EUROPEAN TELECOMMUNICATION STANDARD

DRAFT

pr ETS 300 627

September 1997

Second Edition

GSM ®

GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS

Digital cellular telecommunications system (Phase 2);
Subscriber and equipment trace
(GSM 12.08 version 4.5.0)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Contents

Blank page

Foreword

This European Telecommunication Standard (ETS) has been produced by the Special Mobile Group (SMG) of the European Telecommunications Standards Institute (ETSI) and is now submitted for the One-step Approval Procedure (OAP) of the ETSI standards approval process.

This ETS specifies trace facilities for the Digital cellular telecommunications system.

The specification from which this ETS has been derived was originally based on CEPT documentation, hence the presentation of this ETS may not be entirely in accordance with the ETSI/PNE rules.

Proposed transposition dates	
Date of latest announcement of this ETS (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Introduction

The trace facility enables customer administration and network management to trace the activities of various entities when specific events occur within the PLMN. This facility should also enable the tracing of all the information that is available to the PLMN concerning the call path used by the associated entity. Examples of information that could be in a trace record are:

- the identity of the originating and terminating equipment of the mobile or fixed subscriber;

- the identity of the incoming and outgoing circuits of the nodes involved;

- supplementary Services invoked;

- all A-Interface messages.

The trace facility is a useful maintenance aid and development tool which can be used during system testing and proving. In particular it may be used in conjunction with test-MSs to ascertain the digital cell "footprint", the network integrity and also the network QOS as perceived by the PLMN customers.

The facility may be used by subscriber administration and network management for subscriber observation, e.g. following a customer complaint or on suspicion of equipment malfunction by the operator or at the request of the police.

As the amount of information that can be collected for a single call is very large, Network Elements can limit the number of simultaneous traces by either rejecting a trace request or by only producing a sub-set of the information required.

Blank page

1 Scope

This European Telecommunication Standard (ETS) specifies the Trace facility for GSM where it refers to:

- subscriber tracing (tracing of International Mobile Subscriber Identity (IMSI));

- equipment tracing (tracing of International Mobile station Equipment Identity (IMEI)).

It does not cover:

- types of trace which relate more to network elements than to individual subscribers e.g. tracing events within a Base Station System (BSS), and so on;

- tracing of all possible parties in e.g. a multi-party call.

It also refers only to tracing activated from the OSF and not to that activated by means of local Man Machine Interface (MMI).

2 Normative references

This ETS incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1]	GSM 01.04 (ETR 100): "Digital cellular telecommunications system (Phase 2); Abbreviations and acronyms".
[2]	GSM 04.08 (ETS 300 557): "Digital cellular telecommunications system (Phase 2); Mobile radio interface layer 3 specification".
[3]	GSM 08.06 (ETS 300 589): "Digital cellular telecommunications system (Phase 2); Signalling transport mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface".
[4]	GSM 08.08 (ETS 300 590): "Digital cellular telecommunications system (Phase 2); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".
[5]	GSM 08.58 (ETS 300 596): "Digital cellular telecommunications system (Phase 2); Base Station Controller - Base Transceiver Station (BSC - BTS) interface Layer 3 specification".
[6]	GSM 09.02 (ETS 300 599): "Digital cellular telecommunications system (Phase 2); Mobile Application Part (MAP) specification".
[7]	GSM 12.00 (ETS 300 612-1): "Digital cellular telecommunications system (Phase 2); Objectives and structure of Network Management (NM)".
[8]	GSM 12.01 (ETS 300 612-2): "Digital cellular telecommunications system (Phase 2); Common Aspects of GSM Network Management (NM)".
[9]	GSM 12.02 (ETS 300 613): "Digital cellular telecommunications system (Phase 2); Subscriber, Mobile Equipment (ME) and services data administration".
[10]	GSM 12.05 (ETS 300 616): "Digital cellular telecommunications system (Phase 2); Subscriber related event and call data".
[11]	GSM 12.20 (ETS 300 622): "Digital cellular telecommunications system (Phase 2); BSS Management Information".

[12]   CCITT Recommendation X.227 - ISO 8650: "Information technology - Open Systems Interconnection - Connection-oriented protocol for the association control service element: Protocol specification".

[13]   CCITT Recommendation X.721 (ITU-T | ISO/IEC 10165-1): "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[14]   CCITT Recommendation X.734 (ITU-T | ISO/IEC 10164-5): "Information technology - Open Systems Interconnection - Systems Management: Event report management function".

[15]   CCITT Recommendation X.735 (ITU-T | ISO/IEC 10164-6): "Information technology - Open Systems Interconnection - Systems Management: Log control function".

[16]   CCITT Recommendation X.731 (ITU-T | ISO/IEC 10164-2): "Information technology - Open Systems Interconnection - Systems Management: State management function".

3   Definitions and abbreviations

3.1   Definitions

For the purposes of this ETS, the following definitions apply:

activation of a trace: An action taken at the OSF through MMI commands to allow a trace record to be produced for a particular IMSI or IMEI when an Invocation Event occurs. This equates to "activation of a trace" in GSM 09.02 [6].

active pending: The state of an activated trace is called Active Pending in a particular NE when the subscriber or equipment being traced is not registered in that NE.

invocation of a trace: An event relating to a particular IMSI or IMEI that occurs in the network that causes data to be collected in a trace record in circumstances where trace has been activated for that IMSI or IMEI. This equates to "tracing subscriber activity" in GSM 09.02 [6] and "Trace Invocation" in GSM 08.08 [4]. It is possible that an event relating to the IMSI/IMEI may still be active when another event or events relating to the same IMSI/IMEI occurs which requires additional information to be collected. These additional events are termed parallel events. This additional trace information for parallel events is collected in the same trace record as the first event.

trace record: In the NEF a trace record is a set of traceable data collected as determined by the trace type. The trace record is collected under the trace record criteria specified by the OSF and transferred to the OSF.

3.2   Abbreviations

For all abbreviations used in this ETS, refer to GSM 01.04 [1].

4 Trace overview

Figure 1 gives an outline of the subscriber and equipment tracing and shows the relationship between the inputs on activation and deactivation and the trace record outputs.
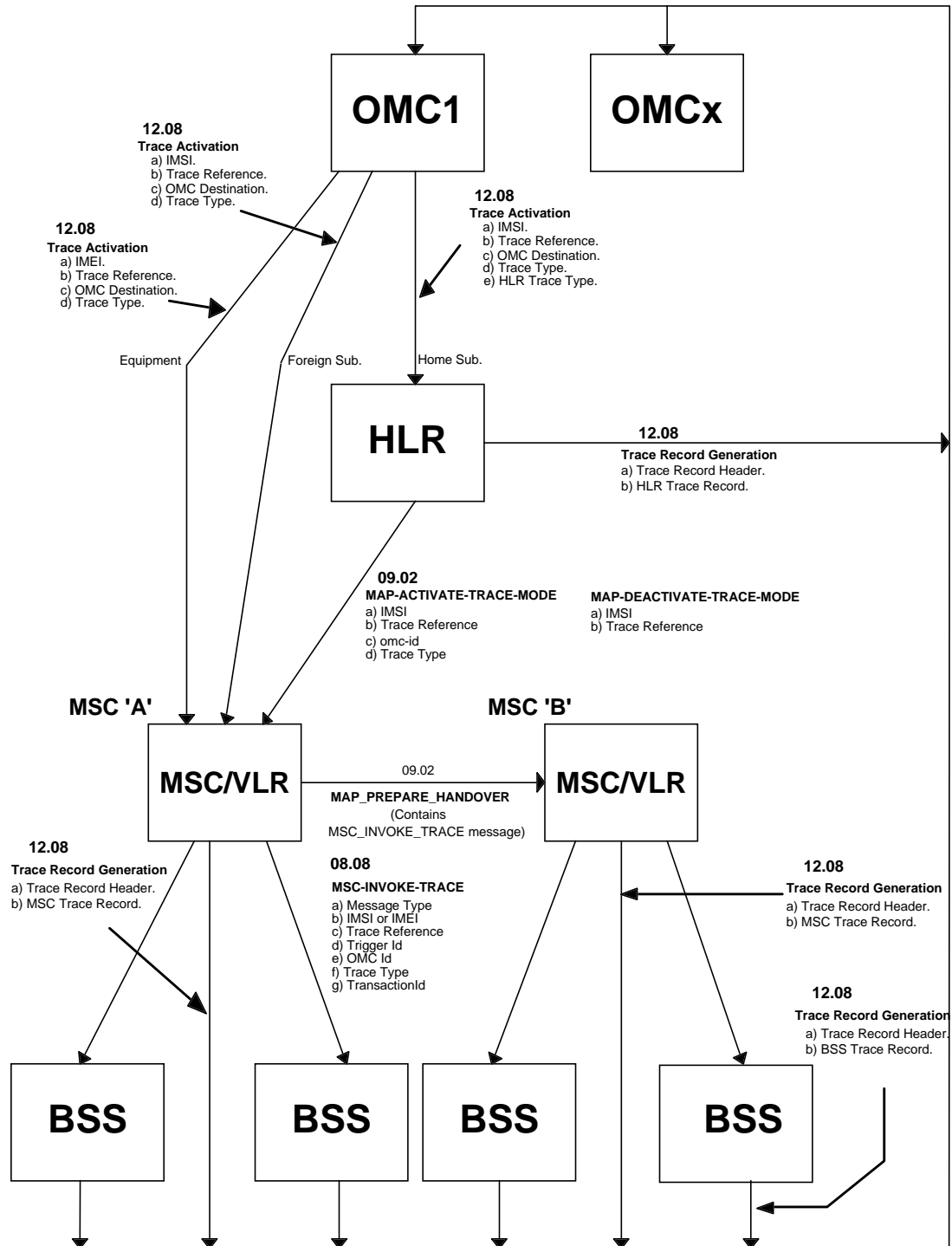
Figure 1: Subscriber and Equipment Trace for 12.08

Trace Activation and Deactivation are described in clause 5.

The Trace Types are defined in clause 6.

The Trace Records are defined in clause 7.

The following events may invoke a MSC or BSS trace:

- Call set-up within MSC (MOC, MTC) (incl. attempts);
- SS-Action;
- Location Update (Normal and Periodic);
- SMS-MO;
- SMS-MT;
- IMSI attach and detach.

Additionally, the following event may invoke a BSS trace:

- Handover.

An HLR Trace may be invoked by one of the following:

- Location updates/cancellations;
- Insert/delete subscriber data;
- Routing enquiry (speech and SM);
- Provide roaming number;
- SS activity;
- SMS: Alert service centre/Ready for SM;

Trace records are generated within the managed elements by the trace control function according to the trace type. Once a trace has been invoked and a trace record is being compiled, subsequent invoking events relating to that IMSI (parallel events) will not cause new records to be compiled simultaneously but will be contained in the same trace record as the first event.

For operator defined trace types the events on which trace records are generated and their contents are defined within the trace record generation control.

These records are then transferred to the OSF (as defined by OMC-Id of the Destination OMC or forwarded by the EFD) either as notifications (CMISE), or with bulk transfer (FTAM).

5 Trace activation and deactivation

5.1 General

This document is only concerned with the activation of a trace from an OSF (OMC), and the OSF shall keep a log of all trace activations and their deactivations. All entries in the log shall be date and time stamped.

In the case of an OSF (OMC) failure, it may be possible to activate and deactivate the trace at a particular network element by means of local MMI, but the procedures for doing this are not covered by this ETS.

Facilities shall exist to allow unsolicited trace data to be received by an OSF. This permits the collection of trace data if the triggering entity (i.e. OSF or network element) is different to the collecting OSF.

5.2 Subscriber Tracing (Tracing of IMSI)

5.2.1 General

The tracing of both home and foreign roaming subscribers can be handled with this function.

If implemented, then the way the trace facility is used and organized, including restrictions due to national laws and regulations, will be a matter for the PLMN Operator.

All trace records created in the HLR, MSC "A", MSC "B" and BSS are forwarded to the OSF either as notifications and/or with bulk transfer, as defined in the trace parameters.

The following scenarios are identified from the HPLMN operation viewpoint:

a) HPLMN Operator traces its own (home) IMSI within the HPLMN;

b) HPLMN Operator traces the HLR activities of its own (home) IMSI while they are roaming in a VPLMN;

c) HPLMN Operator wishes to trace foreign roaming subscribers (IMSI) within its own HPLMN.

5.2.2 HPLMN Operator Traces Home Subscriber within the HPLMN

The Operator may activate a trace for a home subscriber (IMSI) from any OSF by invoking the management function **Activate Home Subscriber Trace** in the HLR where the IMSI is contained. This request includes the trace parameters in the following list:

a) IMSI to be traced;
b) Trace Reference;
c) OMC-Id of the destination OMC;
d) Trace Type;
e) HLR Trace Type.

For each IMSI, only one HPLMN subscriber trace can be active, subsequent requests being rejected.

If the IMSI is roaming within its HPLMN, then the trace request is forwarded to the VLR where the subscriber is registered via a MAP message (MAP-ACTIVATE-TRACE-MODE).

When the HPLMN subscriber trace is activated, a trace record will be created by MSC "A" , MSC "B", HLR or BSS when certain invoking events occur i.e. MOC, MTC, SS-Action, SMS-MO, SMS-MT, Location Update, IMSI attach and detach. The trace action and record layout is defined by the trace type parameters.

A trace may be invoked in the BSS when an Invoking Event, specified in the Invoking Event sub-field in the Trace Type, occurs and the BSS Record Type is set to a value other than "No BSS Trace". A Trace is invoked by sending a BSSMAP MSC_INVOKE_TRACE message from the MSC to the BSS. When the BSS receives this message it starts tracing the necessary fields as specified in the BSS Record associated with the specified BSS Record Type.

