ETSI TS 132 304 V5.1.0 (2002-03)

Technical Specification

Universal Mobile Telecommunications System (UMTS); Telecommunication Management; Configuration Management; Notification Integration Reference Point: CMIP Solution Set Version 1:1 (3GPP TS 32.304 version 5.1.0 Release 5)

Reference

RTS/TSGS-0532304Uv5

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

Copyright Notification

Intellectual Property Rights

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

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

Foreword

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

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

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

Contents

Foreword

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

The present document is part the 32.300-series covering the 3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; Telecommunication Management; Notification Management, as identified below:

32.301: “Notification Integration Reference Point: Requirements”;

32.302: “Notification Integration Reference Point: Information Service Version 2”;

32.303: “Notification Integration Reference Point: CORBA Solution Set Version 2:1”;

32.304: “Notification Integration Reference Point: CMIP Solution Set Version 2:1”;

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 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]. The present document is part 4 (3GPP TS 32.304 - Notification IRP CMIP Solution Set.

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 32.302 [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 Notification 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: "3GPP Telecom Management principles and high level requirements".

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

[3] 3GPP TS 32.302: “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 32.600: "3GPP Configuration Management: Concept and High-level Requirements ".

[12] 3GPP TS 32.312: " Generic IRP Management: Information Service".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions defined in TS 32.101 [1], TS 32.102 [2] and TS 32.302 [3] 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
IOC	Information Object Class
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 provides the GDMO and ASN.1 definitions necessary to implement the Notification IRP for the CMIP interface. The definitions provided in the present document are employed by any other IRP that includes event reporting and/or management of event reporting.

4.1 Architectural aspects

The architecture of the Notification IRP CMIP Solution Set is adapted as much as possible to the event reporting management model as defined in ITU-T Rec. X.734 [10].

4.1.1 Event report management function in ITU-T

4.1.1.1 Event report management model

According to the event reporting management model specified in ITU-T Rec. X.734 [10] each managed object may emit notifications (potential event reports). Conceptually, these potential event reports are distributed to all Event Forwarding Discriminators (EFDs) that are instantiated in the IRPAgent. The event forwarding discriminators process the potential event reports to determine which event reports are to be forwarded to a particular destination. The conditions event reports must satisfy in order to be forwarded are specified by the discriminator construct. This is a set of one or more assertions about the presence or value of attributes of the potential event report.

Operational and administrative states are defined for event forwarding discriminators. The operational state has two possible values: enabled and disabled. In the enabled state the discriminator processes the potential event reports. In the disabled state potential event reports are not processed. The administrative states defined are locked and unlocked. When the state is changed from unlocked to locked forwarding of event reports is suspended. When the administrative state is changed from locked to unlocked event forwarding is resumed.

4.1.1.2 Event forwarding discriminator management

The event forwarding discriminator is a managed object. Event reporting is controlled by performing operations on these objects. The required management operations are defined in ITU-T Rec. X.710 [5].

In order to initiate the transmission of event reports an event forwarding discriminator has to be created in the IRPAgent. For this purpose the CMISE M-CREATE service is used. In order to terminate the transmission the discriminator has to be deleted (M-DELETE). The filtering mechanism may be changed by modifying the discriminator construct attribute. This operation is requested by M-SET. The transmission may be suspended and resumed by changing the administrative state from unlocked to locked and vice versa. Also for modifying the administrative state the M-SET service is used.

4.1.1.3 Definition of notifications

ITU-T Rec. X.734 [10] does not define any specific notifications. Instead, any object of the IRPAgent that shall have the capability to emit notifications must have the GDMO and the supporting ASN.1 syntax definition of these notifications included in the definition of its managed object class. More specifically, whereas the present document defines the managed objects and operations for the event reporting function the other IRPs must specify the information to be carried in the notifications.

The event reports are sent from the IRPAgent to the IRPManager using the CMISE service M-EVENT-REPORT, defined in ITU-T Rec. X.710 [5] and ITU-T Rec. X.711 [6].

