

# ETSI TS 132 106-4 V3.2.0 (2001-12)

---

*Technical Specification*

**Universal Mobile Telecommunications System (UMTS);  
Telecommunication Management;  
Configuration Management;  
Part 4: Notification Integration Reference Point;  
CMIP solution set version 1:1  
(3GPP TS 32.106-4 version 3.2.0 Release 1999)**

---



---

Reference

RTS/TSGS-0532106-4UR2

---

Keywords

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, send your comment to:

[editor@etsi.fr](mailto:editor@etsi.fr)

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2001.  
All rights reserved.

---

## 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 [www.etsi.org/key](http://www.etsi.org/key).

# Contents

Intellectual Property Rights .....	2
Foreword.....	2
Foreword.....	4
Introduction .....	4
1 Scope .....	6
2 References .....	6
3 Definitions and abbreviations.....	6
3.1 Definitions .....	6
3.2 Abbreviations .....	7
4 Basic aspects .....	7
4.1 Architectural aspects .....	7
4.1.1 Notifications .....	7
4.1.2 Event reporting management .....	7
4.1.3 Subscription related operations.....	8
4.2 Mapping .....	8
4.2.1 Mapping of Operations .....	9
4.2.2 Mapping of operation parameters .....	9
4.2.2.1 Mapping of Parameters of ‘subscribe’ .....	9
4.2.2.2 Mapping of Parameters of ‘unsubscribe’ .....	10
4.2.2.3 Mapping of Parameters of ‘getNotificationIRPVersion’ .....	10
4.2.2.4 Mapping of Parameters of ‘getSubscriptionStatus’ .....	10
4.2.2.5 Mapping of Parameters of ‘changeSubscriptionFilter’ .....	10
4.2.2.6 Mapping of Parameters of ‘suspendSubscription’ .....	10
4.2.2.7 Mapping of Parameters of ‘resumeSubscription’ .....	11
4.2.2.8 Mapping of Parameters of ‘getNotificationCategories’ .....	11
4.2.2.9 Mapping of Parameters of ‘getSubscriptionIds’ .....	11
4.3 Mapping of common notification parameters.....	11
5 GDMO definitions.....	12
5.1 Managed Object Classes .....	12
5.1.1 notificationControl.....	12
5.2 Packages .....	12
5.2.1 notificationControlBasicPackage.....	12
5.2.2 notificationControlInfoPackage.....	12
5.2.3 notificationIRPVersionPackage.....	13
5.3 Actions .....	13
5.3.1 changeSubscriptionFilter (O).....	13
5.3.2 getNotificationCategories (O).....	14
5.3.3 getSubscriptionStatus (O).....	15
5.3.4 getSubscriptionIds (O).....	15
5.3.5 getNotificationIRPVersion (M) .....	16
5.3.6 resumeSubscription (O) .....	17
5.3.7 subscribe (M).....	17
5.3.8 suspendSubscription (O).....	18
5.3.9 unsubscribe (M).....	19
5.4 Attributes.....	20
5.4.1 notificationControlId .....	20
5.4.2 supportedNotificationCategories .....	20
5.4.3 supportedNotificationIRPVersions .....	20
6 ASN.1 definitions.....	21
<b>Annex B (informative): Change history .....</b>	<b>24</b>
History .....	25

---

## Foreword

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

The present document is part 4 of a multi-part TS covering the 3<sup>rd</sup> Generation Partnership Project: Technical Specification Group Services and System Aspects; Telecommunication Management; Configuration Management, as identified below:

- Part 1: “3G Configuration Management: Concept and Requirements”;
- Part 2: “Notification Integration Reference Point: Information Service Version 1”;
- Part 3: “Notification Integration Reference Point: CORBA Solution Set Version 1:1”;
- Part 4: “Notification Integration Reference Point: CMIP Solution Set Version 1:1”;**
- Part 5: “Basic Configuration Management IRP Information Model (including NRM) Version 1”;
- Part 6: “Basic Configuration Management IRP CORBA Solution Set Version 1:1”;
- Part 7: “Basic Configuration Management IRP CMIP Solution Set Version 1:1”;
- Part 8: “Name Convention for Managed Objects”.

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

Configuration Management (CM), in general, provides the operator with the ability to assure correct and effective operation of the 3G network as it evolves. CM actions have the objective to control and monitor the actual configuration on the Network Elements (NEs) and Network Resources (NRs), and they may be initiated by the operator or by functions in the Operations Systems (OSs) or NEs.

CM actions may be requested as part of an implementation programme (e.g. additions and deletions), as part of an optimisation programme (e.g. modifications), and to maintain the overall Quality of Service (QoS). The CM actions are initiated either as a single action on a NE of the 3G network or as part of a complex procedure involving actions on many NEs.

The Itf-N interface for CM is built up by a number of Integration Reference Points (IRPs) and a related Name Convention, which realise the functional capabilities over this interface. The basic structure of the IRPs is defined in 3GPP TS 32.101 [1] and 3GPP TS 32.102 [2]. For CM, a number of IRPs (and the Name Convention) are defined herein, used by this as well as other specifications for Telecom Management (TM) produced by 3GPP. All these are included in 3GPP TS 32.106 from Part 2 and onwards.

