

**Source:** SA5 (Telecom Management)  
**Title:** 32.106 CR, "Split of TS - Part 4: Notification Integration Reference Point (IRP): CMIP Solution Set (SS)" (S5-000326)  
**Document for:** Approval  
**Agenda Item:** 6.5.3

---

SA5 has split TS 32.106 Configuration Management (CM) into a multi-part TS as identified below:

Part 1: "3G Configuration Management";  
Part 2: "Notification IRP Information Service";  
Part 3: "Notification IRP CORBA Solution Set";

**Part 4: "Notification IRP CMIP Solution Set";**

Part 5: "Basic Configuration Management IRP Information Model (including NRM)";  
Part 6: "Basic Configuration Management IRP CORBA Solution Set";  
Part 7: "Basic Configuration Management IRP CMIP Solution Set";  
Part 8: "Name Convention for Managed Objects"

Five (5) CRs are submitted to SA#8 for approval; the present one is highlighted in **yellow**:

Spec	CR	Phase	Subject	Cat	Version - Current	Version -New	Doc-2nd-Level
32.106	001	R99	Split of TS - Part 1: Main part of spec - Concept and Requirements	F	3.0.1	3.1.0	S5-000323
32.106	002	R99	Split of TS - Part 2: Notification IRP Information Service (IS)	F	3.0.1	3.1.0	S5-000324
32.106	003	R99	Split of TS - Part 3: Notification IRP CORBA SS	F	3.0.1	3.1.0	S5-000325
32.106	004	R99	Split of TS - Part 4: Notification IRP CMIP SS	F	3.0.1	3.1.0	S5-000326
32.106	005	R99	Split of TS - Part 8: Name Convention for Managed Objects	F	3.0.1	3.1.0	S5-000327

# CHANGE REQUEST

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

**32.106 CR 004**

Current Version: **3.0.1**

GSM (AA.BB) or 3G (AA.BBB) specification number ↑

↑ CR number as allocated by MCC support team

For submission to: **SA#8**  
list expected approval meeting # here ↑

for approval   
for information

strategic  (for SMG use only)  
non-strategic

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

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

**Source:** SA5#12 **Date:** 20 June 2000

**Subject:** Split of TS - Part 4: Notification Integration Reference Point (IRP): CMIP Solution Set (SS)

**Work item:** 32.106 Configuration Management

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

**Reason for change:** This essential part of 32.106, which was missing in version 3.0.1, has now been completed and agreed by SA5 for release 99.

**Clauses affected:** All.

**Other specs affected:** Other 3G core specifications  → List of CRs:  
Other GSM core specifications  → List of CRs:  
MS test specifications  → List of CRs:  
BSS test specifications  → List of CRs:  
O&M specifications  → List of CRs:

**Other comments:**

# 3G TS 32.106-4 V3.0.1d (2000-065)

---

*Technical Specification*

**3rd Generation Partnership Project;  
Technical Specification Group Services and System Aspects;  
Part 4: Notification Integration Reference Point:  
CMIP Solution Set Version 1:1  
(Release 1999)**

---



The present document has been developed within the 3<sup>rd</sup> Generation Partnership Project (3GPP™) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented. This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification. Specifications and reports for implementation of the 3GPP™ system should be obtained via the 3GPP Organisational Partners' Publications Offices.

---

Keywords

---

Fault Management, Alarms

**3GPP**

Postal address

---

3GPP support office address

---

650 Route des Lucioles - Sophia Antipolis  
Valbonne - FRANCE  
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

---

<http://www.3gpp.org>

---

**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.

© 2000, 3GPP Organizational Partners (ARIB, CWTS, ETSI, T1, TTA, TTC).  
All rights reserved.

