18]    NULL OPTIONAL,
    ... ,
    ect-Indicator               [19]    ECT-Indicator OPTIONAL,
    nameIndicator               [20]    NameIndicator OPTIONAL,
    ccbs-Feature                [21]    CCBS-Feature OPTIONAL,
    alertingPattern             [22]    AlertingPattern OPTIONAL}

--  The nameIndicator is defined because of CNAP.

ForwardChargeAdviceArg ::= SEQUENCE{
    ss-Code                     [0]     SS-Code,
    chargingInformation         [1]     ChargingInformation,
    ...}

SS-Notification ::= OCTET STRING (SIZE (1))

--    Bit 8 7 6 5 4 00000 (Unused)

--  Bit 3   Call is forwarded indication to A-subscriber
--      (calling subscriber)
--  0   No information content
--  1   Outgoing call has been forwarded to C

--  Bit 2   Call is forwarded indication to B-subscriber
--      (forwarding subscriber)
--  0   No information content
--  1   Incoming call has been forwarded to C

--  Bit 1   Call is forwarded indication to C-subscriber
--      (forwarded-to subscriber)
--  0   No information content
--  1   Incoming call is a forwarded call

ChargingInformation ::= SEQUENCE{
    e1  [1] E1 OPTIONAL,
    e2  [2] E2 OPTIONAL,
    e3  [3] E3 OPTIONAL,
    e4  [4] E4 OPTIONAL,
    e5  [5] E5 OPTIONAL,
    e6  [6] E6 OPTIONAL,
    e7  [7] E7 OPTIONAL,
    ...}

E1 ::= INTEGER (0..max10TimesUnitsPerTime)
max10TimesUnitsPerTime INTEGER ::= 8191

E2 ::= INTEGER (0..max10TimesTimeInterval)
max10TimesTimeInterval INTEGER ::= 8191

E3 ::= INTEGER (0..max100TimesScalingFactor)
max100TimesScalingFactor INTEGER ::= 8191

E4 ::= INTEGER (0..max10TimesIncrement)
max10TimesIncrement INTEGER ::= 8191

E5 ::= INTEGER (0..max10TimesIncrementPerDataInterval)
max10TimesIncrementPerDataInterval INTEGER ::= 8191

E6 ::= INTEGER (0..maxNumberOfSegmentsPerDataInterval)
maxNumberOfSegmentsPerDataInterval INTEGER ::= 8191

E7 ::= INTEGER (0..max10TimesInitialTime)
max10TimesInitialTime INTEGER ::= 8191

CallOnHold-Indicator    ::= ENUMERATED {
            callRetrieved (0),
            callOnHold (1)}

ForwardCUG-InfoArg ::= SEQUENCE {
    cug-Index           [0] CUG-Index OPTIONAL,
    suppressPrefCUG     [1] NULL OPTIONAL,
```

```
    suppressOA          [2] NULL OPTIONAL,
    ...}

ECT-Indicator   ::= SEQUENCE {
    ect-CallState       [0] ECT-CallState,
    rdn [1] RDN OPTIONAL,
    ...}

ECT-CallState   ::= ENUMERATED {
        alerting (0),
        active (1)}

    NameIndicator ::= SEQUENCE {
        callingName     [0] Name OPTIONAL,
        ...}

    Name ::= CHOICE {
        namePresentationAllowed     [0] NameSet,
        presentationRestricted      [1] NULL,
        nameUnavailable             [2] NULL,
        namePresentationRestricted  [3] NameSet}

    NameSet ::= SEQUENCE {
        dataCodingScheme        [0] USSD-DataCodingScheme,
        lengthInCharacters      [1] INTEGER,
        nameString              [2] USSD-String,
        ...}

-- NameIndicator, Name and NameSet are defined because of CNAP.
-- The USSD-DataCodingScheme shall indicate use of the default alphabet through the
-- following encoding:
--    bit  7   6   5   4   3   2   1   0
--       | 0   0   0   0 | 1   1   1   1|

RDN ::= CHOICE {
    presentationAllowedAddress              [0] RemotePartyNumber,
    presentationRestricted                  [1] NULL,
    numberNotAvailableDueToInterworking     [2] NULL,
    presentationRestrictedAddress           [3] RemotePartyNumber}

