ETSI TS 132 443 V7.0.0 (2007-06)

Technical Specification

Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Trace Management Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS) (3GPP TS 32.443 version 7.0.0 Release 7)

Reference
DTS/TSGS-0532443v700

Keywords
GSM, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or
perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).
In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive
within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   or greater indicates TSG approved document under change control.

y   the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z   the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

32.441   "Trace Management Integration Reference Point (IRP): Requirements".

32.442   "Trace Management Integration Reference Point (IRP): Information Service (IS)".

32.443   "Trace Management Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

32.445   "Trace Management Integration Reference Point (IRP): eXtensible Markup Language (XML) file format definition".

The present document is part of a TS-family which describes the requirements and information model necessary for the Telecommunication Management (TM) of 3G systems. The TM principles and TM architecture are specified in 3GPP TS 32.101 [2] and 3GPP TS 32.102 [3].

Trace provides very detailed information on call level for a specific subscriber or MS. This data is an additional information source to Performance Measurements and allows deeper investigations in problems solving or in case of optimization.

1 Scope

This Solution Set specification is related to 3GPP TS 32.442 V7.0.X.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1]   3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2]   3GPP TS 32.102: "Telecommunication management; Architecture".

[3]   3GPP TS 32.441: "Telecommunication management; Trace Integration Reference Point (IRP): Requirements".

[4]   3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP): Requirements".

[5]   3GPP TS 32.303: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[6]   3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[7]   3GPP TS 32.442: "Telecommunication management; Trace Management Integration Reference Point (IRP): Information Service (IS)".

[8]   3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management: Information Service (IS)".

[9]   OMG TC Document telecom/98-11-01: "OMG Notification Service".
        http://www.omg.org/technology/documents/

[10]   3GPP TS 32.421: "Telecommunication management; Subscriber and Equipment Trace; Concept and requirements".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.421 [10], 3GPP TS 32.441 [3] and the following apply:

IRP document version number string (or "IRPVersion"): See 3GPP TS 32.311 [4].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CM	Configuration Management
CORBA	Common Object Request Broker Architecture
DN	Distinguished Name
EM	Element Manager
IDL	Interface Definition Language
IS	Information Service
MOC	Managed Object Class
NE	Network Element
OMG	Object Management Group
SS	Solution Set

4 Architectural features

The overall architectural feature of TraceIRP is specified in 3GPP TS 32.441 [3].

This clause specifies features that are specific to the CORBA SS.

4.1 Notifications

Notifications are sent according to the Notification IRP: CORBA SS (see 3GPP TS 32.303 [5]).

The contents of the TraceIRP notifications are defined in the present document.

4.2 Syntax for distinguished names and versions

The format of a Distinguished Name is defined in 3GPP TS 32.300 [6].

The version of this IRP is represented as a string (see also clause 3 for versions).

5 Mapping

5.1 Operation and Notification mapping

TraceIRP: IS 3GPP TS 32.442 [7] defines semantics of operation and notification visible across the TraceIRP.
Table 5.1.1 indicates mapping of these operations and notifications to their equivalents defined in this SS.

Table 5.1.1: Mapping from IS Operations and Notification to SS equivalents

IS Operations/ notification 3GPP TS 32.442 [7]	SS Method	Qualifier
activateTraceJob	activate_trace_job	M
deactivateTraceJob	deactivate_trace_job	M
listTraceJob	list_trace_job	M
listActivatedTraceJob	list_activated_trace_job	M
notifyTraceRecordingSessionFailure	push_structured_events(See subclause 6.1)	O
notifyTraceSessionLocalActivation	push_structured_events (See subclause 6.1)	O

5.2 Operation parameter mapping

The TraceIRP: IS 3GPP TS 32.442 [7] defines semantics of parameters carried in operations across the TraceIRP. The
following tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table 5.2.1: Mapping from IS `activateTraceJob` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
iocInstance	TraceIRPConstDefs::MOClassName mo_class_instance	M
listOfInterfaces	TraceIRPConstDefs::ListOfInterfaces list_of_interfaces	O
listOfNeTypes	TraceIRPConstDefs::ListOfNeTypes list_of_ne_types	M
traceDepth	TraceIRPConstDefs::TraceDepth trace_depth	M
traceReference	TraceIRPConstDefs::TraceReference trace_reference	M
traceTarget	TraceIRPConstDefs::TraceTarget trace_target	M
triggeringEvent	TraceIRPConstDefs:: TriggeringEvent triggering_event	CO
unsupportedList	TraceIRPConstDefs:: UnsupportedList unsupported_list	M
status	Return value of type TraceIRPConstDefs::Result Exception: ActivateTraceJob, InvalidTraceDepth, InvalidTraceTarget, NotUniqueTraceReference ManagedGenericIRPSystem::InvalidParameter, ManagedGenericIRPSystem::ParameterNotSupported	M