If the subscriber is roaming in a foreign PLMN then the HPLMN subscriber trace request is stored in the HLR, but the trace is not active in the HPLMN VLRs.

The trace is deactivated by using the management function **Deactivate Home Subscriber Trace** in the HLR. This request includes the trace parameters in the following list:

a)   IMSI;
b)   Trace Reference.

If the IMSI is roaming within its HPLMN then the trace deactivation request is forwarded to the VLR where the subscriber is registered via a MAP message (MAP-DEACTIVATE-TRACE-MODE).

TMN Management Functions required for trace activation (in HLR):

Activate Home Subscriber Trace

Deactivate Home Subscriber Trace

5.2.3   HPLMN Operator traces the HLR activities of own IMSI roaming in a VPLMN

This scenario is identical to the previous scenario with the exception that the only records generated come from the HLR.

5.2.4   PLMN Operator wishes to trace foreign subscribers (IMSI) in own PLMN

In order to trace the IMSIs of roaming subscribers in own PLMN, a list of those IMSIs plus the associated subscriber trace parameters must be stored in the VLR. No HLR trace records are produced for foreign subscriber traces.

The operator may activate a trace for any foreign roaming IMSI from an OSF by invoking the management function **Activate Foreign Subscriber Trace** in one or more VLRs within their own PLMN. If the location of the subscriber is not known it is necessary to activate the trace in all VLRs where the subscriber may be located.

The following trace parameters are sent with this request:

a)   IMSI to be traced;
b)   Trace Reference;
c)   OMC-Id of the destination OMC;
d)   Trace Type.

The trace request is stored in the VLR. If the subscriber subsequently roams into the VLR area the VPLMN subscriber trace will be activated.

For each IMSI only one foreign subscriber trace can be active in a particular VLR, subsequent requests being rejected.

A trace may be invoked in the BSS when an Invoking Event, specified in the Invoking Event sub-field in the Trace Type, occurs and the BSS Record Type is set to a value other than "No BSS Trace". A Trace is invoked by sending a BSSMAP MSC_INVOKE_TRACE message from the MSC to the BSS. When the BSS receives this message it starts tracing the necessary fields as specified in the BSS Record associated with the specified BSS Record Type.

The VPLMN subscriber trace is deactivated by invoking **Deactivate Foreign Subscriber Trace** in the VLR. This request includes the trace parameters in the following list:

a)   IMSI;
b)   Trace Reference.

TMN Management Functions required for trace activation (in VLR):

Activate Foreign Subscriber Trace

Deactivate Foreign Subscriber Trace

5.3   Equipment Tracing (Tracing of IMEI)

5.3.1   General

If the tracing of IMEIs is implemented then the way the trace facility is used and organized, including restrictions due to national laws and regulations, will be a matter for the PLMN Operator.

An IMEI may be traced in order to find out the current IMSI, or the location or behaviour of faulty or stolen equipment reported via the EIR.

The ETS describes one method of handling IMEI tracing i.e. tracing of IMEI via the VLR.

5.3.2   Tracing of IMEI via VLR

The operator may activate an equipment trace for any subscriber's equipment (IMEI) from an OSF by invoking the management function **Activate Equipment Trace** in one or more VLR in the HPLMN. The trace must be activated in all VLRs controlling areas where it is required to trace the target IMEI. The trace parameters are transmitted with the activation request.

The following trace parameters are sent with this request:

a)   IMEI to be traced;
b)   Trace reference;
c)   OMC-Id of the destination OMC;
d)   Trace Type.

For GSM Phase 2 Mobile Stations the IMEI will be available to the Network as it can be included in the BSS-MAP message CIPHER-MODE-COMPLETE. If IMEI trace is required, it is the responsibility of the network operator to specify that CIPHER-MODE-COMPLETE contains IMEIs, or optionally the IMEI is called for in connection with MOC, location update etc. Alternatively the network can ask the MS for the IMEI by sending a GSM 04.08 [2] IDENTITY REQUEST message to the MS, indicating that the IMEI is required.

When a subscriber arrives at a VLR using an equipment with an IMEI for which trace has been activated (but is in pending state) at that VLR then the IMEI trace will become.

For each IMEI only one equipment trace can be active in a particular VLR at any one time, subsequent requests being rejected, although both the IMSI trace (home subscriber tracing and foreign subscriber tracing) and the IMEI trace can be active at the same time.

This equipment trace is deactivated by invoking the management function **Deactivate Equipment Trace** in the VLR. This request includes the trace parameters in the following list:

a)   IMEI;
b)   Trace Reference.

TMN Management Functions required for trace activation (in VLR):

Activate Equipment Trace

Deactivate Equipment Trace

5.4 TMN Management Functions for Activation/Deactivation

5.4.1 List of Functions

5.4.1.1 HLR

Activate Home Subscriber Trace

Deactivate Home Subscriber Trace

5.4.1.2 MSC/VLR

Activate Foreign Subscriber Trace

Deactivate Foreign Subscriber Trace

Activate Equipment Trace

Deactivate Equipment Trace

5.4.2 Activate Home Subscriber Trace

This function is equivalent to the OM_Subscriber_Tracing_Activation_req in GSM 09.02 [6].

The subscriber tracing procedures are used for the management of the trace status and the type of trace.

The subscriber tracing activation procedure operates as follows:

a)   The OSF creates a *tracedHomeSubscriberInHlr* object instance in the HLR of the subscriber to be traced.

b)   If the subscriber is roaming outside of the HPLMN or not currently registered, then the trace is in active pending state. The home subscriber trace for the subscriber is activated in the HLR on a subsequent location update. This activation is shown as an attribute value change in the attribute traceActivatedInVlr.

c)   If the subscriber is already registered then the home subscriber trace becomes immediately active in the HLR (after positive confirmation from the VLR).

When the trace is first activated then the status of the trace indicator attribute traceActivatedInVlr in the *tracedHomeSubscriberInHlr* object instance is set to **False**.

If the subscriber is **registered and is roaming in the home PLMN area** then the HLR will initiate the MAP-ACTIVATE-TRACE-MODE request primitive and the trace indicator status will be set to **True** only in the case of a positive confirmation of the MAP-ACTIVATE-TRACE-MODE. In case of an error, the trace indicator status remains **False**.

If the MAP-ACTIVATE-TRACE-MODE confirm primitive is received indicating an error situation then this is recorded in an error attribute in the *tracedHomeSubscriberInHlr* object instance.

If the subscriber roams to an area outside that where tracing is possible then the status in the *tracedHomeSubscriberInHlr* object instance is updated to **False**.

The trace records are sent from the recording NEF to the OSF by the deployed event reporting mechanism (see chapter Trace Record Transfer). The Trace Type attribute indicates the type of trace records to be produced and the way in which they will be reported i.e. each event record being either directly sent to the OSF in real-time, or being collected in a file for later transfer.

All attribute value changes will be reported with a notification to the OSF.

System management functions:

> Create tracedHomeSubscriberInHlr
> Get Attribute

Notifications:

> objectCreation
> attributeValueChange

5.4.3   Deactivate Home Subscriber Trace

This function is equivalent to the OM_Subscriber_Tracing_Deactivation_req in GSM 09.02 [6].

The subscriber trace is deactivated by the OSF deleting the *tracedHomeSubscriberInHlr* object instance in the HLR.

If the trace status is **True** then the HLR will send the MAP-DEACTIVATE-TRACE-MODE message to VLR.

If the MAP-DEACTIVATE-TRACE-MODE confirm primitive is received indicating an error situation then this is indicated to the OSF via an error attribute in the *tracedHomeSubscriberInHlr* object instance and the object is not deleted.

The home subscriber trace deactivation can be indicated with a notification to the initiating OSF.

System management functions:

> Delete tracedHomeSubscriberInHlr

> Get Attribute

Notifications:

> objectDeletion

> attributeValueChange

5.4.4   Activate Foreign Subscriber Trace

This function is analogous to the OM_Subscriber_Tracing_Activation_req in GSM 09.02 [6], but the trace activation is performed directly in the VLR.

The foreign subscriber trace is activated by the OSF executing the system management function Create *tracedForeignSubscriberInVlr* in the VLR.

THE OSF creates a *tracedForeignSubscriberInVlr* object instance in the VLR(s) in which the network operator wishes to trace the subscriber.

The tracing continues as follows:

a)   If the subscriber is not currently registered, then the foreign subscriber trace for the subscriber is active pending. It is activated (i.e. status attribute value is set to **True**) in the VLR on a subsequent location update. The activation is notified to the OSF as an attribute value change in the attribute foreignSubscriberRegisteredInVlr.

b)   If the subscriber is already registered then the foreign subscriber trace becomes immediately active in the VLR.

When the trace is first activated then the status of the attribute foreignSubscriberRegisteredInVlr is set to **False**. When the traced subscriber registers in the VLR the attribute status of foreignSubscriberRegisteredInVlr is set to **True**.

All attribute value changes will be reported with a notification to the OSF.

The trace records are sent from the corresponding MSC to the OSF by the deployed event reporting mechanism (see chapter Trace Record Transfer). The Trace Type attribute indicates the type of trace records to be produced and the method by which they will be reported.

System management functions:

>   Create tracedForeignSubscriberInVlr

>   Get Attribute

Notifications:

>   objectCreation

>   attributeValueChange

5.4.5   Deactivate Foreign Subscriber Trace

This function is analogous to the OM_Subscriber_Tracing_Deactivation_req in GSM 09.02 [6], but the trace deactivation is performed.

The OSF deactivates subscriber trace by deleting the *tracedForeignSubscriberInVlr* object instance in the VLR(s) in which the object instance had previously been created.

The foreign subscriber trace is deactivated by the OSF executing the system management function Delete *tracedForeignSubscriberInVlr* in the VLR.

System management functions required:

>   Delete tracedForeignSubscriberInVlr

Notifications required:

>   objectDeletion

>   attributeValueChange

5.4.6   Activate Equipment Trace

This function is analogous to the OM_Subscriber_Tracing_Activation_req in GSM 09.02 [6], but the trace activation is performed directly in the VLR.

The equipment trace is activated by the OSF executing the system management function Create *tracedEquipmentInVlr*.

The OSF creates a *traceEquipmentInVlr* object instance in the VLR(s) for the areas to be monitored.

The tracing continues as follows:

a)   If the equipment is not currently registered, then the equipment trace for the equipment is active pending. It is activated (i.e. status attribute value is set to **True**) in the VLR on a subsequent location update or IMSI attach. The activation is notified to the OSF as an attribute value change in the attribute equipmentRegisteredInVlr.

b)   If the equipment is already registered then the equipment trace becomes immediately active in the VLR.

When the trace is first activated then the status of the attribute equipmentRegisteredInVlr is set to **False**. When the equipment registers in the VLR the attribute status of equipmentRegisteredInVlr is set to **True**.

All attribute value changes will be reported with a notification to the OSF.

The trace records are sent from the corresponding MSC to the OSF by the deployed event reporting mechanism (see chapter Trace Record Transfer). The Trace Type attribute indicates the type of trace records to be produced and the method by which they will be reported.

System management functions:

Create tracedForeignSubscriberInVlr

Get Attribute

Notifications:

objectCreation

attributeValueChange

5.4.7   Deactivate Equipment Trace

This function is analogous to the OM_Subscriber_Tracing_Deactivation_req in GSM 09.02 [6], but the trace deactivation is performed in the VLR.

The equipment trace is deactivated by the OSF executing the system management function Delete *tracedEquipmentInVlr*.

The OSF deactivates equipment trace by deleting the *tracedEquipmentInVlr* object instance in the VLR(s) in which the object instance had previously been created.

System management functions:

Delete tracedEquipmentInVlr

Notifications:

objectDeletion

attributeValueChange

5.5 HLR Functional Entities

Figure 2 shows that part of the Subscriber Administration Containment Tree for the HLR relevant to Trace activation and deactivation.

```
                        ┌──────────────┐
                        │  hlrFunction │
                        └──────┬───────┘
                               │
                               │
                 ┌─────────────┴──────────────┐
                 │ tracedHomeSubscriberInHlr  │
                 └────────────────────────────┘
```
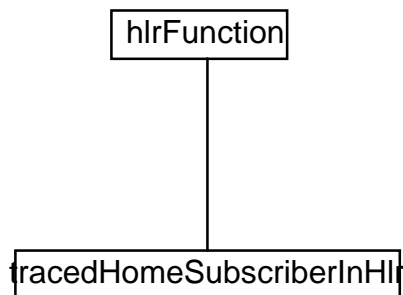
Figure 2: Subscriber Trace Containment Tree for the HLR

5.5.1 Managed Object Classes in HLR

5.5.1.1 tracedHomeSubscriberInHlr

This object class controls the home subscriber trace facility. Each instance of this object represents an IMSI of a home subscriber to be traced i.e. if an instance for an IMSI exists then that means that the trace has been activated for that IMSI.

Name	M/O	Value-Set
IMSI	RDN	Single
traceActivatedInVlr	M	Single
traceReference	M	Single
traceType	M	Single
hlrTraceType	M	Single
operationSystemId	O	Single
mapErrorOnTrace	M	Single

5.5.1.2 Attributes

5.5.1.2.1 tracedHomeSubscriberInHlr

IMSI

This attribute is the RDN of the object *tracedHomeSubscriberInHlr* and defines an IMSI to be traced. It will be an IMSI of a home subscriber for whom tracing is required.

The syntax is defined in MAP-CommonDataTypes IMSI.

traceActivatedInVlr

This attribute is single valued and gives an indication of the status of the Trace. Possible values of this attribute are **True** and **False**.

On creation this attribute is set to **False**.

If the subscriber is registered and roaming within the HPLMN (see GSM 09.02 [6]) then the attribute is set to **TRUE** (in case of positive confirmation from VLR).