RemotePartyNumber   ::= SEQUENCE {
    partyNumber             [0] ISDN-AddressString,
    partyNumberSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

AccessRegisterCCEntryArg    ::= SEQUENCE {
    ...}

CallDeflectionArg   ::= SEQUENCE {
    deflectedToNumber       [0] ISDN-AddressString,
    deflectedToSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

UserUserServiceArg ::= SEQUENCE {
    uUS-Service     [0] UUS-Service,
    uUS-Required    [1] BOOLEAN,
    ... }

UUS-Service ::= ENUMERATED {
    uUS1 (1),
    uUS2 (2),
    uUS3 (3),
    ... }

-- exception handling:
-- In case of UUS-Service with any other value, indicated as "UUS required",
-- but not understood by the MS, the call will be cleared.

LocationNotificationArg ::= SEQUENCE {
    notificationType    [0] NotificationToMSUserNotificationType,
    locationType        [1] LocationType,
    lcsClientExternalID [2] LCSClientExternalID     OPTIONAL,
    lcsClientName       [3] LCSClientName           OPTIONAL,
    ...}
-- exception handling:
-- At reception of an unrecognised notificationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.
-- At reception of an unrecognised locationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.

NotificationType::= ENUMERATED {
    notification        (0),
    privacyVerification (1),
```

```
    ... }
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

LocationNotificationRes ::= SEQUENCE {
    verificationResponse        [0] VerificationResponse OPTIONAL,
    ...}

VerificationResponse::= ENUMERATED {
    permissionDenied    (0),
    permissionGranted   (1),
    ... }

-- exception handling:
-- an unrecognized value shall be treated the same as value 0 (permissionDenied)

LCS-MOLRArg ::= SEQUENCE {
    molr-Type           [0] MOLR-Type,
    locationMethod      [1] LocationMethod          OPTIONAL,
    lcs-QoS             [2] LCS-QoS                 OPTIONAL,
    lcsClientExternalID [3] LCSClientExternalID     OPTIONAL,
    mlc-Number          [4] ISDN-AddressString      OPTIONAL,
    gpsAssistanceData   [5] GPSAssistanceData       OPTIONAL,
    ...}
-- The parameter locationMethod shall be included if and only if the molr-Type is set to value
-- deCipheringKeys or assistanceData.
-- The parameter gpsAssistanceData shall be included if and only if the molr-Type is set to value
-- assistanceData and LocationMethod is set to value assistedGPS.

MOLR-Type::= ENUMERATED {
    locationEstimate            (0),
    assistanceData              (1),
    deCipheringKeys             (2),
    ... }
-- exception handling:
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

LocationMethod::= ENUMERATED {
    msBasedEOTD         (0),
    msAssistedEOTD      (1),
    assistedGPS         (2),
    ... }
-- exception handling:
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

GPSAssistanceData::= OCTET STRING (SIZE (1..38))
-- Octets 1 to 38 are coded in the same way as the octets 3 to 7+2n of Requested GPS Data IE
-- in GSM 09.31.

LCS-MOLRRes::= SEQUENCE {
    locationEstimate        [0] Ext-GeographicalInformation     OPTIONAL,
    decipheringKeys         [1] DecipheringKeys                 OPTIONAL,
    ...}
-- Parameter locationEstimate shall be included if and only if the
-- molr-Type in LocationRequestArg was set to value locationEstimate.
-- Parameter decipheringKeys shall be included if and only if the molr-Type
-- in LocationRequestArg was set to value deCipheringKeys.
--


DecipheringKeys::= OCTET STRING (SIZE (15))
-- Octets in DecipheringKeys are coded in the same way as the octets 3 to 17 of Deciphering Key IE
-- in GSM 09.31. I.e. these octets contain Current Deciphering Key, Next Deciphering Key and
-- Ciphering Key Flag.