The present document is Part 4 of 3GPP TS 32.106 (3GPP TS 32.106-4) - Notification IRP CMIP Solution Set.

**IRP Solution Set version:** The version of this CMIP Solution Set is 1:1, where the first “1” means that it corresponds to the Information Service (3GPP TS 32.106-2 [3]) version 1, and the second “1” means that it is the first CMIP Solution Set corresponding to this Information Service version.

---

# 1 Scope

The present document specifies the Common Management Information Protocol (CMIP) Solution Set (SS) for the Notification Integration Reference Point (IRP): Information Service defined in 3GPP TS 32.106-2 [3]. In detail:

- Clause 4 contains an introduction to some concepts that are the base for some specific aspects of the CMIP interfaces.
- Clause 5 contains the GDMO definitions for the Alarm Management over the CMIP interfaces
- Clause 6 contains the ASN.1 definitions supporting the GDMO definitions provided in clause 5.

---

# 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.

- [1] 3GPP TS 32.101: "3G Telecom Management principles and high level requirements".
- [2] 3GPP TS 32.102: "3G Telecom Management architecture".
- [3] 3GPP TS 32.106-2: "Notification IRP: Information Service".
- [4] 3GPP TS 32.111-2: "Alarm IRP: Information Service".
- [5] ITU-T Recommendation X.710: "Common management information service definition for CCITT applications".
- [6] ITU-T Recommendation X.711: "Common management information protocol specification for CCITT applications".
- [7] ITU-T Recommendation X.721: "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".
- [8] ITU-T Recommendation X.731: "Information technology - Open Systems Interconnection - Systems Management: State management function".
- [9] ITU-T Recommendation X.733: "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".
- [10] ITU-T Recommendation X.734: "Information technology - Open Systems Interconnection - Systems Management: Event report management function".
- [11] 3GPP TS 32.106-1: "3G Configuration Management: Concept and Requirements".

---

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions defined in TS 32.106-1 [11] apply:

## 3.2 Abbreviations

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

ASN.1	Abstract Syntax Notation number 1
CM	Configuration Management
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CMISE	Common Management Information Service Element
EFD	Event Forwarding Discriminator
EM	Element Manager
ETSI	European Telecommunications Standards Institute
GDMO	Guidelines for the Definition of Managed Objects
IRP	Integration Reference Point
ITU-T	International Telecommunication Union – Telecommunications
Itf-N	Interface N (between NM and EM/NE) (3GPP TS 32.102 [2])
MOC	Managed Object Class
MOI	Managed Object Instance
NE	Network Element
NM	Network Manager
NMC	Network Management Centre
OS	Operations System
TMN	Telecommunications Management Network

---

## 4 Basic aspects

The present document defines all the GDMO and ASN.1 definitions necessary to implement the Notification IRP Information Service for the CMIP interface. The definitions provided in the present document are the base to implement any other IRP that includes event reporting and/or management of event reporting.

The terms “manager/agent” are applied in the present document to mean “IRP Manager/IRP Agent” introduced in 3GPP TS 32.106-2 [3].

### 4.1 Architectural aspects

This CMIP Notification IRP is based, as much as possible, on the ITU-T TMN architecture, as defined through the ITU-T X.700 Recommendations series.

#### 4.1.1 Notifications

The Notifications messages are sent from the Agent to the Manager using the CMISE service M-EVENT-REPORT, defined in ITU-T Recommendation X.710 [5] and ITU-T Recommendation X.711 [6].

Any object of the Agent that sends a specific notification to the Manager needs to have, in its Managed Object Class (MOC) Definition, the GDMO definition of that specific “Notification” and the supporting ASN.1 syntax definition. The present document does not define any specific Notification. The specific Notifications are defined in other “CMIP IRP Solution Sets”, as necessary (e.g. the alarm notifications are defined in CMIP Alarm IRP Solution Set).

#### 4.1.2 Event reporting management

In the higher level (protocol independent) description of the Notification IRP Information Service, the event reporting is managed (by the Manager) by means of several operations: subscribe, unsubscribe, suspend, resume subscription, change filter, etc. Most of these operations require the “subscription identifier” parameter to ease the handling of multiple subscriptions.

In the ITU-T TMN architecture the event reporting is managed by means of the MOC Event Forwarding Discriminator (EFD), which is instantiated on the Agent and is controlled by the Manager, by means of CMISE services (M-CREATE, M-SET, etc.). There is no attribute in the EFD that corresponds to the “subscription identifier”.



The mapping between the operations defined in the Notification IRP Information Service and the CMISE services applicable to the EFD is not one-to-one, therefore a mediation function is necessary. This mediation function can be located on the Manager or on the Agent. In the first case, the Manager should translate the subscription-related operations in a sequence of one or more CMISE services, it should assign a subscription identifier and it should handle the mapping between the subscription identifier and the EFDs.