If the subscriber roams to an area which is outside that where tracing is possible the attribute is set to **FALSE**.

Each status change triggers an attributeValueChange notification.

traceReference

This attribute is a unique reference for a particular trace associated with a particular IMSI and is allocated by the OSF.

traceType

This attribute describes the invoking events for which the operator wishes to collect a trace record for a particular IMSI in an MSC or BSS. It also describes the type of record to be collected and indicates whether or not this is a priority trace.

hlrTraceType

This attribute describes the type of trace record (if any) the operator wishes to be collected in the HLR for a particular IMSI. It is assumed for all invoking events.

operationSystemId

This attribute contains the address of the OSF to which the operator wishes the trace records associated with this particular IMSI to be sent.

If EFDs are used then trace records are sent to OSFs defined in EFD.

mapErrorOnTrace

This attribute is single valued and read only. The syntax is defined in GSM-12-02-Syntax MapErrorOnTrace.

It is set by MAP and contains the MAP-Errors that may be returned in the confirm primitives of the ActivateTraceMode and DeactivateTraceMode Operations.

If there are MAP-Errors in case of activation of trace, the traceActivatedInVlr parameter is set to **False**.

If there are Map-Errors in case of deactivation of trace (deleting tracedHomeSubscriberInHlr), the deleting is not completed successfully.

Possible error values are defined in MAP-OperationAndMaintenance Operations and in MAP-Errors.

5.5.1.3   Notifications

For each object:

objectCreation

objectDeletion

AttributeValueChange

5.6 VLR Functional Entities

Figure 3 shows that part of the Subscriber Administration Containment Tree for the VLR relevant to Trace.

```
                    ┌─────────────────┐
                    │   vlrFunction   │
                    └─────────────────┘
                             │
            ┌────────────────┴────────────────┐
┌───────────────────────────┐   ┌──────────────────────┐
│ tracedForeignSubscriberInVlr │ │  tracedEquipmentInVlr │
└───────────────────────────┘   └──────────────────────┘
```
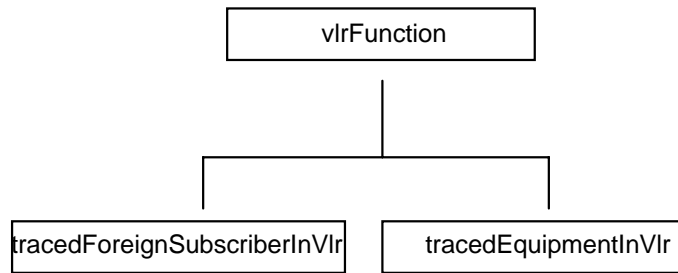
Figure 3: Subscriber Trace Containment Tree for the VLR

5.6.1 Managed Object Classes in VLR

5.6.1.1 tracedForeignSubscriberInVlr

This object class controls the foreign subscriber trace facility. Each instance of this object represents an IMSI of a foreign subscriber to be traced i.e. if an instance for an IMSI exists then that means that the trace has been activated for that IMSI.

Name	M/O	Value-Set
IMSI	RDN	Single
foreignSubscriberRegisteredInVlr	M	Single
traceReference	M	Single
traceType	M	Single
operationSystemId	O	Single

5.6.1.2 tracedEquipmentInVlr

This object class controls the equipment trace facility. Each instance of this object represents an IMEI to be traced i.e. if an instance for an IMEI exists then that means that the trace has been activated for that IMEI.

Name	M/O	Value-Set
IMEI	RDN	Single
equipmentRegisteredInVlr	M	Single
traceReference	M	Single
traceType	M	Single
operationSystemId	O	Single

5.6.1.3 Attributes

5.6.1.3.1 tracedForeignSubscriberInVlr

IMSI

This attribute is the RDN of the object *tracedForeignSubscriberInVlr* and defines an IMSI to be traced. It will be an IMSI of a foreign subscriber for whom tracing is required.

The syntax is defined in MAP-CommonDataTypes IMSI.

foreignSubscriberRegisteredInVlr

This attribute is single valued and gives an indication of the status of the Trace. Possible values of this attribute are **True** and **False**.

On creation this attribute is set to **False**.

If the foreign subscriber is currently registered in the VLR then the attribute is set to **TRUE**.

If the foreign subscriber is not registered in the VLR then the attribute is set to **FALSE**.

Each status change triggers an attributeValueChange notification.

traceReference

This attribute is a unique reference for a particular trace associated with a particular IMSI and is allocated by the OSF.

traceType

This attribute describes the invoking events that the operator wishes to collect a trace record for a particular IMSI in an MSC or BSS. It also describes the type of record to be collected and indicates whether or not this is a priority trace.

operationSystemId

This attribute contains the address of the OSF to which the operator wishes the trace records associated with this particular IMSI to be sent.

If EFDs are used, then trace records are sent to OSFs defined in EFD.

5.6.1.3.2 tracedEquipmentInVlr

IMEI

This attribute is the RDN of the object *tracedEquipmentInVlr* and defines an IMEI to be traced. It will be an IMEI for the equipment for which tracing is required.

The syntax is defined in MAP-CommonDataTypes IMEI.

equipmentRegisteredInVlr

This attribute is single valued and gives an indication of the status of the Trace. Possible values of this attribute are **True** and **False**.

On creation this attribute is set to **False**.

If the equipment is registered in the VLR then the attribute is set to **TRUE**.

If the equipment is not registered in the VLR then the attribute is set to **FALSE**.

Each status change triggers an attributeValueChange notification.

traceReference

This attribute is a unique reference for a particular trace associated with a particular IMSI and is allocated by the OSF.

traceType

This attribute describes the invoking events for which the operator wishes to collect a trace record for a particular IMSI in an MSC or BSS. It also describes the type of record to be collected and indicates whether or not this is a priority trace.

operationSystemId

This attribute contains the address of the OSF to which the operator wishes the trace records associated with this particular IMSI to be sent.

If EFDs are used, then trace records are sent to OSFs defined in EFD.

5.6.1.4   Notifications

objectCreation

objectDeletion

attributeValueChange

6 Trace Types

6.1 MSC/BSS Trace Type

The Trace Type field contains the type of trace activated in the MSC or BSS. The trace type consists of the following components.

8	7	6	5	4	3	2	1
Priority Indication	For future expansion (Set to 0)	BSS Record Type		MSC Record Type		Invoking Event	

Table 1: Invoking Events

Bits		Invoking Events
2	**1**	
0	0	MOC, MTC, SMS MO, SMS MT, SS, Location Updates, IMSI attach, IMSI detach
0	1	MOC, MTC, SMS_MO, SMS_MT, SS only
1	0	Location updates, IMSI attach IMSI detach only
1	1	Operator definable

If the "operator definable" option is selected, all subsequent Trace Record Types are deemed to be "operator definable". In this case the significance of bits 3-6 are operator defined, however the significance of bit 8 remains "Priority Indication". In all cases, for GSM Phase 2 Network Elements the setting of the 7 shall not affect trace record generation.

Table 2: MSC Record Type

Bits		Record Type
4	**3**	
0	0	Basic
0	1	Detailed (Optional)
1	0	Spare
1	1	No MSC Trace

Table 3: BSS Record Type

Bits		Record Type
6	**5**	
0	0	Basic
0	1	Handover
1	0	Radio
1	1	No BSS Trace

Table 4: Priority Indication

Bit 8	Priority
0	No priority
1	Priority

This bitmap of the Trace Type is required to map onto the Trace Type as defined in GSM 09.02 as an Integer with 256 possible values. This is achieved by a binary to decimal conversion of the bitmap, where bit 8 has weight 128 and bit 1 has weight 1.

6.2 HLR Trace Type

The HLR Trace Type field contains the type of trace activated in the HLR. The trace type consists of the following components.

8	7	6	5	4	3	2	1
Priority Indication	For future expansion (Set to 0)			HLR Record Type		Invoking Event	

Table 5: Invoking Events

Bits		Invoking Events
2	**1**	
0	0	All HLR Interactions
0	1	Spare
1	0	Spare
1	1	Operator definable

If the "operator definable" option is selected, all subsequent Trace Record Types are deemed to be "operator definable". In this case the significance of bits 3 and 4 are operator defined, however the

significance of bit 8 remains "Priority Indication". In all cases, for GSM Phase 2 Network Elements the setting of bits 5-7 shall not affect trace record generation.

Table 6: HLR Record Type

Bits		Record Type
4	**3**	
0	0	Basic
0	1	Detailed
1	0	Spare
1	1	No HLR Trace

Table 7: Priority Indication

Bit 8	Priority
0	No priority
1	Priority

This bitmap of the Trace Type is only required in the HLR and is not required to be mapped onto any GSM 09.02 [6] or other Trace Types.

7 Trace record contents

7.1 General

Tables 9, 10 and 11 illustrate the structure of a trace record, and table 8 illustrates the structure of the Trace Record header. This header is used at the start of all trace records.

In the case where trace data is distributed over several records, linkage between the records is provided in the record header. If parallel events are also being traced, additional linkage for the traced data relating to each event is provided in the trace record content. Parallel events are not applicable to BSS trace records.

The trace reference, trace type and operation system identification are all provided on trace activation. Each record may contain an MSC, BSS or HLR event record. A key is included in the table indicating whether or not the field is mandatory. In this table and throughout this document the key field has the following meaning:

M	This field must appear in at least one trace record associated with the invoking event. Any exceptions to this rule are explicitly described.
C	This field is only available under certain conditions. If available this field must be present in at least one trace record associated with the invoking event. The conditions under which this field is available are individually described.
O	This field is optional and its support is a matter for agreement between equipment manufacturer and network operator. Equipment manufacturers do not have to be capable of providing all these fields to claim conformance with this ETS.
X	This field is not required in this instance.

Table 8: Trace Record Header

Field		Description
IMSI or IMEI	M	IMSI or IMEI of subscriber/equipment being traced. See GSM 12.05 annex B definitions for Served IMSI and Served IMEI. The BSS shall include this field in the reace record header only if available in the A-interface MSC INVOKE TRACE message.
Trace Reference	M	An identifier assigned by the OSF at Trace Activation which may be used by the OSF in conjunction with the IMSI/IMEI and the Transaction ID to uniquely identify a record or collection of records for one particular trace. This must always appear in every trace record.
Transaction id	C	An identifier of a particular transaction, described in GSM 08.08. It shall be included if available in the A-Interface message **MSC_INVOKE_TRACE**.
Omc-Id	O	The address of the OS entity that the OSF activating the trace requires priority trace records to be sent to by the NE performing the trace (see also clause 9 Trace Record Transfer).
MSC/BSS Trace Type	C	This field contains the MSC/BSS trace type as provided in the trace activation message (see subclause 6.1 MSC/BSS Trace Type). It must always appear in the first record header.
HLR Trace Type	C	This field contains the HLR trace type as provided in the trace activation message (see subclause 6.2 HLR Trace Type). It must always appear in the first record header.
MSC/BSS Trace Type Used	O	This field contains the MCS/BSS trace type which has been applied. This trace type may be different to the one provided in the trace activation message due to manufacturer constraints. It must always appear in the first record header.
HLR Trace Type Used	O	This field contains the HLR trace type which has been applied. This trace type may be different to the one provided in the trace activation message due to manufacturer constraints. It must always appear in the first record header.
Start Time	M	The time the compilation of the Trace Record was started. It must always appear in the first record header. All timestamps used in the TraceEvent Record are relative to this time.
End Time	M	The time the compilation of the Trace Record was completed. It must always appear in the last record header. It may be used by the OSF as an indication that the trace in that particular Network Element is completed.
Recording Entity	M	For MSC/HLR - the E.164 number of the recording entity. For BSS - the BSC_ID as given in GSM 12.20 [11]. Alternatively the recording entity may be expressed as a graphic string.
Trace Event Record	M	This field contains either an MSC, HLR or BSS trace record as described in subclauses 7.2 to 7.4 below. This must always appear in every trace record.
Sequence Number	C	This field is used to identify the sequence of records from a particular recording entity when more than one trace record is produced for the invoking event.
Reason For Record	C	This specifies why the record was generated by the NE (see subclause 8.2). In addition to these reasons, other manufacturer specific reasons may be specified (see subclause 8.2.3).

7.2 MSC Trace Record Content

The following types of fields are supported in the 2 MSC trace types.

Table 9: MSC Trace Record Content

Field	MSC Trace Type		Description
	Basic	**Detailed**	
Invoking Event	M	M	Event invoking trace (Not available at the non-anchor MSC on Inter-MSC Handover).
Served IMSI	C	C	IMSI of the calling party in the case of MOC or the called party in the event of MTC. Not available in case of emergency call without SIM. This field is only required for IMEI trace.
Served IMEI	C	C	IMEI of the calling ME in the case of MOC or the called party in the event of MTC. This field is only required for IMSI trace.
Served MSISDN	C	C	Primary MSISDN of the party being traced.
Calling/Called Number	C	C	The MSISDN of the calling party in case of MTC. The MSISDN of the called party in case of MOC.
Calling Subaddress	C	C	The subaddress of the calling party (for both MOC and MTC).
Called Subaddress	C	C	The subaddress of the called party (for both MOC and MTC).
Translated Number	C	C	The called number of the party not being traced after digit translation within the MSC (if applicable) (i.e. applies to MOC only).
Connected Number	C	C	The number of the party not being traced (applies to MOC only).
Forwarded-to Number	C	C	The number to which the call will be forwarded (applies to MTC only).
Forwarded-to Subaddress	C	C	The subaddress to which the call will be forwarded (applies to MTC only).
Redirecting Number	C	C	The number from which the call was last redirected (applies to MTC only).
Original Called Number	C	C	The number of the original called party (applies to MTC only).
Roaming Number	C	C	The MSRN of the traced subscriber in the case of MTC, or the MSRN of the called subscriber in case of MOC, if available.
Network Trunk Group Point	C	C	In case of a MOC the outgoing trunk on which the call leaves the MSC. In case of an MTC the incoming trunk on which the call originates as seen from the MSC.
Basic Service	C	C	The bearer- or teleservice employed.
Radio Channel types	O	C	A list of radio channel types used during the compilation of the trace record, each timestamped.
BSS Handover Trunk	O	C	A list of the incoming/outgoing trunk group and member used to connect the MSC to BSS (including the original and each intra-MSC BSS handover) each time-stamped.
MSC Handover Trunk	O	C	A list of the trunk group and member used to connect two MSCs (including the original and each inter-MSC handover) each time-stamped.