Table 5.2.2: Mapping from IS `deactivateTraceJob` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
traceReference	TraceIRPConstDefs:: TraceReference trace_reference	M
traceTarget	TraceIRPConstDefs::TraceTarget trace_target	M
status	Return value of type TraceIRPConstDefs::Result Exception: DeactivateTraceJob, NotUniqueTraceReference	M
traceRecordingSessionReference	TraceIRPConstDefs::TraceRecordingSessionReference trace_recording_session_reference	CM

Table 5.2.3: Mapping from IS `listTraceJob` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
traceReference	TraceIRPConstDefs::TraceReference trace_reference	M
status	Return value of type TraceIRPConstDefs::Result Exception: ListTraceJob, NotUniqueTraceReference	M
iocInstance	TraceIRPConstDefs::MOClassName mo_class_instance	M
listOfInterfaces	TraceIRPConstDefs::ListOfInterfaces list_of_interfaces	O
traceDepth	TraceIRPConstDefs::TraceDepth trace_depth	M
traceRecordingSessionReference	TraceIRPConstDefs::TraceRecordingSessionReference trace_recording_session_reference	CM
traceTarget	TraceIRPConstDefs::TraceTarget trace_target	M
triggeringEvent	TraceIRPConstDefs::TriggeringEvent triggering_event	O

Table 5.2.4: Mapping from IS `listActivatedTraceJobs` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
traceReferenceList	TraceMIRPConstDefs::TraceReferenceList trace_reference_list	M
status	Return value of type TraceIRPConstDefs::Result	M

5.3 Notification parameter mapping

The TraceIRP: IS 3GPP TS 32.442 [7] defines semantics of parameters carried in notifications. The following table indicates the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [9]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [9], is:

```
Header
     Fixed Header
          domain_name
          type_name
          event_name
     Variable Header
Body
     filterable_body_fields
     remaining_body
```

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the TraceIRP: IS 3GPP TS 32.442 [7] defined notification parameters.

Table 5.3.1: Mapping for `notifyTraceRecordingSessionFailure`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See subclause 3.1. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	type_name	M	This is constant string "notifyTraceRecordingSessionFailure".
There is no corresponding IS attribute.	event_name	M	It carries no information.
There is no corresponding IS attribute.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
traceRecordingSessionReference	One NV pair of remaining_body	O	Name of NV pair is the TRACE_RECORDING_SESSION_REFERENCE of TraceIRPNotifications::notifyTraceRecordingSessionFailure. Value of NV pair is TraceRecordingSessionReference of module TraceIRPConstDefs.
traceReference	One NV pair of filterable_body_fields	M	Name of NV pair is the TRACE_REFERENCE of TraceIRPNotifications::notifyTraceRecordingSessionFailure. Value of NV pair is TraceReference of module TraceIRPConstDefs.
reason	One NV pair of remaining_body	O	Name of NV pair is the REASON of traceIRPNotifications:: notifyTraceRecordingSessionFailure. Value of NV pair is a string.

Table 5.3.2: Mapping for `notifyTraceSessionLocalActivation`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See subclause 3.1. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	type_name	M	This is constant string "notifyThresholdMonitorObjectCreation".
There is no corresponding IS attribute.	event_name	M	It carries no information.
There is no corresponding IS attribute.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
notificationId	One NV pair of remaining body	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [5]).
traceReference	One NV pair of filterable_body_fields	M	Name of NV pair is the TRACE_REFERENCE of module TraceIRPNotifications::notifyTraceSessionLocalActivation. Value of NV pair is TraceReference of module TraceIRPConstDefs.
traceTarget	One NV pair of filterable_body_fields	M	Name of NV pair is the TRACE_TARGET of module TraceIRPNotifications::notifyTraceSessionLocalActivation. Value of NV pair is TraceTarget of module TraceIRPConstDefs.
iOCInstance	One NV pair of filterable_body_fields	M	Name of NV pair is the IOC_INSTANCE of module TraceIRPNotifications::notifyTraceSessionLocalActivation. Value of NV pair is MOClassName of module TraceIRPConstDefs.