---

# Contents

Foreword .....	5
Introduction .....	5
1 Scope .....	6
2 References .....	6
3 Definitions and abbreviations .....	6
3.1 Definitions .....	6
3.2 Abbreviations .....	7
4 Basic aspects .....	8
4.1 Architectural aspects .....	8
4.1.1 Notifications .....	8
4.1.2 Event Reporting Management .....	8
4.1.3 Subscription related operations .....	9
4.2 Mapping .....	9
4.2.1 Mapping of Operations .....	9
4.2.2 Mapping of operation parameters .....	10
4.2.2.1 Mapping of Parameters of 'subscribe' .....	10
4.2.2.2 Mapping of Parameters of 'unsubscribe' .....	10
4.2.2.3 Mapping of Parameters of 'getNotificationIRPVersion' .....	11
4.2.2.4 Mapping of Parameters of 'getSubscriptionStatus' .....	11
4.2.2.5 Mapping of Parameters of 'changeSubscriptionFilter' .....	11
4.2.2.6 Mapping of Parameters of 'suspendSubscription' .....	11
4.2.2.7 Mapping of Parameters of 'resumeSubscription' .....	11
4.2.2.8 Mapping of Parameters of 'getNotificationCategories' .....	12
4.2.2.9 Mapping of Parameters of 'getSubscriptionIds' .....	12
4.3 Mapping of common notification parameters .....	12
5 GDMO definitions .....	14
5.1 Managed Object Classes .....	14
5.1.1 notificationControl .....	14
5.2 Packages .....	14
5.2.1 notificationControlBasicPackage .....	14
5.2.2 notificationControlInfoPackage .....	15
5.2.3 notificationIRPVersionPackage .....	15
5.3 Actions .....	16
5.3.1 changeSubscriptionFilter (O) .....	16
5.3.2 getNotificationCategories (O) .....	17
5.3.3 getSubscriptionStatus (O) .....	18
5.3.4 getSubscriptionIds (O) .....	19
5.3.5 getNotificationIRPVersion (M) .....	19
5.3.6 resumeSubscription (O) .....	20
5.3.7 subscribe (M) .....	21
5.3.8 suspendSubscription (O) .....	22
5.3.9 unsubscribe (M) .....	23
5.4 Attributes .....	24
5.4.1 notificationControlId .....	24
5.4.2 supportedNotificationCategories .....	24

6	ASN.1 definitions.....	25
<b>Annex A (informative): Change history .....</b>		<b>313029</b>
	Foreword .....	4
	Introduction .....	4
1	Scope .....	5
2	References .....	5
3	Definitions and abbreviations .....	5
3.1	Definitions .....	5
3.2	Abbreviations .....	6
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 .....	7
4.2	Mapping .....	8
4.2.1	Mapping of Operations .....	8
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' .....	9
4.2.2.3	Mapping of Parameters of 'getNotificationIRPVersion' .....	9
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' .....	10
4.2.2.8	Mapping of Parameters of 'getNotificationCategories' .....	10
4.2.2.9	Mapping of Parameters of 'getSubscriptions' .....	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 .....	13
5.2.3	notificationIRPVersionPackage .....	13
5.3	Actions .....	14
5.3.1	changeSubscriptionFilter (O) .....	14
5.3.2	getNotificationCategories (O) .....	15
5.3.3	getSubscriptionStatus (O) .....	16
5.3.4	getSubscriptions (O) .....	17
5.3.5	getNotificationIRPVersion (M) .....	17
5.3.6	resumeSubscription (O) .....	18
5.3.7	subscribe (M) .....	19
5.3.8	suspendSubscription (O) .....	20
5.3.9	unsubscribe (M) .....	21
5.4	Attributes .....	22
5.4.1	notificationControlId .....	22
5.4.2	supportedNotificationCategories .....	22
6	ASN.1 definitions.....	23
<b>Annex A (informative): Change history .....</b>		<b>27</b>

---

# Foreword

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

~~The present document is part x of multi part 3G TS covering the 3<sup>rd</sup> Generation Partnership Project: Technical Specification Group Services and System Aspects, as identifies below:~~

~~Part 1: “.....”;~~

~~.....~~

~~**Part 4: “Notification Integration Reference Point: CMIP Solution Set”.**~~

~~.....~~

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 multi part technical specification 32.106 of 3GPP describes the rationales and principles for the configuration management of 3G networks. The present document is the 4<sup>th</sup> part of TS 32.106, providing a solution set based on CMIP/CMISE for the Notification Integration Reference Points defined in the 2<sup>nd</sup> part of TS 32.106. 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 NEs and NRs, and they may be initiated by the operator or functions in the 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. The CM actions are initiated either as a single action on a network element of the 3G network or as part of a complex procedure involving actions on many network elements.~~

~~The Itf-N interface for Configuration Management 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 [1] and [2]. For CM, a number of IRPs (and the Name Convention) are defined herein, used by this as well as other technical specifications for telecom management produced by 3GPP. All these documents are included in Parts 2-N of the 3G TS 32.106.~~

~~This document consitutes 32.106 Part 4~~3~~ (32.106-4~~3~~) - Notification IRP ~~CORBA~~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 [35] version 1, and the second “1” means that it is the first CMIP Solution Set corresponding to this Information Service version.~~

---

# 1 Scope

This document (TS 32.106 Part-4) defines the CMIP solution set for the Notification Integration Reference Point Information Service defined in TS 32.106 Part-2 . 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] 3G TS 32.101: "3G Telecom Management principles and high level requirements".