END
```

Document NS-000052

e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

24.080	CR	002r1	Current Version:	3.1.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to: CN#7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑	for information		non-strategic	X	

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

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

Source: SS ad hoc **Date:** 7 Jan 1999

Subject: Correction to Location Notification Type and LCS-MOLR errors

Work item: Location Services (LCS)

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

		Release:	
Correction		Phase 2	
Corresponds to a correction in an earlier release	X	Release 96	
Addition of feature		Release 97	
Functional modification of feature		Release 98	
Editorial modification		Release 99	X
		Release 00	

Reason for change:

Location Notification Type is changed to have three values: notifyLocationAllowed, notifyAndVerifyLocationAllowed and notifyAndVerifyLocationNotAllowed. It describes the intended outcome of the notification procedure more clearly, i.e. there is less room for misinterpretations regarding default outcome in case of no response from the MS. The datatype NotificationToMSUser defined in GSM TS 09.02 is reused instead of defining the type in this specification.

Following new error codes are added to LCS-MOLR operation:

SS-SubscriptionViolation
FacilityNotSupported

Clauses affected: 4.2, 4.4.2

Other specs affected:

Other 3G core specifications		→ List of CRs:	
Other GSM core specifications	X	→ List of CRs:	09.02 A???
MS test specifications		→ List of CRs:	
BSS test specifications		→ List of CRs:	
O&M specifications		→ List of CRs:	

Other comments:

Note that there wasn't official baseline version available at the time of drafting the CR. "Current version"-field should be filled as soon as new baseline version after TSG-CN#6 is available.

****** NEXT MODIFIED SECTION ******

4.2 Operation types

Table 4.1 summarizes the operations defined for supplementary services in the present document and shows which of these operations are call related and call independent. The terms "call related" and "call independent" are defined in GSM 04.10.

Table 4.1: Relevance of supplementary service operations

Operation name	Call related SS	Call independent SS
RegisterSS	-	+
EraseSS	-	+
ActivateSS	-	+
DeactivateSS	-	+
InterrogateSS	-	+
RegisterPassword	-	+
GetPassword	-	+
ProcessUnstructuredSS-Data	+	+
ForwardCheckSS-Indication	-	+
ProcessUnstructuredSS-Request	-	+
UnstructuredSS-Request	-	+
UnstructuredSS-Notify	-	+
ForwardChargeAdvice	+	-
NotifySS	+	-
ForwardCUG-Info	+	-
BuildMPTY	+	-
HoldMPTY	+	-
RetrieveMPTY	+	-
SplitMPTY	+	-
ExplicitCT	+	-
AccessRegisterCCEntry	+	-
EraseCCEntry	-	+
CallDeflection	+	-
UserUserService	+	-
LCS-LocationNotification	-	+
LCS-MOLR	-	+
NOTE: The ProcessUnstructuredSS-Data operation may be used call related by a GSM Phase 1 MS.		

```
SS-Operations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Operations (0) version3 (3)}

DEFINITIONS ::=

BEGIN

EXPORTS

-- exports operation types

-- operations defined in the present document
ProcessUnstructuredSS-Data, NotifySS, ForwardChargeAdvice, ForwardCUG-Info, BuildMPTY, HoldMPTY,
RetrieveMPTY, SplitMPTY, ExplicitCT, AccessRegisterCCEntry, CallDeflection, UserUserService;

IMPORTS

OPERATION FROM
TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

-- The MAP operations:
-- RegisterSS, EraseSS, ActivateSS, DeactivateSS, InterrogateSS, RegisterPassword,
-- GetPassword, ProcessUnstructuredSS-Request, UnstructuredSS-Request, UnstructuredSS-Notify
-- ForwardCheckSS-Indication
-- are imported from MAP-Operations in SS-Protocol module.
```

```
-- imports SS-data types
NotifySS-Arg,
ForwardChargeAdviceArg,
ForwardCUG-InfoArg,
SS-UserData,
AccessRegisterCCEntryArg,
CallDeflectionArg,
UserUserServiceArg
FROM SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Datatypes (2) version3 (3)}

-- imports MAP-SS-data types
RegisterCC-EntryRes
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version4 (4)}

-- imports MAP-errors
IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
SS-Incompatibility, SystemFailure, FacilityNotSupported, CallBarred,UnexpectedDataValue,
ShortTermDenial, LongTermDenial, DataMissing, ForwardingViolation, ForwardingFailed,
PositionMethodFailure
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-Errors (10) version4 (4)}

-- imports SS-Errors
ResourcesNotAvailable, MaxNumberOfMPTY-ParticipantsExceeded,DeflectionToServedSubscriber,
InvalidDeflectedToNumber, SpecialServiceCode, RejectedByUser, RejectedByNetwork
FROM SS-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version3 (3)}
;