6 TraceIRPNotification Interface

OMG CORBA Notification push operation is used to realise the notification of `TraceIRP Notifications`. All the notifications in this interface are implemented using this `push_structured_event` method.

6.1 Method `push` (M)

```
module CosNotifyComm {
…
Interface SequencePushConsumer : NotifyPublish {
void push_structured_events(
in CosNotification::EventBatch notifications)
        raises( CosEventComm::Disconnected);
…
}; // SequencePushConsumer
…
}; // CosNotifyComm
```

NOTE 1: The `push_structured_events` method takes an input parameter of type `EventBatch` as defined in the OMG `CosNotification` module (OMG Notification Service [9]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

NOTE 2: The maximum number of events that will be transmitted within a single invocation of this operation is controlled by IRPAgent wide configuration parameter.

NOTE 3: The amount of time the supplier (IRPAgent) of a sequence of Structured Events will accumulate individual events into the sequence before invoking this operation is controlled by IRPAgent wide configuration parameter as well.

NOTE 4: IRPAgent may push `EventBatch` with only one Structured Event.

Annex A (normative):
IDL specifications

A.1   IDL specification (file name "TraceIRPConstDefs.idl")

```
//File: TraceIRPConstDefs.idl
#ifndef _Trace_IRP_CONST_DEFS_IDL_
#define _Trace_IRP_CONST_DEFS_IDL_

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: TraceIRPConstDefs
This module contains commonly used definitions for Trace IRP
==================================================================
*/
module TraceIRPConstDefs
{

enum ResultEnum {OK, FAILURE, PARTIAL_SUCCESS};


typedef unsigned long TraceReference;

typedef sequence<TraceReference> TraceReferenceList;


typedef unsigned long TraceRecordingSessionReference;


typedef octet InterfaceBit;

const unsigned long A_IF = 1;
const unsigned long CAP_IF = 2;
const unsigned long Cx_IF = 3;
const unsigned long Gb_IF = 4;
const unsigned long Gc_IF = 5;
const unsigned long Ge_IF = 6;
const unsigned long Gi_IF = 7;
const unsigned long Gm_IF = 8;
const unsigned long Gmb_IF = 9;
const unsigned long Gn_IF = 10;
const unsigned long Go_IF = 11;
const unsigned long Gr_IF = 12;
const unsigned long Gs_IF = 13;
const unsigned long Iu_CS_IF = 14;
const unsigned long Iu_PS_IF = 15;
const unsigned long Iu_UP_IF = 16;
const unsigned long Iub_IF = 17;
const unsigned long Iur_IF = 18;
const unsigned long MAP_B_IF = 19;
const unsigned long MAP_C_IF = 20;
const unsigned long MAP_D_IF = 21;
const unsigned long MAP_E_IF = 22;
const unsigned long MAP_F_IF = 23;
const unsigned long MAP_G_IF = 24;
const unsigned long MAP_Gc_IF = 25;
const unsigned long MAP_Gd_IF = 26;
const unsigned long MAP_Gf_IF = 27;
const unsigned long MAP_Gr_IF = 28;
const unsigned long Mc_IF = 29;
const unsigned long Mg_IF = 30;
const unsigned long Mi_IF = 31;
const unsigned long Mr_IF = 32;
const unsigned long Mw_IF = 33;
const unsigned long Nb_UP_IF = 34;
const unsigned long Uu_IF = 35;

typedef sequence<InterfaceBit> ListOfInterfaces;
```

```
/*
ListOfInterfacesOptional is a type carrying a conditional parameter.
The boolean shall be TRUE, if the operation request uses this parameter. In this case the value is
present. Otherwise the value is absent.
*/
union ListOfInterfacesOptional switch (boolean)
    {
    case TRUE: ListOfInterfaces value;
    };



enum NeType {MSC_SERVER,MGW,RNC,SGSN,GGSN,BM_SC};

typedef sequence<NeType> ListOfNeTypes;


enum TraceDepth {MINIMUM, MEDIUM, MAXIMUM};

enum TraceTarget {IMSI, IMEI, IMEISV, PUBLIC_ID, PRIVATE_ID, UTRAN_CELL_ID };

typedef unsigned long TriggeringEvent;

const unsigned long EventContext = 1;
const unsigned long GPRSAttach = 2;
const unsigned long GPRSDetach = 3;
const unsigned long Handover = 4;
const unsigned long IMSIAttach = 5;
const unsigned long IMSIDetach = 6;
const unsigned long LocationUpdate = 7;
const unsigned long MBMSContext = 8;
const unsigned long MBMSMulticastServiceActivation = 9;
const unsigned long MobileOriginatedCall = 10;
const unsigned long MobileOriginatedSMS = 11;
const unsigned long MobileTerminatedCall = 12;
const unsigned long MobileTerminatedSMS = 13;
const unsigned long PDPContext = 14;
const unsigned long RoutingAreaUpdate = 15;
const unsigned long SIPInviteMethod = 16;
const unsigned long SIPRegisterMethod = 17;
const unsigned long SIPMessageMethod = 18;
const unsigned long SIPSubscribeMethod = 19;
const unsigned long SIPOtherMethods = 20;
const unsigned long SupplementaryService = 21;

/*
TriggeringEventConditional is a type carrying a conditional parameter.
The boolean shall be TRUE, if the operation the condition is fulfilled and the request uses this
parameter. In this case the value is present. Otherwise the value is absent.
*/
union TriggeringEventConditional switch (boolean)
    {
    case TRUE: TriggeringEvent value;
    };



enum UnsupportedItem {MANAGED_ENTITY, TRACE_DEPTH, LIST_OF_INTERFACES, TRACE_TARGET, REASON};

typedef sequence<UnsupportedItem> UnsupportedList;



/**
* This block identifies attributes which are included as part of the
* notifications defined within TraceIRP. These attribute values should not
* clash with those defined for the attributes of notification
* header (see IDL of Notification IRP).
*/

interface AttributeNameValue
{
    const string TRACE_RECORDING_SESSION_REFERENCE = "TRACE_RECORDING_SESSION_REFERENCE";
    const string TRACE_REFERENCE = "TRACE_REFERENCE";
    const string TRACE_TARGET = "TRACE_TARGET";
    const string MO_INSTANCE = "MO_INSTANCE";
    const string REASON = "REASON";
```

```
};
```

```
};
```

```
};
```

```
};
```