(continued)

Table 9 (concluded): MSC Trace Record Content

Field	MSC Trace Type Basic	MSC Trace Type Detailed	Description
Location	C	C	A list of Location Area Codes / Cell Ids used during the compilation of the trace record starting with the identity of the cell in which the invoking event originated or terminated, each time stamped.
SS Information	C	C	A list of information related to any SS actions carried out during the period of the trace. The SS Information contains the SS Code for each SS Action, the Basic Services for which each SS action was carried out, the type of each SS action carried out, a list of SS parameters associated with each SS action, the result of each SS action and the Invoke Id allocated for each SS Action.
AOC Parameters	O	C	A list of the charge advice parameters sent to the MS (including on call set-up and on changes as a result of a tariff switch over), each timestamped.
MS Classmark 2	C	C	A list of the mobile station classmark 2 information (starting with on call set-up), each timestamped.
Call Termination Diagnostics	C	C	A detailed reason for the release of the connection. See GSM 12.05 annex B - Diagnostics.
A-Interface Messages	X	C	A sequential list of all DTAP and BSSMAP messages passed on the A-Interface.
C-Interface Messages	X	C	A sequential list of all MAP messages passed between the Tracing MSC and the HLR/AUC.
D-Interface Messages	X	C	A sequential list of all MAP messages passed between the Tracing VLR and the HLR/AUC.
E-Interface Messages	X	C	A sequential list of all MAP messages passed between the Tracing MSC and the subsequent MSC.
F-Interface Messages	X	C	A sequential list of all MAP messages passed between the Tracing MSC and the EIR.
G-Interface Messages	X	C	A sequential list of all MAP messages passed between the Tracing VLR and another VLR.
Network Signalling Messages	X	C	A sequential list of all user part messages e.g. ISUP, TUP messages.
Event Start Time	C	C	The time the event was started. It must always appear in case the trace record is already being compiled and the event belonging to this event record for this same subscriber occurs.
Event Stop Time	C	C	The time the event was finished. It must always appear in case the trace record is still being compiled due to an ongoing event and the event belonging to this event record finishes.
Event Number	M	M	The Event Number is used to identify tracing data belonging to the same event.
Record extensions	O	O	A set of network/ manufacturer specific extensions to the record.

7.3 BSS Trace Record Content

The following types of fields are supported in the 3 BSS trace record types:

Table 10: BSS Trace Record Content

Field	BSS Trace Type			Description
	Basic	Handover	Radio	
Invocation Message	M	M	M	GSM 08.08 [4] invocation message which started the trace action.
BTS ID	M	M	M	The ids of all BTSs accessed by the traced party during the period of the trace invocation (as per GSM 12.20 [11]), each timestamped.
TRX ID	M	M	M	The ids of all TRXs accessed by the traced party during the period of the trace invocation (as per GSM 12.20 [11]), each timestamped.
TRAU ID	O	O	O	The ids of all TRAUs accessed by the traced party during the period of the trace invocation (as per GSM 12.20 [11]), each timestamped.
Radio Channel Info.	M	M	M	The radio channel types and descriptions used during the period of the trace invocation, each timestamped.
Request type	C	C	C	The reasons for channel seizure (originating, terminating, re-establishment, handover) (see GSM 04.08 [2]), each timestamped.
End Indication	C	C	C	The reasons for channel release (see GSM 04.08 [2]), each timestamped.
MS Power	X	C	C	The last MS power used before a channel is released (see GSM 12.20 [11]), each timestamped.
BS Power	X	C	C	The last BS power used before a channel is released (see GSM 12.20 [11]), each timestamped.
Timing advance	X	C	C	The last timing advance used before a channel is released (see GSM 12.20 [11]), each timestamped.
MS Classmark 1	C	C	C	The MS Classmark 1 indicated during the period of the trace invocation, each timestamped.
MS Classmark 2	C	C	C	The MS Classmark 2 indicated during the period of the trace invocation, each timestamped.
MS Classmark 3	C	C	C	The MS Classmark 3 indicated during the period of the trace invocation, each timestamped.
BSIC	M	M	M	This field is the combination of Network Colour Code and Base station Colour Code (see GSM 12.20 [11]).
CIC	C	C	C	The terrestrial circuit identification codes used for the call on which the trace is being performed, each timestamped (see GSM 08.08 [4]).
Handover result	O	C	C	The results of each handover occurring during the period of the trace invocation each timestamped.
Handover cause	O	C	C	The reasons for starting each handover attempt during the period of the trace invocation (see GSM 08.08 [4]), each timestamped.

(continued)

Table 10 (concluded): BSS Trace Record Content

Field	BSS Trace Type			Description
	Basic	**Hando ver**	**Radio**	
Handover duration	O	C	C	The times taken between sending the handover command and receiving the handover complete for each successful handover, each timestamped.
Target Cell list	X	C	C	The target cells at the start of each handover attempt, each timestamped.
Synchronization information	X	C	C	The synchronization values for each handover attempt, each timestamped.
SCCP connection event	X	O	O	Each SCCP connection event used during the period of the trace invocation (Connection Request, Confirm, Refuse, Released, Released Complete), each timestamped.
BSSMAP message	X	C	C	L3 Message contents, during the period of the trace invocation, each timestamped, see GSM 08.08 [4].
DTAP message	X	O	O	L3 Message contents, during the period of the trace invocation each timestamped, see GSM 04.08 [2].
RR message	X	C	C	L3 Message contents, during the period of the trace invocation, each timestamped, see GSM 04.08 [2]. Only applies to those parts of the message between the BSC and the MS.
A-bis Messages	X	X	C	All Abis messages except measurement reports and power control, each timestamped, see GSM 08.58 [5].
Timed A-bis Messages	X	C	X	X Abis messages (except measurement reports and power control) received before and Y Abis messages received after a handover, each timestamped. X & Y are operator configurable parameters via MMI and are local to the BSS.
Measurement Reports	X	X	C	All uplink and downlink measurement reports, each timestamped, see GSM 08.58 [5].
Timed Measurement Reports	X	C	X	X uplink and downlink measurement reports received before and Y measurement reports received after a handover, each timestamped. X & Y are operator configurable parameters via MMI and are local to the BSS.
Power Control Messages	X	X	C	All power control messages, each timestamped, see GSM 08.58 [5].
Timed Power Control Message	X	C	X	X power control messages received before and Y power control messages received after a handover, each timestamped. X & Y are operator configurable parameters via MMI and are local to the BSS.
Record extensions	O	O	O	A set of network/ manufacturer specific extensions to the record.

7.4 HLR Trace Record Content

The following types of fields are supported in the 2 HLR trace record types:

Table 11: HLR Trace Record Content

Field	HLR Trace Type		Description
	Basic	**Detailed**	
Invoking Event	M	M	Event invoking trace.
Served MSISDN	C	C	Primary MSISDN of the party being traced.
MSC Address	C	C	Entity number of the serving MSC (GSM 12.05 [10] annex B).
VLR number	C	C	Entity number of the serving VLR (GSM 12.05 [10] annex B).
SS Information	C	C	A list of information related to any SS actions carried out during the period of the trace. The SS Information contains the SS Code for each SS Action, the Basic Services for which each SS action was carried out, the type of each SS action carried out, a list of SS parameters associated with each SS action, the result of each SS action and the Invoke Id allocated for each SS Action.
Subscriber data	O	C	The subscriber data sent to the VLR after a location update.
Roaming number	C	C	The roaming number returned from the serving VLR.
SM Delivery outcome	C	C	The outcome of a MT SM delivery.
Alert reason	C	C	Indicates the reason why the SM service centre was alerted.
Service Centre address	C	C	The address of the SM service centre.
MAP interface messages	X	C	A sequential list of all MAP messages passed to and from the Tracing HLR.
Event Start Time	C	C	The time the event was started. It must always appear in case the trace record is already being compiled and the event belonging to this event record for this same subscriber occurs.
Event Stop Time	C	C	The time the event was finished. It must always appear in case the trace record is still being compiled due to an ongoing event and the event belonging to this event record finishes.
Event Number	M	M	The Event Number is used to identify tracing data belonging to the same event.
Record extensions	O	O	A set of network/ manufacturer specific extensions to the record.

7.5 Trace Record fields

Only those fields which are not defined in GSM 12.05 [10] annex B or are named differently from an identical field in GSM 12.05 [10] annex B are included here. Only supplementary information is included in this clause; where a description in tables 9 - 11 is sufficient, no additional information is provided.

7.5.1 Radio channel information

When instructing the mobile to move to a new channel during procedures like Assignment, Immediate Assignment and Handover, the BTS must give the mobile all the necessary information such as frequency (frequencies if hopping), timeslot number, channel type etc. This is done using the Channel Description field defined in GSM 04.08 [2]. The structure of the Channel Description depends on whether or not frequency hopping is in use. These two cases are described below:

No Frequency Hopping

Channel Description (GSM 04.08 [2]), contains the following:

> Channel Type (TCH, SDCCH etc.);
> Timeslot Number (0 to 7);
> TDMA Offset (0 to 7, used to identify SDCCH etc. within a timeslot);
> Training sequence number;
> Absolute Radio Carrier Frequency number.

Frequency Hopping

Channel Description (GSM 04.08 [2]), contains the following:

> Channel Type (TCH, SDCCH etc.);
> Timeslot Number (0 to 7);
> TDMA Offset (0 to 7, used to identify SDCCH etc. within a timeslot);
> Training sequence number;
> Hopping Sequence Number;
> Mobile Allocation Index Offset.

In this case, the channel description does not contain the list of frequencies to be used for hopping and an additional field indicating the mobile allocation is required. The mobile allocation is the set of frequencies to be used for hopping and is obtained from any of the following:

a) Cell Channel Description and Mobile Allocation;
b) Frequency Channel sequence;
c) Frequency List;
d) Frequency Short List.

In summary, to identify a GSM channel unambiguously the "Channel Description" field is sufficient on its own when frequency hopping is not used but mobile allocation is also required when hopping is in use.

8 Creation of Trace Records

8.1 General

As has already been stated, the sequence of events for the creation of a trace record is as follows:

1) Trace is activated for a particular IMSI or IMEI.

2) The subscriber undertakes such action as to cause an invoking event to start.

3) The compilation of a trace record commences in the NEF as described in the Trace Type and under the control of the traceRecord attribute recordCriteria. This allows trace records to be produced at times other than when the invoking event ends, e.g. after a specific event has occurred.

4) If a further invoking event occurs trace data related to this event is collected in the same trace record.

5) All invoking events end or the recordCriteria attribute is satisfied, (see 3) above).

6) The record is forwarded to the OSF or local filestore (depending on priority).

In certain circumstances it may be undesirable for the invoking event to have to end before the record is forwarded to the OSF or local filestore. Examples of these circumstances may be:

1) The operator requires to know a subscriber's whereabouts at the moment he starts making a call.

2) The operator requires to know when a handover occurs, as soon as it occurs.

3) The buffer in the NEF may be too full to contain any more trace record data.

This is resolved through the use of the attribute recordCriteria in the traceControl object. When this attribute is set to anything other than noCriteria, records are forwarded to either the filestore or the OSF as soon as the specified criteria is satisfied.

8.2 Trace Record Control

8.2.1 General

The trace record collection and generation processes are controlled by the **traceControl** managed object class. There shall be one, and only one, instance of this object class for each NEF that supports the trace function. This object carries out the following functions:

1) to cause the data to be collected in the NEF as defined by the Trace Type;
2) to define the criteria by which records are generated;
3) to generate the trace record notifications.

System management functions:

> Create traceControl;
> Delete traceControl;
> Get Attribute;
> Set Attribute.

Notifications:

> stateChange;
> objectCreation;
> objectDeletion;
> attributeValueChange;
> traceReport.

8.2.2 Attributes

There is one instance of this object class in each NEF that supports the trace function. It contains the following attributes:

Name	M/O	Value-Set
traceControlId	RDN	Single
administrativeState	M	Single
operationalState	M	Single
recordCriteria	M	Single
eventTypes	O	Single

traceControlId

This attribute is a unique identifier for the traceControl MOI in the NEF and is used as an RDN.

administrativeState

This attribute defines the administrative state of the traceControl MOI in the NEF (Recommendation X.731 [16]).

operationalState

This attribute defines the operational status of the traceControl MOI in the NEF (Recommendation X.731 [16]).

recordCriteria

This attribute, if set, defines the criteria by which trace records are generated in the NEF. It may have one or more of the following values:

noCriteria	The NEF will not output trace records of the event type.
event	The NEF will output a trace record every time a particular recordable event occurs, the nature of that event being defined in the attribute eventTypes.

In all cases, a trace record will be produced at the end of the invoking event, or if other criteria are set by the manufacturer, when these criteria are met.

eventTypes

This attribute defines a set of recordable events, the appearance of any will trigger a trace record to be output, assuming the "event" value is set in the recordCriteria attribute.