4.1.2 Mediation between the concepts of Notification IRP IS and ITU-T

The Notification IRP Information Service defines several operations allowing the IRPManager to control the event reporting: subscribe, unsubscribe, suspend subscription, resume subscription, change filter, get subscription status, get subscription identifiers.

The subscription-related operations of the Notification IRP (subscribe, unsubscribe, suspendSubscription, resumeSubscription, changeSubscriptionFilter, getSubscriptionStatus, getSubscriptionIds) are mapped into CMISE services. The remaining operations of the Notification IRP (getNotificationCategories, getNotificationIRPVersion, getOperationProfile, getNotificationProfile) allowing the IRPManager to retrieve information pertaining to the Notification IRP are implemented as GDMO actions by a special managed object in the IRPAgent.

The EFDs are hence directly controlled by the IRPManager. On Itf-N are invoked CMISE services when EFDs are managed and GDMO actions when the IRPManager retrieves information about the Notification IRP.

4.2 Mapping

The semantics of the Notification IRP are defined in 3GPP TS 32.302 [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 the Notification IRP.

4.2.1 Mapping of Information Object Classes (IOC)

Table 1 maps the IOCs defined in the Notification IRP Information Service onto the corresponding Managed Object Classes defined in this CMIP Solution Set. The Managed Object Classes (MOC) are qualified as Mandatory (M) or Optional (O).

Table 1: Mapping of IOC

IOC of the Notification IRP Information Service	MOC or Attributes of the CMIP solution set	Qualifier
NotificationIRP	notificationControl	M
NtfSubscriber	--	
NtfSubscription	--	

4.2.2 Mapping of operations

Table 2 and Table 3 map the operations defined in the 3GPP TS 32.302 [3] (Notification IRP: Information Service) and 3GPP TS 32.312 [12] (Generic IRP Management: Information Service) onto corresponding CMISE services and GDMO actions. The operations are qualified as mandatory (M) or optional (O).

The CMISE services are defined in ITU-T Rec. X.710 [5].

Table 2: Mapping of operations of the Notification IRP IS

Interface	Operation	GDMO Action or CMISE of CMIP SS	Qualifier
NotificationIRPManagement	subscribe	M-CREATE (CMISE) Creation of an EFD	M
	unsubscribe	M-DELETE (CMISE) Deletion of an EFD	M
SubscriptionSuspendOperations	suspendSubscription	M-SET (CMISE) Modification of the administrative state of the EFD to locked	O
	resumeSubscription	M-SET (CMISE) Modification of the administrative state of the EFD to unlocked	O
SubscriptionFilterOperations	changeSubscriptionFilter	M-SET (CMISE) Modification of the discriminator construct in the EFD	O
SubscriptionStatusOperations	getSubscriptionStatus	M-GET (CMISE) Retrieval of EFD attributes	O
SubscriberManagement	getSubscriptionIds	M-GET (CMISE) Retrieval of the object instances of the EFDs having the specified destination attribute	O
IRPManagementOperations	getNotificationCategories	getNotificationCategories	O

Table 3: Mapping of operations of the Generic IRP Management IS

Interface	Operation	GDMO Action of CMIP SS	Qualifier
GenericIRPVersionsOperations	getIRPVersion	getNotificationIRPVersion	M
GenericIRPProfileOperations	getOperationProfile	getOperationProfile	O
	getNotificationProfile	getNotificationProfile	O

4.2.3 Mapping of operation parameters

The tables in the following subclauses show the parameters of each operations defined in the Information Service described in TS 32.302 [3] and their equivalence in this CMIP solution set.

4.2.3.1 Parameter mapping of the operation ‘subscribe’

A manager subscribes to certain notifications by creating an appropriate EFD in the IRPAgent using the CMISE M-CREATE service.

The attribute list parameter of M-CREATE shall contain the values of the EFD attributes for destination and discriminatorConstruct.

The managed object instance of the created EFD is returned to the IRPManager in the M-CREATE success confirmation. According to ITU-T Rec. X.710 [5] this parameter has to be returned, if it is not supplied in the M-CREATE request.