In the second case this mediation is performed by the Agent and is based on the following points:

- A new MOC (i.e. *notificationControl*) is defined to be instantiated on the IRP Agent. This MOC has the purpose to implement the operations defined in Notification IRP Information Service and to interact with the local EFD(s). The operations are implemented as Actions. There is a one-to-one mapping between the operations and the Actions.
- The EFD defined in ITU-T Recommendation X.734 [10] and ITU-T Recommendation X.721 [7] is used for event reporting, however this EFD shall be controlled by the agent. In other words, it shall be created/deleted and its attributes shall be managed by the Agent, via *notificationControl* MOI.
- The Manager shall interact with *notificationControl* MOI located on the IRP Agent to execute the subscription related Actions. It is responsibility of the *notificationControl* MOI to assign the "subscription identifier" and to handle the correspondence between the subscription identifiers, the EFDs and the *discriminatorConstruct* associated to each subscription.  
It is not required that the Manager controls directly the EFD by means of CMISE services.

The second alternative is chosen. The rest of this Solution Set (SS) is based on this choice.

### 4.1.3 Subscription related operations

The operation that allows the Manager to receive notifications from the Agent is *subscribe*.

The IRP concept foresees in different operations a parameter *subscriptionId*, which is generated by the Agent as response to a *subscribe* request and unambiguously identifies a Manager subscription in the scope of the whole Agent. Therefore the Agent is required to maintain at any time a table of correspondence between every subscription and the related EFD instance.

When the forwarding of some notifications is not needed any more, the Manager may invoke an *unsubscribe* operation. In this case one or all subscriptions available for this Manager are cancelled, e.g. the Agent may implicitly delete also the related EFD instance(s).

The creation and deletion of EFD instances on the Manager-Agent interface is therefore "encapsulated", i.e. in the CMIP Solution Set the standardised M-CREATE and M-DELETE services (defined in ITU-T Recommendation X.710 [5] and ITU-T Recommendation X.711 [6]) are not directly used for the EFD management.

Note that only the mandatory EFD attributes (*destination* and *filter*, according to ITU-T Recommendation X.721 [7]) are supported by the *subscribe* operation.

To suspend/resume the forwarding of the notification towards a manager, the subscribed Manager shall use the *suspendSubscription/resumeSubscription* actions of *notificationControl*. These actions result in *locking/unlocking* the administrative state of the related EFD.

To change the filtering constraints associated to a subscription, the subscribed Manager shall use the *changeSubscriptionFilter* action of *notificationControl*. This action results in a change of the *discriminatorConstruct* of the related EFD.

## 4.2 Mapping

The semantics of the Notification IRP is defined in 3GPP TS 32.106-2 [3]. The definitions of the management information defined there are independent of any implementation technology and protocol. This clause maps these protocol independent definitions onto the equivalencies of the CMIP solution set of Notification IRP.

## 4.2.1 Mapping of Operations

Table 1 maps the operations defined in the Notification IRP Information Service onto the equivalent Actions of the CMIP Solution Set. The CMIP Actions are qualified as Mandatory (M) or Optional (O).

The CMIP Actions are based on the M-ACTION service of CMISE, defined in ITU-T Recommendation X.710 [5] and ITU-T Recommendation X.711 [6]).

**Table 1: Mapping of Operations**

Operations of Information Services of the Notification IRP	Equivalent GDMO Actions of the CMIP solution set for the Notification IRP	Qualifier
changeSubscriptionFilter	changeSubscriptionFilter	O
getNotificationCategories	getNotificationCategories	O
getNotificationIRPVersion	getNotificationIRPVersion	M
getSubscriptionIds	getSubscriptionIds	O
getSubscriptionStatus	getSubscriptionStatus	O
resumeSubscription	resumeSubscription	O Implemented if 'suspend-Subscription' is implemented.
subscribe	subscribe	M
suspendSubscription	suspendSubscription	O Implemented if 'resume-Subscription' is implemented.
unsubscribe	unsubscribe	M

## 4.2.2 Mapping of operation parameters

The tables in the following subclauses show the parameters of each operation defined in the Information Service described in TS 32.106-2 and their equivalence in the CMIP solution set of the Notification IRP.

The input parameters of the operations defined in TS 32.106-2 are mapped into "Action information" (see GDMO and ASN.1 definitions for more details).

The output parameters of the operations defined in TS 32.106-2 are mapped into "Action response" (see GDMO and ASN.1 definitions for more details).

### 4.2.2.1 Mapping of Parameters of 'subscribe'

**Table 2: Mapping of Parameters of 'subscribe'**

Operation parameters of the Information Services.	CMIP Solution Set equivalences	Qualifier
managerReference	managerReference	M
timeTick	timeTick	O
notificationCategories	notificationCategoryList	O
filter	filter	O
subscriptionId	subscriptionId	M
status	status	M
no equivalence	destination This information indicates a manager application which is designated to receive the concerned event reports issued by the related agent and is used to create the required EFD in the agent. It can be mapped onto the interface "notify" defined in the Information Service of the Notification IRP.	M

## 4.2.2.2 Mapping of Parameters of 'unsubscribe'

**Table 3: Mapping of Parameters of 'unsubscribe'**

Operation parameters of the Information Services.	CMIP Solution Set equivalencies	Qualifier
managerReference	managerReference	M
subscriptionId	subscriptionId	M
status	status	M