8.2.3 Other Trace Record Criteria

Regardless of the trace record criteria set by the operator, there are circumstances under which a trace record may be generated, with the criteria being set by the manufacturer. These will usually be due to a lack of resources such as "Buffer Full" or "Processor Overload".

9 Trace Record Transfer

9.1 General

This clause is concerned solely with the management of the trace record collection process. This service component controls the transfer of the trace records from the NEFs to the OSF. The conceptual model is illustrated in figure 4, which employs both the event report function (CCITT X.734 [14]) and the log control function (CCITT X.735 [15]).

The trace control function collects traceable events within the NEF and formats them into trace records. These trace records may be stored locally within the NEF filestore or transferred to the OSF in the form of event reports. This is controlled by means of the "priority" indicator, which is a part of the trace type. If the "priority" indicator is not set then the trace records shall stored within the local filestore and subsequently transferred to the OSF in bulk via FTAM.

If a trace is activated with the "priority" indicator set then the trace records shall be sent to the OSF either direct by the trace control function or through Event Forwarding Discriminators (EFDs).

If EFDs are used then all trace records are offered to the EFDs. The EFDs determine which of the records are to be transmitted to the OSF in the form of event reports and the Operation System Id field in the header is ignored. The EDFs have complex filter constructs which allow the operator to define the criteria for destinations and filters.

If EFDs are not used then all priority records are forwarded to the OSF whose address is given in the Operation System Id field. The NEF is required to supply additional information to provide the OS management application entity title.

Finally, the trace records may also be stored in the form of log entries within the log of the NEF. It is up to the operator to decide if the log function is needed in parallel with the event reporting and file store. Once stored, the log records may be individually accessed by the OSF via the appropriate object management functions. Care should be taken of filter criteria for log records to avoid unnecessary overheads.
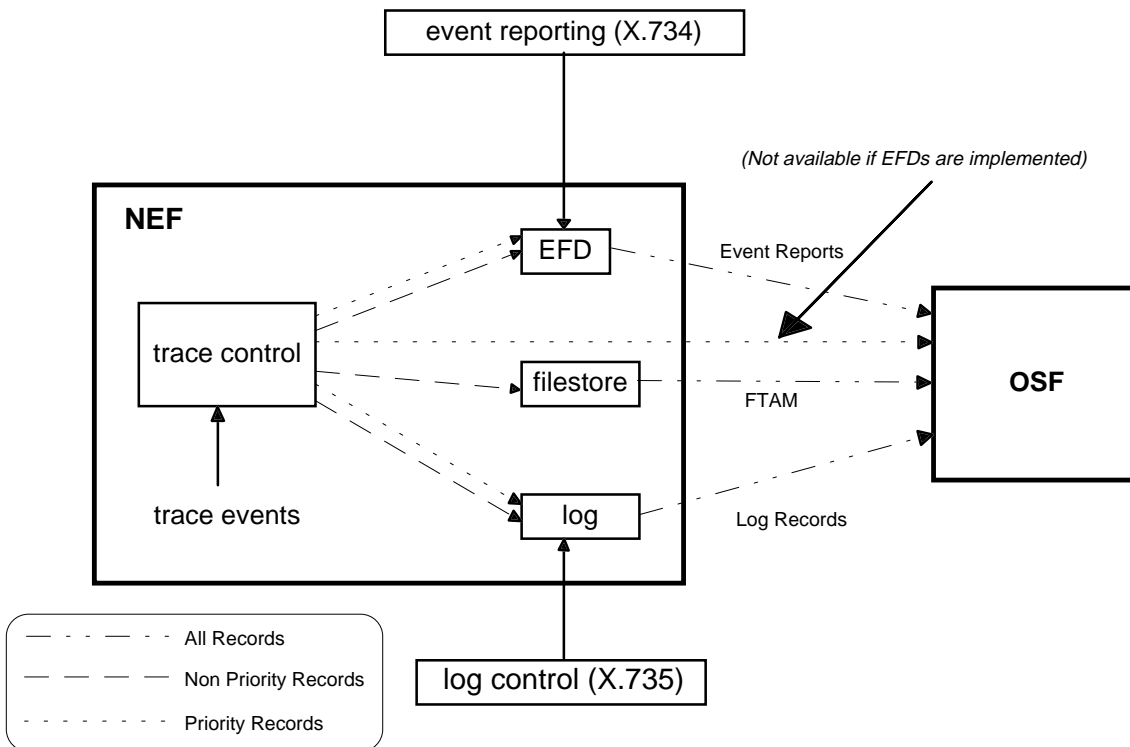
Figure 4: Data collection model

This service component contains the following groups of TMN management functions:

- bulk record transfer;
- event reporting;
- log control;
- log access.

9.2 Transfer of Records

9.2.1 Bulk record transfer

This group of TMN functions is concerned with the bulk transfer of trace records from the NEF record filestore to the OSF.

The trace records shall be transferred from the NEF to the OSF by the use of FTAM services. For further details of the use of FTAM see GSM 12.01 [8].

In addition to the simple file transfer services provided by FTAM, peer-to-peer application process communication may also be supported. The use of CMIS services for the uploading of files from the NEF to the OSF is specified in the GSM 12.00 [7].

When the procedure defined in GSM 12.00 [7] and GSM 12.01 [8] are used to transfer the trace records, the file type shall be traceRecords and the format of the file is given by the type TraceFileFormat.

9.2.2 Log control

This function permits the trace record to be stored and retrieved from logs within the NEF. The logging of these records is performed in accordance with the log control function specified in CCITT X.735 [15] and no additional management functions are required.

9.2.3 Log access

This TMN function controls the access to the log described above. Each log defined may contain one or more log entries. Each log entry contains a single trace record.

> NOTE: The term log entry has been used instead of the term log record to avoid confusion between records contained within the local filestore and the records stored within logs.

For further details concerning to use of logs, see CCITT X.735 [15].

The following system management functions are required:

    Get/Delete traceLogEntry

9.2.4 Event Reporting

9.2.4.1 Event Forwarding Discriminators

For short-term recording of tracing events and for more complicated filter conditions the event forwarding discriminator construct defined in CCITT X.734 [14] and CCITT X.721 [13] can be employed. The event forwarding discriminator construct is extremely flexible permitting the combination of a number of fields and logical operations with a wide variety of scheduling options. The EFD also controls the destinations to which the event reports are sent. Several such filters may be defined and scheduled for operation at different times and for different time periods.

The following system management functions are required:

    Create/Set/Get/Delete eventForwardingDiscrimator

9.2.4.2 Direct Transfer by Trace Control Function

This function permits the NEF to transmit trace records direct to the OSF. In general the trace record shall be sent on completion of the call or the traceable event. This function is controlled by means of the "priority" indicator, which is contained in the trace type. If the priority indicator is not set, then the trace records shall be stored on file within the NE filestore. These records may be subsequently collected via bulk record transfer as described in subclause 9.2.1. If the trace type specified on activation includes the "priority" indicator then all of the records shall be sent via trace reports to the OSF specified by the operation system id.

> NOTE: As the operations system id. provided is an AddressString (e.g. CCITT E.164 number) some form of translation or directory service may be required within the NE in order to provide the appropriate OS management application entity title.

The following system management functions are required:

Notification traceReport

10 Managed Object Model

10.1 Naming Hierarchy

The naming (containment) tree for the objects defined within this clause is illustrated in figure 5. It should be noted that the GSM 12.08 object classes are shown relative to the "managedElement". The MO traceControl is contained in every NEF (mscFunction, hlrFunction and bssFunction from GSM 12.00 [7]) that supports trace functionality. For further details of the upper layers of the containment tree see GSM 12.00 [7]. For further details concerning the log class see CCITT X.721 [13].
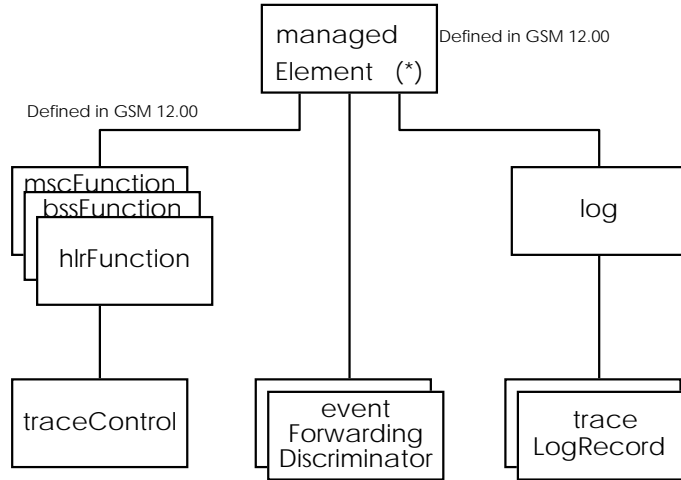
Figure 5: Trace Record Transfer Containment Tree

10.2 Inheritance

The inheritance tree for GSM 12.08 is illustrated in figure 6 below. The object classes "log", "logRecord", "eventLogRecord" and "eventForwardingDiscriminator" are defined in CCITT X.721 [13].
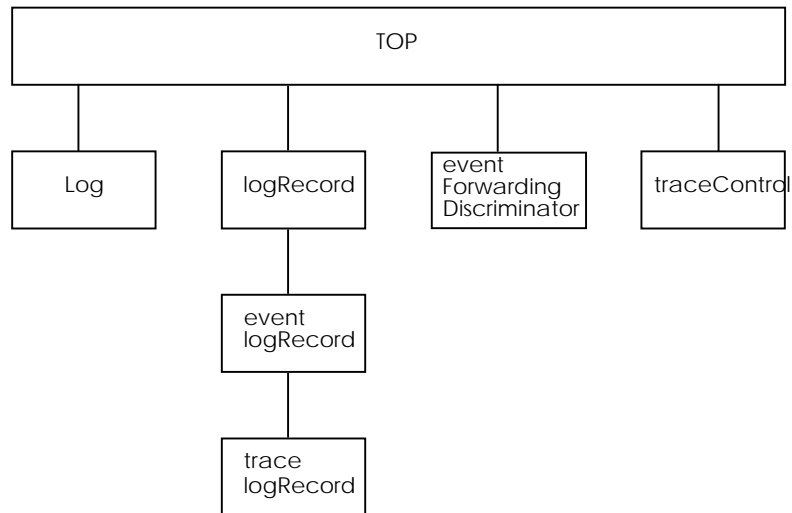
Figure 6: Trace Record Transfer Inheritance Tree

10.3   Object Classes

10.3.1   tracedHomeSubscriberInHlr

tracedHomeSubscriberInHlr MANAGED OBJECT CLASS
        DERIVED FROM
            "CCITT Rec. X.721: 1992":top;
        CHARACTERIZED BY
            tracedHomeSubscriberInHlrPackage,
"CCITT Rec. M.3100: 1992":   createDeleteNotificationsPackage,
"CCITT Rec. M.3100: 1992":   attributeValueChangeNotificationPackage;

        CONDITIONAL PACKAGES
            operationSystemIdPackage   PRESENT IF "an instance supports it";

REGISTERED AS {Trace-DataTypes.gsm-1208-objectClass 100};

tracedHomeSubscriberInHlrPackage PACKAGE
        BEHAVIOUR
            tracedHomeSubscriberInHlrBehaviour;
        ATTRIBUTES
            imsi   GET,
            traceActivatedInVlr   GET,
            traceReference   GET,
            traceType   GET,
            hlrTraceType   GET,
            mapErrorOnTrace   GET
        ;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 100};

tracedHomeSubscriberInHlrBehaviour BEHAVIOUR
        DEFINED AS
            "see GSM 12.08 clause 5.5.1.1";
operationSystemIdPackage PACKAGE
        BEHAVIOUR
            operationSystemIdBehaviour;
        ATTRIBUTES
            operationSystemId   GET;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 200};

operationSystemIdBehaviour BEHAVIOUR
        DEFINED AS
                "This package provides the attribute to specify destination operation system. The use
                of this attribute is described in chapter Trace record transfer. The package is
                conditional to allow the attribute operationSystemId to be optional.";

10.3.2   tracedForeignSubscriberInVlr

tracedForeignSubscriberInVlr MANAGED OBJECT CLASS
        DERIVED FROM
            "CCITT Rec. X.721: 1992":top;
        CHARACTERIZED BY
            tracedForeignSubscriberInVlrPackage,
"CCITT Rec. M.3100: 1992":   createDeleteNotificationsPackage,
"CCITT Rec. M.3100: 1992":   attributeValueChangeNotificationPackage;

        CONDITIONAL PACKAGES
            operationSystemIdPackage   PRESENT IF "an instance supports it";

REGISTERED AS {Trace-DataTypes.gsm-1208-objectClass 200};

```
tracedForeignSubscriberInVlrPackage PACKAGE
      BEHAVIOUR
            tracedForeignSubscriberInVlrBehaviour;
      ATTRIBUTES
            imsi                                GET,
            foreignSubscriberRegisteredInVlr    GET,
            traceReference                      GET,
            traceType                           GET
      ;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 300};


tracedForeignSubscriberInVlrBehaviour BEHAVIOUR
      DEFINED AS
            "see GSM 12.08 clause 5.6.1.1";
```

10.3.3   tracedEquipmentInVlr

```
tracedEquipmentInVlr MANAGED OBJECT CLASS
      DERIVED FROM
            "CCITT Rec. X.721: 1992":top;
      CHARACTERIZED BY
            tracedEquipmentInVlrPackage,
"CCITT Rec. M.3100: 1992":    createDeleteNotificationsPackage,
"CCITT Rec. M.3100: 1992":    attributeValueChangeNotificationPackage;

      CONDITIONAL PACKAGES
            operationSystemIdPackage            PRESENT IF "an instance supports it";

REGISTERED AS {Trace-DataTypes.gsm-1208-objectClass 300};


tracedEquipmentInVlrPackage PACKAGE
      BEHAVIOUR
            tracedEquipmentInVlrBehaviour;
      ATTRIBUTES
            imei                      GET,
            equipmentRegisteredInVlr  GET,
            traceReference            GET,
            traceType                 GET
      ;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 400};


tracedEquipmentInVlrBehaviour BEHAVIOUR
      DEFINED AS
            "see GSM 12.08 clause 5.6.1.2";
```