-- operation types definition

ProcessUnstructuredSS-Data::=   OPERATION -- Timer T(PUSSD)= 15s to 30s
    ARGUMENT
    ss-UserData     SS-UserData
    RESULT
    ss-UserData     SS-UserData
-- optional
    ERRORS{
    SystemFailure,
    UnexpectedDataValue}

NotifySS ::=    OPERATION
    ARGUMENT
    notifySS-Arg    NotifySS-Arg

ForwardChargeAdvice ::= OPERATION -- Timer T(AoC)= 1s to 40s
    ARGUMENT
    forwardChargeAdviceArg  ForwardChargeAdviceArg
    RESULT

ForwardCUG-Info ::= OPERATION
    ARGUMENT
    forwardCUG-InfoArg  ForwardCUG-InfoArg

BuildMPTY ::=   OPERATION -- Timer T(BuildMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    SystemFailure,
    ResourcesNotAvailable,
    MaxNumberOfMPTY-ParticipantsExceeded}

HoldMPTY ::=    OPERATION -- Timer T(HoldMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure}

RetrieveMPTY ::=    OPERATION -- Timer T(RetrieveMPTY)= 5s to 30s
    RESULT
    ERRORS{
```

```
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure}

SplitMPTY ::=   OPERATION -- Timer T(SplitMPTY)= 5s to 30s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure}

ExplicitCT ::=  OPERATION -- Timer T(ECT)= 5s to 15s
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    CallBarred}

AccessRegisterCCEntry ::= OPERATION -- Timer T(AccRegCCEntry)= 30s
    ARGUMENT
    accessRegisterCCEntryArg    AccessRegisterCCEntryArg
    RESULT
    registerCCEntryRes  RegisterCC-EntryRes
    ERRORS{
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    CallBarred,
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-Incompatibility,
    ShortTermDenial,
    LongTermDenial,
    FacilityNotSupported}

-- the timer value is defined by T308, see also in GSM 04.08 for definition of timer T308

CallDeflection ::=  OPERATION -- Timer T(CD)= 30s
    ARGUMENT
    callDeflectionArg   CallDeflectionArg
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    ForwardingViolation,
    CallBarred,
    DeflectionToServedSubscriber,
    InvalidDeflectedToNumber,
    SpecialServiceCode,
    ForwardingFailed}

-- the timer value is defined by T305, see also in GSM 04.08 for definition of timer T305
-- extensionContainer shall not be used with this operation

UserUserService ::= OPERATION -- Timer T(UUS3)= 10s
    ARGUMENT
    userUserServiceArg  UserUserServiceArg
    RESULT
    ERRORS{
    IllegalSS-Operation,
    SS-ErrorStatus,
    SS-NotAvailable,
    SS-Incompatibility,
    FacilityNotSupported,
    SystemFailure,
    ResourcesNotAvailable,
    RejectedByNetwork,
    RejectedByUser}
```

```
-- The timer value for UUS3 is 10s; it is applicable only if UUS3 is activated by FACILITY
-- message. If UUS service (UUS1, UUS2 or UUS3) is activated by SETUP message, no timers are
-- needed. In those cases Return Result or Return Error must be received within certain call
-- control messages, see GSM 04.87.
-- extensionContainer shall not be used with this operation.


LCS-LocationNotification::= OPERATION -- Timer T(LCSN)= 10s to 20s
    ARGUMENT
    locationNotificationArg     LocationNotificationArg
    RESULT
    locationNotificationRes     LocationNotificationRes
    ERRORS{
    SystemFailure,
    UnexpectedDataValue}


LCS-MOLR::= OPERATION -- Timer T(LCSL)= 10s to 30s
    ARGUMENT
    lcs-MOLRArg     LCS-MOLRArg
    RESULT
    lcs-MOLRRes     LCS-MOLRRes
    ERRORS{
    SystemFailure,
    UnexpectedDataValue,
    DataMissing,
    FacilityNotSupported,
    SS-SubscriptionViolation,
    PositionMethodFailure}

END
```

****** NEXT MODIFIED SECTION ******

4.3 Error types

4.3.1 Error types ASN.1 specification

The following ASN.1 module provides an ASN.1 specification of errors. Errors from MAP are imported in the SS-Protocol module in subclause 4.5.