[2] 3G TS 32.102: "3G Telecom Management architecture".

[3] 3G TS 32.106-2: "Notification IRP: Information Service", ~~3G TS 32.106: "3G Configuration Management"~~.

[4] 3G TS 32.111-2: "Alarm IRP: Information Service", ~~3G TS 32.111: "3G Fault Management"~~.

[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".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions defined in TS 32.106 ~~Part-1~~ apply:



## 3.2 Abbreviations

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

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

---

## 4 Basic aspects

This 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 this 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 this document to mean “IRP Manager/IRP Agent” introduced in TS 32.106-2.

### 4.1 Architectural aspects

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

#### 4.1.1 Notifications

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

Any object of the Agent that sends a specific notification to the Manager needs to have, in its Managed Object Class Definition, the GDMO definition of that specific “Notification” and the supporting ASN.1 syntax definition. This 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 [10] and [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 is based on this choice.

### 4.1.3 Subscription related operations

The operation that allows the Manager to receive notifications from the AgentManager 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 a *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 [5] and [6]) are not directly used for the EFD management.

Note that only the mandatory EFD attributes (*destination* and *filter*, according to [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 TS 32.106-2. The definitions of the management information defined there are independent of any implementation technology and protocol. This section maps these protocol independent definitions onto the equivalencies of the CMIP solution set of Notification IRP.

### 4.2.1 Mapping of Operations

The table below 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 or Optional.

The CMIP Actions are based on the M-ACTION service of CMISE, defined in [5] and [6].

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

**Table 1111: Mapping of Operations**

## 4.2.2 Mapping of operation parameters

The tables in the following subsections shows the parameters of each operations 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’

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
managerReference	managerReference	M
timeTick	timeTick	O
notificationCategories	notificationCatagoryList	O
filter	filter	O
subscriptionId	subscriptionId	M
status	status	M
<u>no equivalence</u>	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

**Table 2222: Mapping of Parameters of ‘subscribe’**

### 4.2.2.2 Mapping of Parameters of ‘unsubscribe’

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
managerReference	managerReference	M
subscriptionId	subscriptionId	M
status	status	M

**Table 3333: Mapping of Parameters of 'unsubscribe'**

4.2.2.3 Mapping of Parameters of 'getNotificationIRPVersion'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
versionNumberList	versionNumberList	M
status	status	M

**Table 4444: Mapping of Parameters of 'getNotificationIRPVersion'**

4.2.2.4 Mapping of Parameters of 'getSubscriptionStatus'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
subscriptionId	subscriptionId	M
notificationCategoryList	notificationCategoryList	M
filterInEffect	filterInEffect	M
subscriptionStatus	subscriptionStatus	O
timeTick	timeTick	O
status	status	M

**Table 5555: Mapping of Parameters of 'getSubscriptionStatus'**

4.2.2.5 Mapping of Parameters of 'changeSubscriptionFilter'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
subscriptionId	subscriptionId	M
filter	filter	M
status	status	M

**Table 6666: Mapping of Parameters of 'changeSubscriptionFilter'**

4.2.2.6 Mapping of Parameters of 'suspendSubscription'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
subscriptionId	subscriptionId	M
status	status	M

**Table 7777: Mapping of Parameters of 'suspendSubscription'**

4.2.2.7 Mapping of Parameters of 'resumeSubscription'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
---	---------------------------------------	-----------

subscriptionId	subscriptionId	M
status	status	M

**Table 8888: Mapping of Parameters of 'resumeSubscription'**

#### 4.2.2.8 Mapping of Parameters of 'getNotificationCategories'

Operation parameters of the Information Services.	CMIP <u>Solution Set</u> equivalences	Qualifier
notificationCategoryList	notificationCategoryList	M
extendedEventTypeList	there is no equivalence in the CMIP solution set!	$\neq \emptyset$
status	status	M