10.3.4 Trace control

This managed object class represents the trace collection process and generates the trace report notifications. There shall be one, and only one, instance of this object class for each NEF that supports the trace function.

```
traceControl MANAGED OBJECT CLASS
        DERIVED FROM "CCITT Rec. X.721: 1992":top;
        CHARACTERIZED BY
        traceControlPackage,
        "CCITT Rec. M.3100: 1992": attributeValueChangeNotificationPackage,
        "CCITT Rec. M.3100: 1992": createDeleteNotificationsPackage,
        "CCITT Rec. M.3100: 1992": stateChangeNotificationPackage;

CONDITIONAL PACKAGES
        eventTypeCriteriaPackage            PRESENT IF "an instance supports it"
;
REGISTERED AS {Trace-DataTypes.gsm-1208-objectClass 400};

traceControlPackage PACKAGE
        BEHAVIOUR
        traceControlBehaviour BEHAVIOUR
        DEFINED AS
                "This managed object class is employed to generate trace report notifications. There
                can be only one instance of this object class in each NEF that supports trace
                functionality.
                For the administrativeState, the value LOCKED causes all tracing activity in the NEF to
                cease. The value UNLOCKED allows tracing activity. The value SHUTTING-DOWN
                prevents any new invocation of a trace. Current invocations will continue until they are
                finished. When all current invocations finish, the state will automatically transit to
                LOCKED.";;
        ATTRIBUTES
            traceControlId              GET,
            "CCITT Rec. X.721: 1992": administrativeState      GET-REPLACE,
            "CCITT Rec. X.721: 1992": operationalState GET,
            recordCriteria             GET-REPLACE ADD-REMOVE;
        NOTIFICATIONS
            traceReport;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 500};

eventTypeCriteriaPackage PACKAGE
        BEHAVIOUR
            eventTypeCriteriaBehaviour     BEHAVIOUR
        DEFINED AS
                "This package provides the attribute to specify eventType record generation criteria.
                The use of this attribute is described in clause 8.2.2. The package is conditional to
                allow the attribute to be optional.";;

        ATTRIBUTES
            eventTypes                 GET-REPLACE ADD-REMOVE;

REGISTERED AS {Trace-DataTypes.gsm-1208-package 520};
```

10.3.5 Trace log record

This managed object class is a subclass of the "eventLogRecord" class described in CCITT X.735 and defined in CCITT X.721 and therefore inherits all of the properties of both the "logRecord" and "eventLogRecord" classes. This includes the name binding "logRecord-log" defined in CCITT X.721.

```
traceLogRecord MANAGED OBJECT CLASS
      DERIVED FROM "CCITT Rec. X.721: 1992":eventLogRecord;
      CHARACTERIZED BY
      traceLogRecordPackage;
CONDITIONAL PACKAGES
      traceReferenceLogPackage          PRESENT IF "an instance supports it",
      mscBssTraceTypeLogPackage         PRESENT IF "an instance supports it",
      hlrTraceTypeLogPackage            PRESENT IF "an instance supports it",
      mscBssTraceTypeUsedLogPackage     PRESENT IF "an instance supports it",
      hlrTraceTypeUsedLogPackage        PRESENT IF "an instance supports it";
REGISTERED AS {Trace-DataTypes.gsm-1208-objectClass 500};


traceLogRecordPackage  PACKAGE
            BEHAVIOUR
            traceLogRecordBehaviour BEHAVIOUR
                  DEFINED AS "This managed object is used to store a single trace record.";;
            ATTRIBUTES
            traceRecordContent          GET;
REGISTERED AS {Trace-DataTypes.gsm-1208-package 600};


traceReferenceLogPackage PACKAGE
BEHAVIOUR
            traceReferenceLogBehaviour BEHAVIOUR
      DEFINED AS
                  "This package provides the attribute to specify  traceReference for trace report
                  searching criteria in the Log.  Optional.";;

      ATTRIBUTES
            traceReference              GET;


REGISTERED AS {Trace-DataTypes.gsm-1208-package 610};


mscBssTraceTypeLogPackage PACKAGE
BEHAVIOUR
            mscBssTraceTypeLogBehaviour BEHAVIOUR
      DEFINED AS
                  "This  package  provides  the  attribute  to  specify  searching  criteria  to  be
                  mscBssTraceType  of trace report in the Log.  Optional.";;

      ATTRIBUTES
            mscBssTraceType             GET;


REGISTERED AS {Trace-DataTypes.gsm-1208-package 620};


hlrTraceTypeLogPackage PACKAGE
BEHAVIOUR
            hlrTraceTypeLogBehaviour BEHAVIOR
      DEFINED AS
                  "This package provides the attribute to specify searching criteria to be hlrTraceType  of
                  trace report in the Log.  Optional.";;

      ATTRIBUTES
            hlrTraceType                GET;


REGISTERED AS {Trace-DataTypes.gsm-1208-package 630};


mscBssTraceTypeUsedLogPackage PACKAGE
BEHAVIOUR
            mscBssTraceTypeUsedLogBehaviour BEHAVIOUR
      DEFINED AS
                  "This  package  provides  the  attribute  to  specify  searching  criteria  to  be
                  mscBssTraceTypeUsed  of trace report in the Log.  Optional.";;
```

ATTRIBUTES
mscBssTraceTypeUsed GET;

REGISTERED AS {Trace-DataTypes.gsm-1208-package 640};

hlrTraceTypeUsedLogPackage PACKAGE
BEHAVIOUR
hlrTraceTypeUsedLogBehaviour BEHAVIOUR
DEFINED AS
"This package provides the attribute to specify searching criteria to be
hlrTraceTypeUsed of trace report in the Log. Optional.";;

ATTRIBUTES
hlrTraceTypeUsed   GET;

REGISTERED AS {Trace-DataTypes.gsm-1208-package 650};

10.3.6   Log

This managed object class is described in CCITT X.735 and defined in CCITT X.721.

10.3.7   Event Forwarding Discriminators

The use of event forwarding discriminators (EFDs) is described in detail in CCITT X.734. The object class
itself is a subclass of the "discriminator" object class. Both discriminator and event forwarding
discriminator classes are defined in CCITT X.721.

10.4   Attributes

10.4.1   traceActivatedInVlr

traceActivatedInVlr ATTRIBUTE
WITH ATTRIBUTE SYNTAX
Trace-DataTypes.TraceStatus;
MATCHES FOR
EQUALITY;
BEHAVIOUR
traceActivatedInVlrBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 10};

traceActivatedInVlrBehaviour BEHAVIOUR
DEFINED AS
"see GSM 12.08 clause 5.5.1.2.1";

10.4.2   foreignSubscriberRegisteredInVlr

foreignSubscriberRegisteredInVlr ATTRIBUTE
WITH ATTRIBUTE SYNTAX
Trace-DataTypes.TraceStatus;
MATCHES FOR
EQUALITY;
BEHAVIOUR
foreignSubscriberRegisteredInVlrBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 20};

foreignSubscriberRegisteredInVlrBehaviour BEHAVIOUR
DEFINED AS
"see GSM 12.08 clause 5.6.1.3.1";

10.4.3 equipmentRegisteredInVlr

```
equipmentRegisteredInVlr ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
                Trace-DataTypes.TraceStatus;
        MATCHES FOR
                EQUALITY;
        BEHAVIOUR
                equipmentRegisteredInVlrBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 30};

equipmentRegisteredInVlrBehaviour BEHAVIOUR
        DEFINED AS
                "see GSM 12.08 clause 5.6.1.3.2";
```

10.4.4 mapErrorOnTrace

```
mapErrorOnTrace ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
                Trace-DataTypes.MapErrorOnTrace;
        MATCHES FOR
                EQUALITY, ORDERING;
        BEHAVIOUR
                mapErrorOnTraceBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 40};

mapErrorOnTraceBehaviour BEHAVIOUR
        DEFINED AS
                "see GSM 12.08 clause 5.5.1.2.1";
```

10.4.5 IMEI

```
imei ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
                MAP-CommonDataTypes.IMEI;
        MATCHES FOR
                EQUALITY, ORDERING;
        BEHAVIOUR
                imeiBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 50};

imeiBehaviour BEHAVIOUR
        DEFINED AS
                "see GSM 12.08 clause 5.6.1.3.2";
```

10.4.6 IMSI

```
imsi ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
                MAP-CommonDataTypes.IMSI;
        MATCHES FOR
                EQUALITY, ORDERING;
        BEHAVIOUR
                imsiBehaviour;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 60};

imsiBehaviour BEHAVIOUR
        DEFINED AS
                "see GSM 12.08 clause 5.6.1.3.1";
```

10.4.7 Trace record content

```
traceRecordContent ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceRecord;
     BEHAVIOUR
     traceRecordContentBehaviour BEHAVIOUR
     DEFINED AS
     "This attribute contains the contents of a trace record.";;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 70};
```

10.4.8 Trace control id.

```
traceControlId ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceControlId;
     MATCHES FOR EQUALITY;
     BEHAVIOUR
     traceControlIdBehaviour BEHAVIOUR
     DEFINED AS
     "This attribute uniquely identifies a trace control object.";;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 80};
```

10.4.9 HLR Trace type

```
hlrTraceType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceType;
     MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 90};
```

10.4.10 Trace reference

```
traceReference ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceReference;
     MATCHES FOR EQUALITY, ORDERING;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 100};
```

10.4.11 Trace type

```
traceType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceType;
     MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 110};
```

10.4.12 Record criteria

```
recordCriteria ATTRIBUTE
     WITH ATTRIBUTE SYNTAX Trace-DataTypes.RecordCriteria;
     MATCHES FOR EQUALITY;
     BEHAVIOUR
     recordCriteriaBehaviour BEHAVIOUR
     DEFINED AS
                "This attribute specifies the criteria for the generation of a trace record by the network
                element.";;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 120};
```

10.4.13 Event types

eventTypes ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Trace-DataTypes.EventTypes;
      MATCHES FOR EQUALITY;
      BEHAVIOUR
      eventTypeBehaviour BEHAVIOUR
      DEFINED AS
            "This attribute specifies the type of event triggering the generation of a trace record by the
network element.";;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 140};

10.4.14 Operation system ID

operationSystemId ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Trace-DataTypes.OmcId;
      MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 160};

10.4.15 Operational State

This attribute is described in Recommendation X.731 and the syntax is defined in X.721.

10.4.16 Administrative State

This attribute is described in Recommendation X.731 and the syntax is defined in X.721.

10.4.17 MSC BSS trace type used

mscBssTraceTypeUsed ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceType;
      MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 170};

10.4.18 HLR trace type used

hlrTraceTypeUsed ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceType;
      MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 180};

10.4.19 MSC BSS trace type

mscBssTraceType ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Trace-DataTypes.TraceType;
      MATCHES FOR EQUALITY;
REGISTERED AS {Trace-DataTypes.gsm-1208-attribute 190};

10.5 Notifications

10.5.1 General

All notifications listed below are defined in CCITT X.721:

attributeValueChange
objectCreation
objectDeletion
stateChange

10.5.2 Trace report

```
traceReport              NOTIFICATION
      BEHAVIOUR
           traceReportBehaviour;
      WITH INFORMATION SYNTAX Trace-DataTypes.TraceRecord
           AND ATTRIBUTE IDS
           traceReference                traceReference,
           mscBssTraceType               mscBssTraceType,
           hlrTraceType                  hlrTraceType,
           mscBssTraceTypeUsed           mscBssTraceTypeUsed,
           hlrTraceTypeUsed              hlrTraceTypeUsed;
REGISTERED AS {Trace-DataTypes.gsm-1208-notification 100};
```

```
traceReportBehaviour BEHAVIOUR
      DEFINED AS
                "This notification is issued by the trace control function to transmit a trace report to the
                OS. The attribute Ids may be used by Event Forwarding Discriminators to specify
                additional filter conditions.";
```

10.6   Name Bindings

10.6.1   tracedHomeSubscriberInHlr-hlrFunction Name Binding

```
tracedHomeSubscriberInHlr-hlrFunction NAME BINDING
      SUBORDINATE OBJECT CLASS tracedHomeSubscriberInHlr;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":hlrFunction;
      WITH ATTRIBUTE imsi;
      BEHAVIOUR tracedHomeSubscriberInHlr-hlrFunctionBhv;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 100};
```

```
tracedHomeSubscriberInHlr-hlrFunctionBhv BEHAVIOUR
      DEFINED AS
           "see GSM 12.08 clause 5.5";
```

10.6.2   tracedForeignSubscriberInVlr-vlrFunction Name Binding

```
tracedForeignSubscriberInVlr-vlrFunction NAME BINDING
      SUBORDINATE OBJECT CLASS tracedForeignSubscriberInVlr;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":vlrFunction;
      WITH ATTRIBUTE imsi;
      BEHAVIOUR tracedForeignSubscriberInVlr-vlrFunctionBhv;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 200};
```

```
tracedForeignSubscriberInVlr-vlrFunctionBhv BEHAVIOUR
      DEFINED AS
           "see GSM 1208 clause 5.6";
```