A.2 IDL specification (file name "TraceIRPSystem.idl")

```
//File: TraceIRPSystem.idl
#ifndef _TRACE_IRP_SYSTEM_IDL_
#define _TRACE_IRP_SYSTEM_IDL_

#include <KernelCmConstDefs.idl>
#include <GenericIRPManagementConstDefs.idl>
#include <GenericIRPManagementSystem.idl>#include <TraceIRPConstDefs.idl>

//This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* Module: TraceIRPSystem
This module contains the specification of all operations of Trace IRP Agent.
===============================================================================
*/
module TraceIRP
{
   exception ActivateTraceJob { string reason; };
   exception NotUniqueTraceReference { string reason; };
   exception DeactivateTraceJob { string reason; };
   exception ListTraceJob { string reason; };
   exception ListActivatedTraceJob { string reason; };

  interface TraceIRP
  {

    /**
     * Request to activate a TraceJob through Itf-N.
     **/

    TraceIRPConstDefs::ResultEnum activateTraceJob (
      in KernelCmConstDefs::DN                      moInstance,
      in TraceIRPConstDefs::ListOfInterfacesOptional listOfInterfaces,
      in TraceIRPConstDefs::ListOfNeTypes           listOfNeTypes,
      in TraceIRPConstDefs::TraceDepth              traceDepth,
      in TraceIRPConstDefs::TraceReference          traceReference,
      in TraceIRPConstDefs::TraceTarget             traceTarget,
      in TraceIRPConstDefs::TriggeringEventConditional triggeringEvent,
      out TraceIRPConstDefs:: UnsupportedList        unsupportedList
    )
  raises (ActivateTraceJob,
        GenericIRPManagementSystem::InvalidParameter,
        GenericIRPManagementSystem::ValueNotSupported,
        GenericIRPManagementSystem::OperationNotSupported,
        NotUniqueTraceReference);

    /**
     * Request to deactivate a TraceJob through Itf-N.
     **/

    TraceIRPConstDefs::ResultEnum deactivateTraceJob (
      in TraceIRPConstDefs::TraceReference              traceReference,
      in TraceIRPConstDefs::TraceTarget                 traceTarget,
      out TraceIRPConstDefs::TraceRecordingSessionReference traceRecordingSessionReference)
    raises (DeactivateTraceJob,
          NotUniqueTraceReference,
          GenericIRPManagementSystem::InvalidParameter,
          GenericIRPManagementSystem::ValueNotSupported,
          GenericIRPManagementSystem::OperationNotSupported);

    /**
     * Request to list the parameters of a specific TraceJob through Itf-N.
     **/

    TraceIRPConstDefs::ResultEnum listTraceJob (
      in TraceIRPConstDefs::TraceReference              traceReference,
      out KernelCmConstDefs::DN                         moInstance,
      out TraceIRPConstDefs::ListOfInterfaces           listOfInterfaces,
      out TraceIRPConstDefs::TraceDepth                 traceDepth,
      out TraceIRPConstDefs::TraceRecordingSessionReference traceRecordingSessionReference,
      out TraceIRPConstDefs::TraceTarget                traceTarget,
      out TraceIRPConstDefs::TriggeringEvent            triggeringEvent)
```

```
    raises (ListTraceJob,
            NotUniqueTraceReference,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ValueNotSupported,
            GenericIRPManagementSystem::OperationNotSupported);

    /**
    * Request to list the activated TraceJobs through Itf-N.
    **/

    TraceIRPConstDefs::ResultEnum listActivatedTraceJob (
      out TraceIRPConstDefs::TraceReferenceList          traceReferenceList)
    raises (ListActivatedTraceJob,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ValueNotSupported,
            GenericIRPManagementSystem::OperationNotSupported);
  };

};
#endif // _TRACE_IRP_SYSTEM_IDL_
```