## 4.2.2.3 Mapping of Parameters of 'getNotificationIRPVersion'

**Table 4: Mapping of Parameters of 'getNotificationIRPVersion'**

Operation parameters of the Information Services.	CMIP Solution Set equivalencies	Qualifier
versionNumberList	versionNumberList	M
status	status	M

## 4.2.2.4 Mapping of Parameters of 'getSubscriptionStatus'

**Table 5: Mapping of Parameters of 'getSubscriptionStatus'**

Operation parameters of the Information Services.	CMIP Solution Set equivalencies	Qualifier
subscriptionId	subscriptionId	M
notificationCategoryList	notificationCategoryList	M
filterInEffect	filterInEffect	M
subscriptionStatus	subscriptionStatus	O
timeTick	timeTick	O
status	status	M

## 4.2.2.5 Mapping of Parameters of 'changeSubscriptionFilter'

**Table 6: Mapping of Parameters of 'changeSubscriptionFilter'**

Operation parameters of the Information Services.	CMIP Solution Set equivalencies	Qualifier
subscriptionId	subscriptionId	M
filter	filter	M
status	status	M

## 4.2.2.6 Mapping of Parameters of 'suspendSubscription'

**Table 7: Mapping of Parameters of 'suspendSubscription'**

Operation parameters of the Information Services.	CMIP Solution Set equivalencies	Qualifier
subscriptionId	subscriptionId	M
status	status	M

## 4.2.2.7 Mapping of Parameters of 'resumeSubscription'

Table 8: Mapping of Parameters of 'resumeSubscription'

Operation parameters of the Information Services.	CMIP Solution Set equivalences	Qualifier
subscriptionId	subscriptionId	M
status	status	M

## 4.2.2.8 Mapping of Parameters of 'getNotificationCategories'

Table 9: Mapping of Parameters of 'getNotificationCategories'

Operation parameters of the Information Services.	CMIP Solution Set equivalences	Qualifier
notificationCategoryList	notificationCategoryList	M
extendedEventTypeList	there is no equivalence in the CMIP solution set!	--
status	status	M

## 4.2.2.9 Mapping of Parameters of 'getSubscriptionIds'

Table 10: Mapping of Parameters of 'getSubscriptionIds'

Operation parameters of the Information Services.	CMIP Solution Set equivalences	Qualifier
managerReference	managerReference	M
subscriptionIdList	subscriptionIdList	M
status	status	M

## 4.3 Mapping of common notification parameters

The following table gives the mapping between the common parameters of TS 32.106-2 onto the common parameters of M-EVENT-REPORT

Table 11: Mapping of common notification parameters

Common Parameters	M-EVENT-REPORT Parameters	Qualifier
(see NOTE 1)	Invoke identifier	M
ManagedObjectClass	Managed object class	M
ManagedObjectInstance	Managed object instance	M
NotificationId	(see NOTE 2)	
EventType	Event type (see NOTE 3)	M
ExtendedEventType		
EventTime	Event time	M
SystemDN	(see NOTE 4)	--
NOTE 1: There is no common parameter in IRP Notification that corresponds to Invoke Identifier defined in [5].		
NOTE 2: The common parameter NotificationId is mapped onto notificationIdentifier ([7] [9]) which is not part of the M-EVENT-REPORT header, indeed it is one of the parameters of the event information.		
NOTE 3: The common parameters EventType and ExtendedEventType are condensed in one parameter (Event type) of the M-EVENT-REPORT header.		
NOTE 4: The common parameter SystemDN is conditional in TS 32.106-2 and is not used on the CMIP interfaces.		

## 5 GDMO definitions

### 5.1 Managed Object Classes

#### 5.1.1 notificationControl

```
notificationControl MANAGED OBJECT CLASS
DERIVED FROM
  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
  notificationControlBasicPackage,
  notificationControlInfoPackage,
  notificationIRPVersionPackage;
REGISTERED AS { ts32-106NotificationsObjectClass 1};
```

### 5.2 Packages

#### 5.2.1 notificationControlBasicPackage

```
notificationControlBasicPackage PACKAGE
BEHAVIOUR
  notificationControlBasicPackageBehaviour;
ATTRIBUTES
  notificationControlId GET;
ACTIONS
  changeSubscriptionFilter,
  resumeSubscription,
  subscribe,
  suspendSubscription,
  unsubscribe;
REGISTERED AS { ts32-106NotificationsPackage 1};
```

```
notificationControlBasicPackageBehaviour BEHAVIOUR
```

```
DEFINED AS
```

“The object class *notificationControl* offers all functions defined in the Notification IRP IS enabling managers to subscribe to agents for getting notifications they are concerned. It enables the managers to control the behaviour and to retrieve the management information related to subscriptions

An instance of the 'notificationControl' MOC is identified by the value of the attribute 'notificationControlId'.

The action 'changeSubscriptionFilter' is the means, for the Manager, to change the active filter for the current subscription.

The action 'resumeSubscription' is invoked by the Manager to resume a subscription previously suspended.

The action 'subscribe' is the means, for the Manager, to establish the communication to an Agent in order to receive event reports.