10.6.3   tracedEquipmentInVlr-vlrFunction Name Binding

```
tracedEquipmentInVlr-vlrFunction NAME BINDING
      SUBORDINATE OBJECT CLASS tracedEquipmentInVlr;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":vlrFunction;
      WITH ATTRIBUTE imei;
      BEHAVIOUR tracedEquipmentInVlr-vlrFunctionBhv;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 300};

tracedEquipmentInVlr-vlrFunctionBhv BEHAVIOUR
      DEFINED AS
            "see GSM 1208 clause 5.6";
```

10.6.4   traceLogRecord-Log Name Binding

```
traceLogRecord-Log NAME BINDING
      SUBORDINATE OBJECT CLASS traceLogRecord;
      NAMED BY SUPERIOR OBJECT CLASS "CCITT Rec. X.721: 1992":log;
      WITH ATTRIBUTE "CCITT Rec. X.721: 1992":logRecordId;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 400};
```

10.6.5   traceControl-hlrFunction Name Binding

```
traceControl-hlrFunction NAME BINDING
      SUBORDINATE OBJECT CLASS traceControl;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":hlrFunction;
      WITH ATTRIBUTE traceControlId;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 500};
```

10.6.6   traceControl-mscFunction Name Binding

```
traceControl-mscFunction NAME BINDING
      SUBORDINATE OBJECT CLASS traceControl;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":mscFunction;
      WITH ATTRIBUTE traceControlId;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 600};
```

10.6.7   traceControl-bssFunction Name Binding

```
traceControl-bssFunction NAME BINDING
      SUBORDINATE OBJECT CLASS traceControl;
      NAMED BY
      SUPERIOR OBJECT CLASS "prETS 300 612-1:1995":bssFunction;
      WITH ATTRIBUTE traceControlId;
      CREATE;
      DELETE;
REGISTERED AS {Trace-DataTypes.gsm-1208-nameBinding 700};
```