```
SS-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version3 (3)}

DEFINITIONS ::=

BEGIN

IMPORTS

ERROR FROM
TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)};

-- The MAP errors
-- UnknownSubscriber, BearerServiceNotProvisioned, TeleserviceNotProvisioned,
-- IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
-- SS-Incompatibility, SystemFailure, DataMissing, UnexpectedDataValue, FacilityNotSupported,
-- PW-RegistrationFailure, NegativePW-Check, CallBarred, NumberOfPW-AttemptsViolation,
-- AbsentSubscriber, IllegalSubscriber, IllegalEquipment, USSD-Busy, UnknownAlphabet,
-- ForwardingViolation, ForwardingFailed
-- are imported from MAP-Errors in SS-Protocol module.

-- error types definition
ResourcesNotAvailable ::= ERROR
MaxNumberOfMPTY-ParticipantsExceeded ::= ERROR
InvalidDeflectedToNumber ::= ERROR
SpecialServiceCode ::= ERROR
DeflectionToServedSubscriber ::= ERROR
RejectedByNetwork ::= ERROR
RejectedByUser ::= ERROR

END
```

4.3.2 Error types description

For each error type this subclause provides a brief prose description.

4.3.2.1 UnknownSubscriber

This error is returned by the network when it is requested to perform an operation concerning an unknown subscriber.

4.3.2.2 BearerServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested bearer service group has been subscribed to.

4.3.2.3 TeleServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested teleservice group has been subscribed to.

4.3.2.4 IllegalSS-Operation

This error is returned by the network when it is requested to perform an illegal operation which is defined as not applicable for the relevant supplementary service(s) (e.g. registration request for a service which must be registered by

the administration). For the definition of the allowed operations for the individual supplementary services, see GSM 04.8x and 04.9x-series of technical specifications.

4.3.2.5 SS-ErrorStatus

This error is returned by the network when it is requested to perform an operation which is not compatible with the current status of the relevant supplementary service. The current status may be given as additional information by use of the SS-parameter.

4.3.2.6 SS-NotAvailable

This error is returned by the network when it is requested to perform an operation on a supplementary service which is not available in the current location area.

4.3.2.7 SS-SubscriptionViolation

This error is returned by the network when it is requested to perform an operation on a supplementary service, transgressing the subscription restrictions. The nature of the restriction or the transgressed options may be sent as parameters.

****** NEXT MODIFIED SECTION ******

4.4 Data types and identifiers

4.4.1 General

The data types used in the SS protocol specifications are described in the ASN.1 module provided in subclause 4.4.2, while subclause 4.4.3 provides an overview of the identifiers used in SS ASN.1 specifications.

Since size constraints are subject to modifications named values have been defined in the following module for the upper boundaries of the value ranges associated to several sub-type specifications.

4.4.2 ASN.1 data types

This subclause provides an ASN.1 module defining the abstract data types in operations and errors specification. Only data types which are specific for the present document are defined. All other data types are imported from MAP together with the import of operations and errors.

```
SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-DataTypes (2) version3 (3)}

DEFINITIONS

IMPLICIT TAGS ::=

BEGIN

-- exports all data types defined in this module

IMPORTS

SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-Code (15) version3 (3)}

-- imports MAP-SS-DataTypes
SS-Status, USSD-DataCodingScheme, USSD-String, CCBS-Feature
-- USSD-DataCodingScheme, USSD-String were introduced because of CNAP.
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-DataTypes (14) version4 (4)}

CUG-Index,
NotificationToMSUser
FROM MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-MS-DataTypes (11) version53 (53)}

maxSignalInfoLength,
ISDN-AddressString,
ISDN-SubaddressString,
AlertingPattern,
LCSClientExternalID
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-CommonDataTypes (18) version54 (54)}

LocationType,
LCSClientName,
LCS-QoS,
Horizontal-Accuracy,
ResponseTime,
Ext-GeographicalInformationFROM MAP-LCS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version5 (5)}

;