The action 'suspendSubscription' is invoked by the Manager to suspend an active subscription.

The action 'unsubscribe' is invoked by the Manager to cancel one or all subscriptions to the Agent.”;

#### 5.2.2 notificationControlInfoPackage

```
notificationControlInfoPackage PACKAGE
```

**BEHAVIOUR**

notificationControlInfoPackageBehaviour;

**ATTRIBUTES**

supportedNotificationCategories GET;

**ACTIONS**

getNotificationCategories,

getSubscriptionStatus,

getSubscriptionIds;

REGISTERED AS { ts32-106NotificationsPackage 2};

notificationControlInfoPackageBehaviour **BEHAVIOUR****DEFINED AS**

“This package has been defined to allow the Manager to get information about its currently active subscriptions.

The attribute 'supportedNotificationCategories' indicates the categories of notifications supported by the current Agent.

The action 'getNotificationCategories' is the means, for the Manager, to query the supported categories of notifications.

The action 'getSubscriptionStatus' is invoked by the Manager to get information about the status of the specified subscription.

The action 'getSubscriptionIds' allows the Manager to get all currently valid *subscriptionId* values assigned by the Agent to this Manager.”;

### 5.2.3 notificationIRPVersionPackage

notificationIRPVersionPackage **PACKAGE****BEHAVIOUR**

notificationIRPVersionPackageBehaviour;

**ATTRIBUTES**

supportedNotificationIRPVersions GET;

**ACTIONS**

getNotificationIRPVersion;

REGISTERED AS { ts32-106NotificationsPackage 3};

notificationIRPVersionPackageBehaviour **BEHAVIOUR****DEFINED AS**

“This package has been defined to allow the Manager to get information about the Notification IRP versions supported by the Agent.

The attribute 'supportedNotificationIRPVersions' indicates all versions of the NotificationIRP currently supported by the Agent.

The action 'getNotificationIRPVersion' is invoked by the Manager to get information about the NotificationIRP versions supported by the Agent.”;

## 5.3 Actions

### 5.3.1 changeSubscriptionFilter (O)

changeSubscriptionFilter **ACTION****BEHAVIOUR**

changeSubscriptionFilterBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.ChangeSubscriptionFilter;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.ChangeSubscriptionFilterReply;  
**REGISTERED AS** { ts32-106NotificationsAction 1};

changeSubscriptionFilterBehaviour **BEHAVIOUR**

**DEFINED AS**

”A Manager invokes this action to change the active filter for the subscription specified with ‘subscriptionId’ in the request. The Agent will modify in the related EFD instance the value of the attribute *discriminatorConstruct* accordingly.

The ‘Action information’ contains the following data:

- *subscriptionId*

This mandatory parameter identifies unambiguously the Manager subscription.

- *filter*

This mandatory parameter is used to change the value of the attribute *discriminatorConstruct* of the EFD taking into account the additional information:

- Parameter *notificationCategories* (as specified in the *subscribe* action)
- An insertion which discriminates all notifications containing at the beginning of the attribute *additionalText* the string‘(ALIGNMENT’. (see TS 32.111-4 for more details).

The ‘Action response’ is composed of the following data:

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.2 getNotificationCategories (O)

getNotificationCategories **ACTION**

**BEHAVIOUR**

getNotificationCategoriesBehaviour;

**MODE**

CONFIRMED;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.GetNotificationCategoriesReply;

**REGISTERED AS** { ts32-106NotificationsAction 2};

getNotificationCategoriesBehaviour **BEHAVIOUR**

**DEFINED AS**

” A manager may invoke this action to query the categories of notifications supported by a concerned agent. This action is irrelevant to any subscriptions. A manager may invoke this action before or after a subscription.

The ‘Action response’ is composed of the following data:

- *notificationCategoryList*

This parameter identifies a list of categories of notifications supported by the concerned agent. A list containing no element, i.e. a NULL list means that the agent does not support any category of notification.

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.3 getSubscriptionStatus (O)

getSubscriptionStatus **ACTION**

**BEHAVIOUR**

getSubscriptionStatusBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.GetSubscriptionStatus;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule. GetSubscriptionStatusReply;

**REGISTERED AS** { ts32-106NotificationsAction 3};

getSubscriptionStatusBehaviour **BEHAVIOUR**

**DEFINED AS**

”A manager invokes this action to query the status of the current subscription, identified by means of the *subscriptionId* value, returned by the Agent in the *subscribe* operation.

Some subscription status values relate to attributes of the EFD instance created by the manager within the agent, while other parameters refer to properties of the Manager-Agent communication.

The 'Action information' field contains the following data:

- *subscriptionId*

This mandatory parameter identifies unambiguously the Manager subscription.

The response of this action is composed of the following data:

- *notificationCategoryList*

This parameter identifies the categories of notifications supported in the current subscription. If the parameter value is NULL, all notification categories supported by the Agent are emitted towards the Manager.

- *filterInEffect*

This parameter specifies the current *discriminatorConstruct* value of the EFD instance used by the Agent in the communication with the Manager. The value NULL means that no filter constraint applies to the notifications generated by the Agent.