A.3 IDL specification (file name "TraceIRPNotifications.idl")

```
//File: TraceIRPNotifications.idl
#ifndef _TRACE_IRP_NOTIFICATIONS_IDL_
#define _TRACE_IRP_NOTIFICATIONS_IDL_

#include <TraceIRPConstDefs.idl>
#include <NotificationIRPNotifications.idl>

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* Module: TraceIRPNotifications
This module contains the specification of all notifications of Trace IRP Agent.
================================================================================
*/
module TraceIRPNotifications
{

  /**
   * Constant definitions for the notifyTraceRecordingSessionFailure notification
   **/

  interface NotifyTraceRecordingSessionFailure: NotificationIRPNotifications::Notify
  {
    const string EVENT_TYPE = "notifyTraceRecordingSessionFailure";

    /**
     * This constant defines the name of the TraceRecordingSessionReference property.
     * The data type for the value of this property is
     * TraceIRPConstDefs::TraceRecordingSessionReference.
     **/

    const string TRACE_RECORDING_SESSION_REFERENCE =
TraceIRPConstDefs::AttributeNameValue::TRACE_RECORDING_SESSION_REFERENCE;

    /**
     * This constant defines the name of the TraceReference property.
     * The data type for the value of this property is
     * TraceIRPConstDefs::TraceReference.
     **/

    const string TRACE_REFERENCE = TraceIRPConstDefs::AttributeNameValue::TRACE_REFERENCE;


    /**
     * This constant defines the name of the reason property.
     * The data type for the value of this property is string.
     */

    const string REASON = TraceIRPConstDefs::AttributeNameValue::REASON;
  };

  /**
   * Constant definitions for the notifyTraceSessionLocalActivation notification
   **/

  interface NotifyTraceSessionLocalActivation: NotificationIRPNotifications::Notify
  {
    const string EVENT_TYPE = "notifyTraceSessionLocalActivation";

    /**
     * This constant defines the name of the TraceReference property.
     * The data type for the value of this property is
     * TraceIRPConstDefs::TraceReference.
     **/

    const string TRACE_REFERENCE = TraceIRPConstDefs::AttributeNameValue::TRACE_REFERENCE;

    /**
     * This constant defines the name of the TraceTarget property.
     * The data type for the value of this property is
     * TraceIRPConstDefs::TraceTarget.
```

```
    **/

    const string TRACE_TARGET = TraceIRPConstDefs::AttributeNameValue::TRACE_TARGET;


    /**
    * This constant defines the name of the Managed Entity Object Instance property.
    * The data type for the value of this property is string.
    */

    const string MO_INSTANCE = TraceIRPConstDefs::AttributeNameValue::MO_INSTANCE;
    };

};

#endif // _TRACE_IRP_NOTIFICATIONS_IDL_
```

Annex B (informative):
Change history

<table>
<tr><th colspan="9">Change history</th></tr>
<tr><th>Date</th><th>TSG #</th><th>TSG Doc.</th><th>CR</th><th>Rev</th><th>Subject/Comment</th><th>Cat</th><th>Old</th><th>New</th></tr>
<tr><td>Jun 2007</td><td>SA_36</td><td>SP-070289</td><td>--</td><td>--</td><td>Submitted to SA#36 for Approval</td><td>--</td><td>1.0.0</td><td>7.0.0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

History

Document history		
V7.0.0	June 2007	Publication