**Table 9999: Mapping of Parameters of 'getNotificationCategories'**

#### 4.2.2.9 Mapping of Parameters of 'getSubscriptionIds'

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

**Table 10101010: Mapping of Parameters of 'getSubscriptionIds'**

### 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.

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>	--

**Table 11411111: Mapping of common notification parameters**

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 ~~emipNotification3GPP\_SA5ObjectClass-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 ~~emipNotification3GPP\_SA5Package-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 ~~emipNotification3GPP\_SA5Package-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 emipNotification3GPP\_SA5Package-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**

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.ChangeSubscriptionFilter;

**WITH REPLY SYNTAX**

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.ChangeSubscriptionFilterReply;

**REGISTERED AS** { ts32-106NotificationsAction emipNotification3GPP\_SA5Action-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

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.GetNotificationCategoriesReply;~~

~~REGISTERED AS { ts32-106NotificationsAction cmipNotification3GPP\_SA5Action-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**

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.GetSubscriptionStatus;

#### **WITH REPLY SYNTAX**

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.GetSubscriptionStatusReply;

**REGISTERED AS** { ts32-106NotificationsAction emipNotification3GPP\_SA5Action-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 getSubscriptionsgetSubscriptionIds (O)

getSubscriptionsgetSubscriptionIds ACTION

#### BEHAVIOUR

getSubscriptionsgetSubscriptionIdsBehaviour;

#### MODE

CONFIRMED;

#### WITH INFORMATION SYNTAX

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.GetSubscriptionIdsGetSubscriptions;

#### WITH REPLY SYNTAX

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.GetSubscriptionsGetSubscriptionIdsReply;

**REGISTERED AS** { ts32-106NotificationsAction\_emipNotification3GPP\_SA5Action4};

getSubscriptionsgetSubscriptionIdsBehaviour **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

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.GetNotificationIRPVersionReply;~~

**REGISTERED AS** { ~~ts32-106NotificationsAction\_emipNotification3GPP\_SA5Action-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

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.ResumeSubscription;~~

## WITH REPLY SYNTAX

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.ResumeSubscriptionReply;~~

**REGISTERED AS** { ~~ts32-106NotificationsAction\_emipNotification3GPP\_SA5Action-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

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.Subscribe;

### WITH REPLY SYNTAX

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.SubscribeReply;

**REGISTERED AS** { ts32-106NotificationsAction\_emipNotification3GPP\_SA5Action-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.

The parameter value is indicated in whole minutes. A value between 1 and 15 is considered as 15 minutes, otherwise the indicated value applies. If the specified value is NULL, negative or 0, the Agent set it to infinite, therefore the timer never expires.

- *notificationCategoryListies*

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 *notificatioCategories*
- An insertion which discriminates all notifications containing at the beginning of the attribute *additionaText* 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**

CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.SuspendSubscription;



### WITH REPLY SYNTAX

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.SuspendSubscriptionReply;~~

**REGISTERED AS** { ~~ts32-106NotificationsAction~~ ~~emipNotification3GPP\_SA5Action-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

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.Unsubscribe;~~

### WITH REPLY SYNTAX

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-  
NotificationsAsn1TypeModule.UnsubscribeReply;~~

**REGISTERED AS** { ~~ts32-106NotificationsAction~~ ~~emipNotification3GPP\_SA5Action-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 notificationControllId

notificationControllId **ATTRIBUTE**

**WITH ATTRIBUTE SYNTAX**

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.GeneralObjectId;~~

**MATCHES FOR** EQUALITY;

**BEHAVIOUR** notificationControllIdBehaviour;

**REGISTERED AS** { ~~ts32-106NotificationsAttribute\_emipNotification3GPP\_SA5Attributes-1~~};

notificationControllIdBehaviour **BEHAVIOUR**

**DEFINED AS**

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

### 5.4.2 supportedNotificationCategories

supportedNotificationCategories **ATTRIBUTE**

**WITH ATTRIBUTE SYNTAX**

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModuleTS32-106-NotificationsAsn1TypeModule.NotificationCategoriesNotificationCategory;~~

**MATCHES FOR**

EQUALITY;

**BEHAVIOUR**

supportedNotificationCategoriesBehaviour;

**REGISTERED AS** { ~~ts32-106NotificationsAttribute\_emipNotification3GPP\_SA5Attributes-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**