- *subscriptionStatus*

This optional parameter specifies if the current subscription is in the state 'suspended' or not.

- *timeTick*

This optional parameter identifies the value of a timer controlled by the Agent for the supervision of the current subscription. The value is set by the Manager in the *subscribe* operation.

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.4 getSubscriptionIds (O)

getSubscriptionIds **ACTION**

**BEHAVIOUR**

getSubscriptionIdsBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**



TS32-106-NotificationsAsn1TypeModule.GetSubscriptionIds;  
**WITH REPLY SYNTAX**  
 TS32-106-NotificationsAsn1TypeModule.GetSubscriptionIdsReply;  
**REGISTERED AS** { ts32-106NotificationsAction 4};

getSubscriptionIdsBehaviour **BEHAVIOUR**

#### DEFINED AS

“A Manager invokes this action to query all currently valid *subscriptionId* values assigned by Agent to this Manager as result of previous *subscribe* operations triggered by this Manager.

The 'Action information' field contains the following data:

- *managerReference*

This parameter identifies unambiguously the Manager invoking the current operation.

The response of this action is composed of the following data:

- *subscriptionIdList*

This parameter identifies all *subscriptionId* currently valid for the Manager invoking this operation. The value of this parameter is NULL, if the Manager did not yet subscribed to that Agent or the Manager lost all subscription-related information.

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.5 getNotificationIRPVersion (M)

getNotificationIRPVersion **ACTION**  
**BEHAVIOUR**  
 getNotificationIRPVersionBehaviour;  
**MODE**  
 CONFIRMED;  
**WITH REPLY SYNTAX**  
 TS32-106-NotificationsAsn1TypeModule.GetNotificationIRPVersionReply;  
**REGISTERED AS** { ts32-106NotificationsAction 5};

getNotificationIRPVersionBehaviour **BEHAVIOUR**

#### DEFINED AS

“A Manager invokes this action to enquiry about the version of the Notification IRP the concerned Agent supports.

The 'Action information' field contains no data:

The ‘Action response’ is composed of the following data:

- *versionNumbersList*

It contains a list of versions supported by the concerned agent which are backwards compatible. A list containing no element, i.e. a NULL list means that the concerned agent doesn't support any version of the Notification IRP.

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.6 resumeSubscription (O)

resumeSubscription **ACTION**

**BEHAVIOUR**

resumeSubscriptionBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.ResumeSubscription;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.ResumeSubscriptionReply;

**REGISTERED AS** { ts32-106NotificationsAction 6};

resumeSubscriptionBehaviour **BEHAVIOUR**

**DEFINED AS**

”A Manager invokes this action to resume a subscription previously suspended. The Agent will set to 'unlocked' the value of the attribute *administrativeState* of the EFD instance related to the subscription specified in the Manager request. Therefore the forwarding of notifications according to the current filter (*discriminatorConstruct* attribute value) is possible again.

The 'Action information' field contains the following data:

- *subscriptionId*

This mandatory parameter identifies unambiguously the Manager subscription which shall be resumed.

The 'Action response' is composed of the following data:

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.7 subscribe (M)

subscribe **ACTION**

**BEHAVIOUR**

subscribeBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.Subscribe;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.SubscribeReply;

**REGISTERED AS** { ts32-106NotificationsAction 7};

subscribeBehaviour **BEHAVIOUR**

**DEFINED AS**

”A Manager invokes this action to establish a subscription to the Agent for the specified notifications.

In the context of the CMIP Solution Set for Notification IRP, the availability of at least one EFD instance is a necessary pre-requisite for the Manager to receive event reports from the Agent. The *subscribe* action allows the Manager to specify parameters related to the Manager-Agent communication.

After receiving the *subscribe* request, the Agent defines an unambiguous *subscriptionId* value for the current subscription and, if necessary, creates a new EFD instance according to the parameters specified in the action request.

The 'Action information' contains the following data:

- *managerReference*

This parameter identifies unambiguously the Manager invoking the current *subscribe* operation.

- *destination*

This parameter identifies the destination to which event reports that have passed the filter conditions are sent. According to ITU-T X.721, if no destination is specified in the request, then the discriminator is created with the destination defaulted to the AE-Title of the invoker.

- *filter*

This parameter defines the conditions a notification shall fulfil in order to be forwarded to the Manager.

- *timeTick*

This optional parameter identifies the value of a timer controlled by the Agent for the supervision of the current subscription. The timer is reset every time the Manager invokes the *getSubscriptionStatus* action. If the timer expires, the Agent considers the communication with the current Manager as aborted and subsequently releases the resources allocated for this Manager (similar behaviour as in case of an *unsubscribe* action). In order to re-establish the communication, the Manager shall invoke again the *subscribe* action.

- *notificationCategoryList*

This optional parameter identifies one or more types of notifications required in the current subscription. If the parameter value is NULL or absent, the Manager requires that all notification types supported by the Agent shall be emitted.

NOTE: The *discriminatorConstruct* of the EFD is composed taking into account the following information:

- Parameter *filter*
- Parameter *notificationCategories*
- An insertion which discriminates all notifications containing at the beginning of the attribute *additionalText* the string '(ALIGNMENT)'. (see TS 32.111-4 for more details).