-- data types definition

SS-UserData ::= IA5String (SIZE (1.. maxSignalInfoLength))
```

```
NotifySS-Arg ::= SEQUENCE{
    ss-Code                     [1]     SS-Code OPTIONAL,
    ss-Status                   [4]     SS-Status OPTIONAL,
    ss-Notification             [5]     SS-Notification OPTIONAL,
    callIsWaiting-Indicator     [14]    NULL OPTIONAL,
    callOnHold-Indicator        [15]    CallOnHold-Indicator OPTIONAL,
    mpty-Indicator              [16]    NULL OPTIONAL,
    cug-Index                   [17]    CUG-Index OPTIONAL,
    clirSuppressionRejected     [18]    NULL OPTIONAL,
    ... ,
    ect-Indicator               [19]    ECT-Indicator OPTIONAL,
    nameIndicator               [20]    NameIndicator OPTIONAL,
    ccbs-Feature                [21]    CCBS-Feature OPTIONAL,
    alertingPattern             [22]    AlertingPattern OPTIONAL}

--  The nameIndicator is defined because of CNAP.

ForwardChargeAdviceArg ::= SEQUENCE{
    ss-Code                     [0]     SS-Code,
    chargingInformation         [1]     ChargingInformation,
    ...}

SS-Notification ::= OCTET STRING (SIZE (1))

--    Bit 8 7 6 5 4 00000 (Unused)

--  Bit 3   Call is forwarded indication to A-subscriber
--      (calling subscriber)
--  0   No information content
--  1   Outgoing call has been forwarded to C

--  Bit 2   Call is forwarded indication to B-subscriber
--      (forwarding subscriber)
--  0   No information content
--  1   Incoming call has been forwarded to C

--  Bit 1   Call is forwarded indication to C-subscriber
--      (forwarded-to subscriber)
--  0   No information content
--  1   Incoming call is a forwarded call

ChargingInformation ::= SEQUENCE{
    e1  [1] E1 OPTIONAL,
    e2  [2] E2 OPTIONAL,
    e3  [3] E3 OPTIONAL,
    e4  [4] E4 OPTIONAL,
    e5  [5] E5 OPTIONAL,
    e6  [6] E6 OPTIONAL,
    e7  [7] E7 OPTIONAL,
    ...}

E1 ::= INTEGER (0..max10TimesUnitsPerTime)
max10TimesUnitsPerTime INTEGER ::= 8191

E2 ::= INTEGER (0..max10TimesTimeInterval)
max10TimesTimeInterval INTEGER ::= 8191

E3 ::= INTEGER (0..max100TimesScalingFactor)
max100TimesScalingFactor INTEGER ::= 8191

E4 ::= INTEGER (0..max10TimesIncrement)
max10TimesIncrement INTEGER ::= 8191

E5 ::= INTEGER (0..max10TimesIncrementPerDataInterval)
max10TimesIncrementPerDataInterval INTEGER ::= 8191

E6 ::= INTEGER (0..maxNumberOfSegmentsPerDataInterval)
maxNumberOfSegmentsPerDataInterval INTEGER ::= 8191

E7 ::= INTEGER (0..max10TimesInitialTime)
max10TimesInitialTime INTEGER ::= 8191

CallOnHold-Indicator    ::= ENUMERATED {
            callRetrieved (0),
            callOnHold (1)}

ForwardCUG-InfoArg ::= SEQUENCE {
    cug-Index           [0] CUG-Index OPTIONAL,
    suppressPrefCUG     [1] NULL OPTIONAL,
    suppressOA          [2] NULL OPTIONAL,
    ...}
```

```
ECT-Indicator   ::= SEQUENCE {
    ect-CallState       [0] ECT-CallState,
    rdn [1] RDN OPTIONAL,
    ...}

ECT-CallState   ::= ENUMERATED {
        alerting (0),
        active (1)}

    NameIndicator ::= SEQUENCE {
        callingName     [0] Name OPTIONAL,
        ...}

    Name ::= CHOICE {
        namePresentationAllowed     [0] NameSet,
        presentationRestricted      [1] NULL,
        nameUnavailable             [2] NULL,
        namePresentationRestricted  [3] NameSet}

    NameSet ::= SEQUENCE {
        dataCodingScheme        [0] USSD-DataCodingScheme,
        lengthInCharacters      [1] INTEGER,
        nameString              [2] USSD-String,
        ...}

-- NameIndicator, Name and NameSet are defined because of CNAP.
-- The USSD-DataCodingScheme shall indicate use of the default alphabet through the
-- following encoding:
--    bit  7   6   5   4   3   2   1   0
--       | 0   0   0   0 | 1   1   1   1|