~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModule~~TS32-106-  
NotificationsAsn1TypeModule.SupportedNotificationIRPVersions;

**MATCHES FOR**

EQUALITY;

**BEHAVIOUR**

supportedNotificationIRPVersionsBehaviour;

**REGISTERED AS** { ~~ts32-106NotificationsAttribute~~ ~~cmipNotification3GPP\_SA5Attributes-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~~~~CMIP\_NOTIFICATION\_3GPP\_SA5TypeModule~~ --  
{ObjectIdentifierValue} to be defined

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 ::= {baseNode (1)} --to be defined
ts32-106Notifications OBJECT IDENTIFIER ::= { baseNode3gpp ts32-106Notif (1)}
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)}
Cmip3GPP-SA5-NotificationObjectClass OBJECT IDENTIFIER ::= {managedObjectClass (3)}
Cmip3GPP-SA5-NotificationAttributes OBJECT IDENTIFIER ::= {attribute (7)}
Cmip3GPP-SA5-NotificationAction OBJECT IDENTIFIER ::= {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,
  notificationCategoryList NotificationCategories,
  status          ErrorCauses
}
```

**GetNotificationIRPVersionReply ::= SEQUENCE**

```
{
  versionNumbersList SupportedNotificationIRPVersions,
  status          ErrorCauses
}
```

**GetSubscriptionStatus ::= SEQUENCE**

```
{
  subscriptionId      GraphicString
}
```

**GetSubscriptionStatusReply ::= SEQUENCE**

```
{
  notificationCategoryList NotificationCategoryListNotificationCategories,
  filterInEffect          CMISFilter, -- ITU-T X.711
  subscriptionState      SubscriptionState OPTIONAL,
  timeTick                INTEGER OPTIONAL,
  status                  ErrorCauses
}
```

**~~GetSubscriptions~~ GetSubscriptionIds ::= SEQUENCE**

```
{
managerReference  INTEGER
}
```

**GetSubscriptionsGetSubscriptionIdsReply** ::= SEQUENCE

```
{
subscriptionIdList  SubscriptionIdList,
status              ErrorCauses
}
```

IRPVersionNumber ::= GraphicString

**NotificationCategory** ::= ENUMERATED ~~—to be completed~~

```
{
alarmNotification          (10), --the notification category defined in the alarm IRP
basicCMstateChangeNotification (24), --the notification category defined in the basic CM IRP
attribValueChangeNotification (2),
relationshipChangeNotification (3),
objectCreationNotification (4),
objectDeletionNotification (5),
alarmListRebuilt (6)
}
```

NotificationCategoryListies ::= 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,
  notificationCategoryListies NotificationCategoryListies 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~~CMP\_NOTIFICATION\_3GPP\_SA5TypeModule~~



## Annex A (informative): Change history

Change history					
TSG SA#	Version	CR	Tdoc SA	New Version	Subject/Comment
S_07	2.0.0	-	SP-000012	3.0.0	Approved at TSG SA #7 and placed under Change Control
<del>Post S5#10S_04S5CM#01</del>	<del>3.0.03-0.3.0-10</del>	<del>-001-</del>	<del>- S5-000227SP-993S5CM-000019, S5CM-0000xx08</del>	<del>3.0.13.1-3.1.00</del>	<del>Updated by MCC staff with editorial changes according to documentation rules. MeehaniUpdated according to S5CM#01. To be approved at TSG SA WG5 #11.sm for data integrity of signalling messages</del>
<del>S_S5#11S_04</del>	<del>3.0.13-0.0</del>	<del>-002</del>	<del>?SP-99308</del>	<del>3.0.1a3.1-0</del>	<del>Updated according to S5#10bis (S5-000192) and S5#11 (decision to create separate parts for main body and earlier annexes). To be agreed at S5#11bis and approved at S5 #12, together with possible new updates according to S5#11bis. Description of layer on which ciphering takes place</del>

SA5 internal Change history				
<u>SA/SA5 meeting</u>	<u>Version</u>	<u>Tdoc SA/SA5</u>	<u>New version</u>	<u>Subject/comment</u>
Post S5#12	3.0.1a	S5C000048, S5C000065	3.0.1c	Updated according to agreements at meeting #12
Post S5#12	3.0.1c	S5C000068	3.0.1d	Updated according to email discussion after meeting #12