The 'Action response' is composed of the following data:

- *subscriptionId*

This parameter identifies unambiguously the current Manager subscription in the scope of the Agent and shall be used later only by the Manager invoking this action.

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.8 suspendSubscription (O)

suspendSubscription ACTION

**BEHAVIOUR**

suspendSubscriptionBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.SuspendSubscription;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.SuspendSubscriptionReply;

**REGISTERED AS** { ts32-106NotificationsAction 8};

suspendSubscriptionBehaviour **BEHAVIOUR**

**DEFINED AS**

”A Manager invokes this action to suspend an active subscription. The Agent will set to 'locked' the value of the attribute *administrativeState* of the EFD instance related to the subscription specified in the Manager request. The forwarding of notifications via the current EFD instance is not possible any more.

The 'Action information' field contains the following data:

- *subscriptionId*

This mandatory parameter identifies unambiguously the Manager subscription which shall be suspended.

The 'Action response' is composed of the following data:

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

### 5.3.9 unsubscribe (M)

unsubscribe **ACTION**

**BEHAVIOUR**

unsubscribeBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-106-NotificationsAsn1TypeModule.Unsubscribe;

**WITH REPLY SYNTAX**

TS32-106-NotificationsAsn1TypeModule.UnsubscribeReply;

**REGISTERED AS** { ts32-106NotificationsAction 9};

unsubscribeBehaviour **BEHAVIOUR**

**DEFINED AS**

”A Manager invokes this action to cancel a subscription to the Agent. For the CMIP solution set this may result in the deletion of the related EFD instance.

The 'Action information' contains the following data:

- *managerReference*

This parameter identifies unambiguously the Manager invoking the current *unsubscribe* operation. In order to cancel a particular subscription, the Manager shall indicate additionally a specific *subscriptionId* value.

- *subscriptionId*

This parameter identifies unambiguously a Manager subscription, established by means of a previous *subscribe* operation. If the parameter value is NULL, all current subscriptions of the Manager identified by means of the *managerReference* are cancelled, i.e. all related EFD instances may be deleted as well.

The 'Action response' is composed of the following data:

- *status*

It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).”;

## 5.4 Attributes

### 5.4.1 notificationControlId

notificationControlId **ATTRIBUTE**  
**WITH ATTRIBUTE SYNTAX**  
TS32-106-NotificationsAsn1TypeModule.GeneralObjectId;  
**MATCHES FOR** EQUALITY;  
**BEHAVIOUR** notificationControlIdBehaviour;  
**REGISTERED AS** { ts32-106NotificationsAttribute 1};

notificationControlIdBehaviour **BEHAVIOUR**

**DEFINED AS**

“This attribute names an instance of a ‘notificationControl’ object class.”;

### 5.4.2 supportedNotificationCategories

supportedNotificationCategories **ATTRIBUTE**  
**WITH ATTRIBUTE SYNTAX**  
TS32-106-NotificationsAsn1TypeModule.NotificationCategory;  
**MATCHES FOR**  
EQUALITY;  
**BEHAVIOUR**  
supportedNotificationCategoriesBehaviour;  
**REGISTERED AS** { ts32-106NotificationsAttribute 2};

supportedNotificationCategoriesBehaviour **BEHAVIOUR**

**DEFINED AS**

”This attribute provides the information concerning the categories of notifications currently supported by the Agent.”;

### 5.4.3 supportedNotificationIRPVersions

supportedNotificationIRPVersions **ATTRIBUTE**  
**WITH ATTRIBUTE SYNTAX**  
TS32-106-NotificationsAsn1TypeModule.SupportedNotificationIRPVersions;  
**MATCHES FOR**  
EQUALITY;  
**BEHAVIOUR**  
supportedNotificationIRPVersionsBehaviour;  
**REGISTERED AS** { ts32-106NotificationsAttribute 3};

supportedNotificationIRPVersionsBehaviour **BEHAVIOUR**

**DEFINED AS**

”This attribute provides the information concerning the NotificationIRP versions currently supported by the Agent.”;

## 6 ASN.1 definitions

TS32-106-NotificationsAsn1TypeModule {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-Operation-Maintenance(3) ts-32-106(106) part4(4) informationModel(0) asn1Module(2) version1(1)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

--EXPORTS everything

IMPORTS

Destination, DiscriminatorConstruct

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

CMISFilter

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

baseNode3gpp OBJECT IDENTIFIER ::= {itu-t (0) identified-organization (4) etsi (0) mobileDomain (0) umts-Operation-Maintenance (3)}

ts32-106Notifications OBJECT IDENTIFIER ::= { baseNode3gpp ts-32-106 (106) part4 (4) informationModel (0)}

ts32-106NotificationsObjectClass OBJECT IDENTIFIER ::= {ts32-106Notifications managedObjectClass (3)}

ts32-106NotificationsPackage OBJECT IDENTIFIER ::= {ts32-106Notifications package (4)}

ts32-106NotificationsAttribute OBJECT IDENTIFIER ::= {ts32-106Notifications attribute (7)}

ts32-106NotificationsAction OBJECT IDENTIFIER ::= {ts32-106Notifications action (9)}