RDN ::= CHOICE {
    presentationAllowedAddress              [0] RemotePartyNumber,
    presentationRestricted                  [1] NULL,
    numberNotAvailableDueToInterworking     [2] NULL,
    presentationRestrictedAddress           [3] RemotePartyNumber}

RemotePartyNumber   ::= SEQUENCE {
    partyNumber             [0] ISDN-AddressString,
    partyNumberSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

AccessRegisterCCEntryArg    ::= SEQUENCE {
    ...}

CallDeflectionArg   ::= SEQUENCE {
    deflectedToNumber       [0] ISDN-AddressString,
    deflectedToSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

UserUserServiceArg ::= SEQUENCE {
    uUS-Service     [0] UUS-Service,
    uUS-Required    [1] BOOLEAN,
    ... }

UUS-Service ::= ENUMERATED {
    uUS1 (1),
    uUS2 (2),
    uUS3 (3),
    ... }

-- exception handling:
-- In case of UUS-Service with any other value, indicated as "UUS required",
-- but not understood by the MS, the call will be cleared.

LocationNotificationArg ::= SEQUENCE {
    notificationType    [0] NotificationToMSUserNotificationType,
    locationType        [1] LocationType,
    lcsClientExternalID [2] LCSClientExternalID     OPTIONAL,
    lcsClientName       [3] LCSClientName           OPTIONAL,
    ...}
-- exception handling:
-- At reception of an unrecognised notificationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.
-- At reception of an unrecognised locationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.

NotificationType:= ENUMERATED {
    notification        (0),
    privacyVerification (1),
    ... }
-- an unrecognized value shall be rejected by the receiver with a return error cause of
```

```
-- unexpected data value.

LocationNotificationRes ::= SEQUENCE {
    verificationResponse        [0] VerificationResponse OPTIONAL,
    ...}

VerificationResponse::= ENUMERATED {
    permissionDenied    (0),
    permissionGranted   (1),
    ... }

-- exception handling:
-- an unrecognized value shall be treated the same as value 0 (permissionDenied)

LCS-MOLRArg ::= SEQUENCE {
    molr-Type           [0] MOLR-Type,
    locationMethod      [1] LocationMethod          OPTIONAL,
    lcs-QoS             [2] LCS-QoS                 OPTIONAL,
    lcsClientExternalID [3] LCSClientExternalID     OPTIONAL,
    mlc-Number          [4] ISDN-AddressString      OPTIONAL,
    gpsAssistanceData   [5] GPSAssistanceData       OPTIONAL,
    ...}
-- The parameter locationMethod shall be included if and only if the molr-Type is set to value
-- deCipheringKeys or assistanceData.
-- The parameter gpsAssistanceData shall be included if and only if the molr-Type is set to value
-- assistanceData and LocationMethod is set to value assistedGPS.

MOLR-Type::= ENUMERATED {
    locationEstimate            (0),
    assistanceData              (1),
    deCipheringKeys             (2),
    ... }
-- exception handling:
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

LocationMethod::= ENUMERATED {
    msBasedEOTD         (0),
    msAssistedEOTD      (1),
    assistedGPS         (2),
    ... }
-- exception handling:
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

GPSAssistanceData::= OCTET STRING (SIZE (1..38))
-- Octets 1 to 38 are coded in the same way as the octets 3 to 7+2n of Requested GPS Data IE
-- in GSM 09.31.

LCS-MOLRRes::= SEQUENCE {
    locationEstimate        [0] Ext-GeographicalInformation     OPTIONAL,
    decipheringKeys         [1] DecipheringKeys                 OPTIONAL,
    ...}
-- Parameter locationEstimate shall be included if and only if the
-- molr-Type in LocationRequestArg was set to value locationEstimate.
-- Parameter decipheringKeys shall be included if and only if the molr-Type
-- in LocationRequestArg was set to value deCipheringKeys.
--


DecipheringKeys::= OCTET STRING (SIZE (15))
-- Octets in DecipheringKeys are coded in the same way as the octets 3 to 17 of Deciphering Key IE
-- in GSM 09.31. I.e. these octets contain Current Deciphering Key, Next Deciphering Key and
-- Ciphering Key Flag.



END
```