11 Syntax

```
Trace-DataTypes {ccitt (0) identified-organization (4) etsi (0) mobileDomain (0) gsm-Operation-
Maintenance (3) gsm-12-08 (8) informationModel (0) asn1Module (2)}

DEFINITIONS IMPLICIT TAGS    ::=

BEGIN

-- EXPORTS everything

IMPORTS

gsm-12-08
FROM GSM-DomainDefinitions {ccitt (0) identified-organization (4) etsi (0) mobileDomain (0) gsm-
Operation-Maintenance (3) gsm-12-30 (30) informationModel (0) asn1Module (2) gsm-OM-
DomainDefinitions (0) version (1)}

SS-Info,
InterrogateSS-Res,
SS-UserData,
Password,
RegisterSS-Arg,
SS-ForBS-Code,
USSD-Arg,
USSD-Res,
GuidanceInfo
FROM MAP-SS-DataTypes {ccitt (0) identified-organization (4) etsi (0) mobileDomain (0) gsm-
Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}

AddressString, ISDN-AddressString, ISDN-SubaddressString, BasicServiceCode, IMSI, IMEI
FROM MAP-CommonDataTypes { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network
(1) modules (3) map-CommonDataTypes (18) version2 (2) }

BearerServiceCode
FROM MAP-BS-Code { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network (1)
modules (3) map-BS-Code (20) version2 (2) }

TeleserviceCode
FROM MAP-TS-Code { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network (1)
modules (3) map-TS-Code (19) version2 (2) }

SS-Code
FROM MAP-SS-Code { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network (1)
modules (3) map-SS-Code (15) version2 (2) }

SubscriberData
FROM MAP-MS-DataTypes { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network
(1) modules (3) map-MS-DataTypes (11) version2 (2) }

SM-DeliveryOutcome,
AlertReason
FROM MAP-SM-DataTypes { ccitt identified-organization (4) etsi(0) mobileDomain (0) gsm-Network
(1) modules (3) map-SM-DataTypes (16) version2 (2) }

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

ObjectInstance
FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules (0) protocol (3)}

ManagementExtension
FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2 (2) asn1Module(2) 1}


AOCParameters, Diagnostics, LocationAreaAndCell, IMSIorIMEI
FROM GSM1205-DataTypes { ccitt (0) identified-organization (4) etsi (0) mobileDomain (0) gsm-
Operation-Maintenance (3) gsm-12-05 (5) informationModel (0) asn1Module (2) 1 };
```

```
-------------------------------------------------------------------------------
--                      OBJECT IDENTIFIERS
-------------------------------------------------------------------------------
gsm-1208-informationModel   OBJECT IDENTIFIER ::= {gsm-12-08 informationModel (0)}

gsm-1208-objectClass        OBJECT IDENTIFIER ::= {gsm-1208-informationModel
                                 managedObjectClass (3)}

gsm-1208-package            OBJECT IDENTIFIER ::= {gsm-1208-informationModel
                                 package (4)}

gsm-1208-nameBinding        OBJECT IDENTIFIER ::= {gsm-1208-informationModel
                                 nameBinding (6)}

gsm-1208-attribute          OBJECT IDENTIFIER ::= {gsm-1208-informationModel
                                 attribute (7)}

gsm-1208-notification       OBJECT IDENTIFIER ::= {gsm-1208-informationModel
                                 notification (10)}


    --these values are based on ETR 128 June 1994 (GSM 12.30, ETSI object
    --identifier tree; Common domain Mobile domain O&M managed object
    --registration definition.
    --Resulting values are e.g. {0 4 0 0 3 8 0 3 zzz} for gsm-1208-objectClass.

-------------------------------------------------------------------------------
--                      TRACE ACTIVATION
-------------------------------------------------------------------------------


TraceStatus             ::= BOOLEAN
    -- TRUE  = active
    -- FALSE = active pending

TraceType               ::= OCTET STRING (SIZE(1))
    -- see GSM 12.08  subclause 6.1 for encoding of mscBssTraceType
    -- see GSM 12.08  subclause 6.2 for encoding of hlrTraceType

MapErrorOnTrace         ::= ENUMERATED
{
    noError                 (0),
    systemFailure           (1),
    dataMissing             (2),
    unexpectedDataValue     (3),
    facilityNotSupported    (4),
    unidentifiedSubscriber  (5),
    tracingBufferFull       (6)
}
```

```
-----------------------------------------------------------------------------
--                      TRACE RECORD
-----------------------------------------------------------------------------


TraceRecord             ::= SET
{
    tracedParty             [0] IMSIorIMEI,
    traceReference          [1] TraceReference,
    transactionID           [2] TransactionID                     OPTIONAL,
    omcID                   [3] OmcId                             OPTIONAL,
    mscBssTraceType         [4] TraceType                         OPTIONAL,
    hlrTraceType            [5] TraceType                         OPTIONAL,
    mscBsstraceTypeUsed     [6] TraceType                         OPTIONAL,
    hlrTraceTypeUsed        [7] TraceType                         OPTIONAL,
    startTime               [8] StartTime                         OPTIONAL,
    endTime                 [9] EndTime                           OPTIONAL,
    recordingEntity         [10] RecordingEntity,
    traceEventRecords       [11] SET OF TraceEventRecord,
    sequenceNumber          [12] INTEGER                          OPTIONAL,
    recordReason            [13] ReasonForRecord                  OPTIONAL
}
ReasonForRecord         ::= SEQUENCE
{
    recordCriteria          [0] RecordCriterionUsed,              OPTIONAL,
    eventType               [1] EventTypeUsed                     OPTIONAL
}

RecordingEntity         ::= CHOICE
{
    number                  [0] ISDN-AddressString,
    name                    [1] GraphicString,
    bssIdentifier           [2] ObjectInstance
}

EndTime                 ::= GeneralizedTime

StartTime               ::= GeneralizedTime

TraceReference          ::= OCTET STRING
    -- see GSM 08.08

TransactionID           ::= OCTET STRING
    -- see GSM 08.08

OmcId                   ::= AddressString
    -- see GSM 08.08

TraceEventRecord        ::= CHOICE
{
    mscEventRecord          [0] MSCEventRecord,
    bssEventRecord          [1] BSSEventRecord,
    hlrEventRecord          [2] HLREventRecord
}
```

```
-------------------------------------------------------------------------------
--                BSS TRACE RECORD CONTENTS
-------------------------------------------------------------------------------

BSSEventRecord          ::= SET
{
    invokingEvent          [0] BSCInvokingEvent                     OPTIONAL,
    btsId                  [1] SEQUENCE OF BtsId                    OPTIONAL,
    trxId                  [2] SEQUENCE OF TRXID                    OPTIONAL,
    trauId                 [3] SEQUENCE OF TranscoderID             OPTIONAL,
    radioChannelInfo       [4] SEQUENCE OF RadioChannelInfo         OPTIONAL,
    requestType            [5] SEQUENCE OF TimedEstablishmentCause  OPTIONAL,
    endIndication          [6] SEQUENCE OF EndIndication            OPTIONAL,
    msPower                [7] SEQUENCE OF MsTxPower                 OPTIONAL,
    bsPower                [8] SEQUENCE OF BsTxPower                 OPTIONAL,
    timingAdvance          [9] SEQUENCE OF TimedTimingAdvance       OPTIONAL,
    msClassmark1           [10] SEQUENCE OF TimedMsClassmark1       OPTIONAL,
    msClassmark2           [11] SEQUENCE OF TimedMsClassmark2       OPTIONAL,
    msClassmark3           [12] SEQUENCE OF TimedMsClassmark3       OPTIONAL,
    bsic                   [13] SEQUENCE OF BSIdentityCode          OPTIONAL,
    cic                    [14] SEQUENCE OF CIC                     OPTIONAL,
    handoverResult         [15] SEQUENCE OF TimedHandoverResult     OPTIONAL,
    handoverCause          [16] SEQUENCE OF Cause                   OPTIONAL,
    handoverDuration       [17] SEQUENCE OF TimedHandoverDuration   OPTIONAL,
    targetCellList         [18] SEQUENCE OF TimedTargetCellList     OPTIONAL,
    synchInfo              [19] SEQUENCE OF SynchInfo               OPTIONAL,
    sccpConnectionEvent    [20] SEQUENCE OF TimedTraceSCCPEvent     OPTIONAL,
    bssmapEvent            [21] SEQUENCE OF TimedBSSMAPEvent        OPTIONAL,
    dtapEvent              [22] SEQUENCE OF TimedDTAPEvent          OPTIONAL,
    rrEvent                [23] SEQUENCE OF TimedRREvent            OPTIONAL,
    abisEvent              [24] SEQUENCE OF TimedABISEvent          OPTIONAL,
    timedAbisEvent         [25] SEQUENCE OF TimedABISEvent          OPTIONAL,
    measurementReport      [26] SEQUENCE OF TimedMeasurementEvent   OPTIONAL,
    timedMeasurementReport [27] SEQUENCE OF TimedMeasurementEvent   OPTIONAL,
    powerControlEvent      [28] SEQUENCE OF TimedPowerControlEvent  OPTIONAL,
    timedPowerControlEvent [29] SEQUENCE OF TimedPowerControlEvent  OPTIONAL,
    additionalExtensions   [30] SET OF ManagementExtension          OPTIONAL
}

ABISEvent               ::= OCTET STRING
        -- this type contains an Abis message other than measurement
        -- reports and power control
        -- see GSM 08.58 for encoding.

BSIdentityCode          ::= SEQUENCE
{
        ncc [0] NetworkColourCode,
        bcc             [1] BTSColourCode
}

BSSMAPEvent             ::= OCTET STRING
        -- This type contains a BSSMAP layer 3 message contents,
        -- see GSM 08.08 for encoding

BsTxPower               ::= SEQUENCE
{
        txPower         [0] TxPower,
        changeTime      [1] TimerData
}

BTSColourCode           ::= INTEGER (0..7)

BtsId                   ::= SEQUENCE
{
        relatedBts      [0] ObjectInstance,
        changeTime      [1] TimerData
}

BSCInvokingEvent        ::= OCTET STRING
        -- see GSM 08.08 for encoding

Cause                   ::= SEQUENCE
{
        cause           [0] OCTET STRING,
        changeTime      [1] TimerData
}
        -- see GSM 08.08 for encoding

ChannelDescription      ::= OCTET STRING
        -- see GSM 04.08

ChannelType             ::= OCTET STRING
        -- see GSM 08.08
```

```
CIC                     ::= SEQUENCE
{
        circuitIdentityCode [0] CircuitIdentityCode,
        changeTime          [1] TimerData
}

CircuitIdentityCode     ::= OCTET STRING
        -- see GSM 08.08 for encoding

DTAPEvent               ::= OCTET STRING
        -- This type contains a DTAP layer 3 message contents,
        -- see GSM 04.08 for encoding

EndIndication           ::= SEQUENCE
{
    rrCause [0] RRCause,
    changeTime              [1] TimerData
}

EstablishmentCause      ::= OCTET STRING
        -- see GSM 04.08 for encoding

HandoverResult          ::= ENUMERATED
{
    successful              (0),
    fail                    (1)
}

HandoverDuration        ::= INTEGER
        -- in milliseconds

MeasurementEvent        ::= OCTET STRING
        -- This type contains uplink and downlink measurement
        -- reports,
        -- see GSM 08.58 for encoding

MobileStationClassmark1 ::= OCTET STRING
        -- see GSM 04.08 for encoding

MobileStationClassmark2 ::= OCTET STRING
        -- see GSM 04.08 for encoding

MobileStationClassmark3 ::= OCTET STRING
        -- see GSM 04.08 for encoding

MsTxPower               ::= SEQUENCE
{
    txPower                 [0] TxPower,
    changeTime              [1] TimerData
}

NetworkColourCode       ::= INTEGER (0..7)

PowerControlEvent       ::= OCTET STRING
        -- This type contains power control messages,
        -- see GSM 08.58 for encoding

RadioChannelInfo        ::= SEQUENCE
{
    channelType             [0] ChannelType,
    channelDescription      [1] ChannelDescription,
    changeTime              [2] TimerData
}

RRCause                 ::= OCTET STRING
        -- see GSM 04.08 for encoding

RREvent                 ::= OCTET STRING
        -- see GSM 04.08 for encoding

SynchInfo               ::= SEQUENCE
{
    syncChannelInfo         [0] SynchronisationChannelInformation,
    changeTime              [1] TimerData
}

SynchronisationChannelInformation   ::= OCTET STRING
        -- see GSM 04.08 for encoding

TargetCellList          ::= OCTET STRING
        -- see GSM 08.08 for encoding
```

```
TimedABISEvent           ::= SEQUENCE
{
    abisEvent              [0] ABISEvent,
    changeTime             [1] TimerData
}

TimedBSSMAPEvent         ::= SEQUENCE
{
    bssmapEvent            [0] BSSMAPEvent,
    changeTime             [1] TimerData
}

TimedDTAPEvent           ::= SEQUENCE
{
    dtapEvent              [0] DTAPEvent,
    changeTime             [1] TimerData
}

TimedEstablishmentCause ::= SEQUENCE
{
    establishmentCause     [0] EstablishmentCause,
    changeTime             [1] TimerData
}

TimedHandoverDuration    ::= SEQUENCE
{
    handoverDuration       [0] HandoverDuration,
    changeTime             [1] TimerData
}

TimedHandoverResult      ::= SEQUENCE
{
    handoverResult         [0] HandoverResult,
    changeTime             [1] TimerData
}

TimedMeasurementEvent    ::= SEQUENCE
{
    measurementEvent       [0] MeasurementEvent,
    changeTime             [1] TimerData
}

TimedMsClassmark1        ::= SEQUENCE
{
    mobileStationClassmark1 [0] MobileStationClassmark1,
    changeTime             [1] TimerData
}

TimedMsClassmark2        ::= SEQUENCE
{
    mobileStationClassmark2 [0] MobileStationClassmark2,
    changeTime             [1] TimerData
}

TimedMsClassmark3        ::= SEQUENCE
{
    mobileStationClassmark3 [0] MobileStationClassmark3,
    changeTime             [1] TimerData
}

TimedPowerControlEvent   ::= SEQUENCE
{
    powerControlEvent      [0] PowerControlEvent,
    changeTime             [1] TimerData
}

TimedRREvent             ::= SEQUENCE
{
    rrEvent                [0] RREvent,
    changeTime             [1] TimerData
}

TimedTargetCellList      ::= SEQUENCE
{
    targetCellList         [0] TargetCellList,
    changeTime             [1] TimerData
}

TimedTimingAdvance       ::= SEQUENCE
{
    timingAdvance          [0] TimingAdvance,
    changeTime             [1] TimerData
}
```

```
TimingAdvance             ::= OCTET STRING
        -- see GSM 04.08 for encoding

TraceSCCPEvent            ::= OCTET STRING
        -- This type contains an BSSMAP message,
        -- see GSM 08.06 for encoding

TimedTraceSCCPEvent       ::= SEQUENCE
{
    traceSCCPEvent         [0] TraceSCCPEvent,
    changeTime             [1] TimerData
}

TimerData                 ::= SEQUENCE
{
    timeUnit               [0] TimeUnit,
    timeValue              [1] INTEGER
}

TimeUnit                  ::= ENUMERATED
{
    mSec                   (0),
    sec                    (1),
    min                    (2),
    noOfTDMAFrames         (3),
    noOfSlots              (4),
    factor                 (5)
}

TranscoderID              ::= SEQUENCE
{
    relatedTranscoderID    [0] ObjectInstance,
    changeTime             [1] TimerData
}

TRXID                     ::= SEQUENCE
{
    relatedBasebandTransceiverID    [0] ObjectInstance,
    relatedRadioCarrierID  [1] ObjectInstance,
    changeTime             [2] TimerData
}

TxPower                   ::= INTEGER
```

```
--------------------------------------------------------------------------------
--                    MSC TRACE RECORD CONTENTS
--------------------------------------------------------------------------------


MSCEventRecord          ::= SET
{
    invokingEvent          [0] MSCInvokingEvent                   OPTIONAL,
    servedIMSI             [1] IMSI                               OPTIONAL,
    servedIMEI             [2] IMEI                               OPTIONAL,
    servedMSISDN           [3] ISDN-AddressString                OPTIONAL,
    callingcalledNumber    [4] ISDN-AddressString                OPTIONAL,
    callingSubaddress      [5] ISDN-SubaddressString             OPTIONAL,
    calledSubaddress       [6] ISDN-SubaddressString             OPTIONAL,
    translatedNumber       [7] ISDN-AddressString                OPTIONAL,
    connectedNumber        [8] ISDN-AddressString                OPTIONAL,
    forwardedToNumber      [9] ISDN-AddressString                OPTIONAL,
    forwardedToSubaddress  [10] ISDN-SubaddressString            OPTIONAL,
    redirectingNumber      [11] ISDN-AddressString               OPTIONAL,
    originalCalledNumber   [12] ISDN-AdressString                OPTIONAL,
    roamingNumber          [13] ISDN-AddressString               OPTIONAL,
    networkTKGP            [14] TrunkGroup                        OPTIONAL,
    basicService           [15] BasicServiceCode                 OPTIONAL,
    radioChannelTypes      [16] SEQUENCE OF RadioChanneTypes      OPTIONAL,
    bssHandoverTrunk       [17] SEQUENCE OF BSSTrunkInfo          OPTIONAL,
    mscHandoverTrunk       [18] SEQUENCE OF MSCTrunkInfo          OPTIONAL,
    location               [19] SEQUENCE OF TimedLocation         OPTIONAL,
    ssInformation          [20] SEQUENCE OF SSInformation         OPTIONAL,
    aocParameters          [21] SEQUENCE OF AOCParameters         OPTIONAL,
    msClassmark            [22] SEQUENCE OF TimedMsClassmark2      OPTIONAL,
    callTermDiagnostics    [23] Diagnostics                      OPTIONAL,
    aIntMess               [24] SEQUENCE OF AINTMess              OPTIONAL,
    cIntMess               [25] SEQUENCE OF CINTMess              OPTIONAL,
    dIntMess               [26] SEQUENCE OF DINTMess              OPTIONAL,
    eIntMess               [27] SEQUENCE OF EINTMess              OPTIONAL,
    fIntMess               [28] SEQUENCE OF FINTMess              OPTIONAL,
    gIntMess               [29] SEQUENCE OF GINTMess              OPTIONAL,
    netSigMess             [30] SEQUENCE OF NetSigMess            OPTIONAL,
    eventStartTime         [31] TimerData                        OPTIONAL,
    eventStopTime          [32] TimerData                        OPTIONAL,
    eventNumber            [33] INTEGER,
    recordExtensions       [34] SET OF ManagementExtension       OPTIONAL
}

BSSTrunkInfo            ::= SEQUENCE
{
    changeTime             [0] TimerData,
    bssTrunkInfo           [1] TrunkInfo
}

TimedLocation           ::= SEQUENCE
{
    locationAreaAndCell    [0] LocationAreaAndCell,
    changeTime             [1] TimerData
}

MSCInvokingEvent        ::= ENUMERATED
{
    moc                    (0),
    mtc                    (1),
    ssAction               (2),
    locationUpdate         (3),
    sms-mo                 (4),
    sms-mt                 (5),
    imsiAttach             (6),
    imsiDetach             (7)
}

NetSigMess              ::= SEQUENCE
{
    userPartMess           [0] OCTET STRING,
    changeTime             [1] TimerData
}

MSCTrunkInfo            ::= SEQUENCE
{
    changeTime             [0] TimerData,
    interMSCTrunkInfo      [1] TrunkInfo
}

RadioChannelTypes       ::= SEQUENCE
{
    channelType            [0] ChannelType,
    channelTime            [1] TimerData
}
```

```
SSInformation          ::= SEQUENCE
{
    supplServicesUsed    [1] SS-Code                       OPTIONAL,
    basicServices        [2] BasicServiceCode              OPTIONAL,
    ssAction             [3] SSActionType                  OPTIONAL,
    ssParameters         [4] SSParameters                  OPTIONAL,
    ssActionResult       [5] SSActionResult                OPTIONAL,
    ssInvokeId           [6] INTEGER                       OPTIONAL,
    changeTime           [7] TimerData
}

TrunkInfo              ::= SEQUENCE
{
    trunkGroup           [0] TrunkGroup,
    trunkMember          [1] INTEGER                       OPTIONAL
}

TrunkGroup             ::= CHOICE
{
    tkgpNumber           [0] INTEGER,
    tkgpName             [1] GraphicString,
    tkgpString           [2] IA5STRING (SIZE(1..7))
}

SSActionType           ::= ENUMERATED
{
    registration              (0),
    erasure                   (1),
    activation                (2),
    deactivation              (3),
    interrogation             (4),
    invocation                (5),
    processUnstructuredSS-Data      (6),
    processUnstructuredSS-Request   (7),
    unstructuredSS-Request (8),
    unstructuredNotifySS   (9),
    registerPassword       (10),
    getPassword            (11)
}

SSParameters           ::= CHOICE
{
    registerSS-Arg           [0] RegisterSS-Arg,
    ss-ForBS                 [1] SS-ForBS-Code,
    ss-UserData              [2] SS-UserData,
    ussd-Arg                 [3] USSD-Arg,
    ss-Code                  [4] SS-Code,
    guidanceInfo             [5] GuidanceInfo
}

SSActionResult         ::= CHOICE
{
    ss-Info                  [0] SS-Info,
    interrogateSS-Res        [1] InterrogateSS-Res,
    ss-UserData              [2] SS-UserData,
    ussd-Res                 [3] USSD-Res,
    password                 [4] Password,
    error                    [5] ERROR
}

AINTMess               ::= SEQUENCE
{
    changeTime             TimerData,
    aIntEvent              AINTEvent
}

AINTEvent              ::= CHOICE
{
    bssMapEvent            [0] BSSMAPEvent,
    dtapEvent              [1] DTAPEvent
}

CINTMess               ::= SEQUENCE
{
    changeTime             TimerData,
    cIntMess               OCTET STRING
}

DINTMess               ::= SEQUENCE
{
    changeTime             TimerData,
    dIntMess               OCTET STRING
}
```

```
EINTMess                 ::= SEQUENCE
{
    changeTime               TimerData,
    eIntMess                 OCTET STRING
}

FINTMess                 ::= SEQUENCE
{
    changeTime               TimerData,
    fIntMess                 OCTET STRING
}

GINTMess                 ::= SEQUENCE
{
    changeTime               TimerData,
    gIntMess                 OCTET STRING
}


-----------------------------------------------------------------------------
--                       HLR TRACE RECORD CONTENTS
-----------------------------------------------------------------------------


HLREventRecord           ::= SET
{
    invokingEvent          [0] HLRInvokingEvent                    OPTIONAL,
    servedMSISDN           [2] ISDN-AddressString                  OPTIONAL,
    mscAddress             [3] AddressString                       OPTIONAL,
    vlrNumber              [4] AddressString                       OPTIONAL,
    ssInformation          [5] SEQUENCE OF SSInformation           OPTIONAL,
    subscriberData         [7] SEQUENCE OF SubscriberData          OPTIONAL,
    roamingNumber          [8] ISDN-AddressString                  OPTIONAL,
    smDeliveryOutcome      [9] SEQUENCE OF SM-DeliveryOutcome      OPTIONAL,
    alertReason            [10] SEQUENCE OF AlertReason            OPTIONAL,
    serviceCentreAddress   [11] SEQUENCE OF AddressString          OPTIONAL,
    mapInterfaceMessages   [12] SEQUENCE OF MAPIntMess             OPTIONAL,
    eventStartTime         [13] TimerData                          OPTIONAL,
    eventStopTime          [14] TimerData                          OPTIONAL,
    eventNumber            [15] INTEGER,
    recordExtensions       [16] SET OF ManagementExtension         OPTIONAL
}

HLRInvokingEvent         ::= ENUMERATED
{
    locationChange         (0),
    subscriberDataChange   (1),
    routingEnquiry         (2),
    provideRoamingNumber   (3),
    ssActivity             (4),
    password               (5),
    sms                    (6)
}

MAPIntMess               ::= SEQUENCE
{
    changeTime               TimerData,
    mapIntMess               OCTET STRING
}
```

```
------------------------------------------------------------------------------
--                        TRACE RECORD CONTROL
------------------------------------------------------------------------------

TraceControlId          ::= INTEGER

RecordCriteria          ::= SET OF ENUMERATED
{
    noCriteria              (0),
    eventType               (1)
}

EventTypes              ::= SET OF INTEGER
{
    handOver                (0),
    ss-action               (1),
    sms                     (2),
    setup                   (3),
    release                 (4),
    -- values 5-100 are reserved
    -- values 101-200 are manufacturer specific
    -- values 201-... are reserved
}
TraceFileFormat         ::= SET OF TraceRecord


------------------------------------------------------------------------------
--                        TRACE RECORD OUTPUT
------------------------------------------------------------------------------

RecordCriterionUsed     ::= ENUMERATED
{
    noCriterion             (0),
    event                   (1),
    manufSpecificCriterion  (2),
    deactivation            (3)
}

EventTypeUsed           ::= INTEGER
{
    handOver                (0),
    ss-action               (1),
    sms                     (2),
    setup                   (3),
    release                 (4)

    -- values 5-100 are reserved
    -- values 101-200 are manufacturer specific
    -- values 201-... are reserved
}
```

History

Document history			
January 1996	Public Enquiry	PE 100:	1996-01-22 to 1996-05-17
November 1996	Vote	V 115:	1996-11-25 to 1996-02-21
March 1997	First Edition		
September1997	One-step Approval Procedure (Second Edition)	OAP 9803:	1997-09-19 to 1998-01-16