-- Start of 3GPP SA5 own definitions

**ErrorCauses** ::= ENUMERATED

```
{
  noError (0),           -- operation / notification successfully performed
  wrongSubscriptionId (1), -- the value of the parameter subscriptionId is not known for the Agent
  wrongManagedReference (2), -- for the current Manager there is no subscription available
  notificationIRPVersionNotSupported (3), -- Notification IRP version requested by NM not supported by Agent
  wrongFilter (4),       -- the value of the filter parameter is not valid
  wrongDestination (5), -- the value of the destination parameter (subscribe) is not valid
  duplicatedSubscription (6), -- the current Manager already performed a subscription with the same parameters
  wrongTimeTick (7),     -- the value of the timeTick parameter (subscribe) is not valid
  wrongNotificationCategory (8), -- the notification category specified in the subscribe request is unknown
  unspecifiedErrorReason (255) -- operation failed, specific error unknown
}
```

**ChangeSubscriptionFilter** ::= SEQUENCE

```
{
  subscriptionId      GraphicString,
  filter              CMISFilter -- ITU-T X.711
}
```

**ChangeSubscriptionFilterReply** ::= SEQUENCE

```
{
  status              ErrorCauses
}
```

**GeneralObjectId** ::= INTEGER

**GetNotificationCategoriesReply** ::= SEQUENCE

```
{
  notificationCategoryList NotificationCategoryList,
}
```

```

    status                ErrorCauses
  }

```

**GetNotificationIRPVersionReply** ::= SEQUENCE

```

  {
    versionNumbersList    SupportedNotificationIRPVersions,
    status                ErrorCauses
  }

```

**GetSubscriptionStatus** ::= SEQUENCE

```

  {
    subscriptionId        GraphicString
  }

```

**GetSubscriptionStatusReply** ::= SEQUENCE

```

  {
    notificationCategoryList NotificationCategoryList,
    filterInEffect           CMISFilter, -- ITU-T X.711
    subscriptionState        [2] SubscriptionState OPTIONAL,
    timeTick                 [3] INTEGER OPTIONAL,
    status                   [4] ErrorCauses
  }

```

**GetSubscriptionIds** ::= SEQUENCE

```

  {
    managerReference      INTEGER
  }

```

**GetSubscriptionIdsReply** ::= SEQUENCE

```

  {
    subscriptionIdList    SubscriptionIdList,
    status                ErrorCauses
  }

```

IRPVersionNumber ::= GraphicString

**NotificationCategory** ::= ENUMERATED

```

  {
    alarm                (1)--the notification category defined in the alarm IRP
    basicCM              (2)--the notification category defined in the basic CM IRP
  }

```

NotificationCategoryList ::= SET OF NotificationCategory

**ResumeSubscription** ::= SEQUENCE

```

  {
    subscriptionId        GraphicString
  }

```

**ResumeSubscriptionReply** ::= SEQUENCE

```

  {
    status                ErrorCauses
  }

```

**Subscribe** ::= SEQUENCE

```

  {
    managerReference      INTEGER,
    destination           Destination, -- ITU-T X.721
    filter                DiscriminatorConstruct, -- ITU-T X.721
    timeTick              INTEGER OPTIONAL,
    notificationCategoryList NotificationCategoryList OPTIONAL
  }

```

```
}
```

```
SubscribeReply ::= SEQUENCE
```

```
{  
  subscriptionId    GraphicString,  
  status            ErrorCauses  
}
```

```
SubscriptionIdList ::= SET OF GraphicString
```

```
SubscriptionState ::= ENUMERATED
```

```
{  
  suspended        (0),  
  notSuspended     (1)  
}
```

```
SupportedNotificationIRPVersions ::= SET OF IRPVersionNumber
```

```
SuspendSubscription ::= SEQUENCE
```

```
{  
  subscriptionId    GraphicString  
}
```

```
SuspendSubscriptionReply ::= SEQUENCE
```

```
{  
  status            ErrorCauses  
}
```

```
Unsubscribe ::= SEQUENCE
```

```
{  
  managerReference  INTEGER,  
  subscriptionId    GraphicString  
}
```

```
UnsubscribeReply ::= SEQUENCE
```

```
{  
  status            ErrorCauses  
}
```

```
END -- of module TS32-106-NotificationsAsn1TypeModule
```



---

## Annex B (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Mar 2000	S_07	SP-000012	--	--	Approved at TSG SA #7 and placed under Change Control	2.0.0	3.0.0
Mar 2000	--	--	--	--	cosmetic	3.0.0	3.0.1
Jun 2000	S_08	SP-000244	004	--	Split of TS - Part 4: Notification Integration Reference Point (IRP): CMIP Solution Set (SS)	3.0.1	3.1.0
Dec 2001	S_14	SP-010635	001	--	Correction of undefined, conflicting and invalid ASN.1 definitions	3.1.0	3.2.0

---

# History

<b>Document history</b>		
V3.1.0	July 2000	Publication
V3.2.0	December 2001	Publication