Table 4: Parameter mapping of the operation 'subscribe'

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
managerReference	IN	M-CREATE request parameter 'Attribute list': attribute identifier and value for the EFD 'destination' attribute	M
timeTick	IN	--	--
notificationCategories	IN	M-CREATE request parameter 'Attribute list': attribute identifier and value for the EFD 'discriminatorConstruct' attribute	O
filter	IN	M-CREATE request parameter 'Attribute list': attribute identifier and value for the EFD 'discriminatorConstruct' attribute	O
subscriptionId	OUT	M-CREATE success confirmation parameter 'Managed object instance'	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-CREATE success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-CREATE failure confirmation.
			M

4.2.3.2 Parameter mapping of the operation 'unsubscribe'

The IRPManager can unsubscribe from receiving certain notifications by deleting the associated EFD using the M-DELETE service. The EFD to be deleted is identified by the M-DELETE parameters for the base object class and the base object instance.

The Notification IRP Information Service TS 32.302 [3] specifies that a NtfSubscriber (IRPManager) may only delete subscriptions that are involved in a subscription relationship with the NtfSubsciber identified by the ManagerReference input parameter. This behaviour is mapped to a filtering mechanism in CMIP. The filter must specify an assertion on the EFD attribute 'destination' so that only EFDs whose destination attribute value specifies the IRPManager invoking this operation are selected for deletion.

In TS 32.302 [3] it is also specified that all subscriptions made by the IRPManager specified in the managerReference input parameter shall be deleted when no subscriptionId is provided. This feature is mapped to a scoping and filtering mechanism. Scoped are all EFDs, selected by the filter are only those whose destination attribute specifies the invoking IRPManager.

Table 4: Parameter mapping of the operation 'unsubscribe'

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
managerReference	IN	M-DELETE request parameters 'Scope' and 'Filter' Note: The filter parameter must specify an assertion selecting only EFDs whose destination attribute value specifies the IRPManager identified by managerReference.	M
subscriptionId	IN	M-DELETE request parameters 'Base object class' and 'Base object instance'	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-DELETE success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-DELETE failure confirmation.
			M

4.2.3.3 Parameter mapping of the operation 'getSubscriptionIds'

The IRPManager may retrieve a list of its subscriptions using the M-GET service. For this purpose the M-GET parameter 'Filter' must specify an assertion selecting only EFDs whose destination attribute value specifies the IRPManager identified by managerReference. The object identifiers of the selected EFDs are returned in the M-GET response parameter ‘Managed object instance’. The attributes selected in the M-GET request parameter ‘Attribute identifier list’ and the values returned in the parameter ‘Attribute list’ are of no interest.

Table 5: Parameter mapping of the operation ‘getSubscriptionIds’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
managerReference	IN	M-GET request parameters ‘Base object class’, ‘Base object instance’, ‘Scope’ and ‘Filter’ Note: The filter parameter must specify an assertion selecting only EFDs whose destination attribute value specifies the IRPManager identified by managerReference.	M
subscriptionIdSet	OUT	M-GET response parameter ‘Managed object instance’	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-GET success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-GET failure confirmation.
			M

4.2.3.4 Parameter mapping of the operation ‘getSubscriptionStatus’

The status of an EFD may be retrieved by the IRPManager by reading the attribute values of the EFD. For this purpose the CMIS service M-GET is used.

The emission of certain notifications is suspended when the administrative state of the corresponding EFD is locked. In the unlocked state notifications are forwarded to the IRPManager.

Table 6: Parameter mapping of the operation ‘getSubscriptionStatus’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
subscriptionId	IN	M-GET request parameters ‘Base object class’ and ‘Base object instance’	M
notificationCategoryList	OUT	--	--
filterInEffect	OUT	M-GET response parameter ‘Attribute list’: attribute identifier and value for the EFD ‘discriminatorConstruct’ attribute	M

subscriptionStatus	OUT	M-GET response parameter ‘Attribute list’: attribute identifier and value for the EFD ‘administrativeState’ attribute
		administrativeState
		locked = suspended
		unlocked = not suspended/resumed
			O

timeTick	OUT	--	--

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-GETsuccess confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-GET failure confirmation.
			M

4.2.3.5 Parameter mapping of the operation ‘changeSubscriptionFilter’

The IRPManager may change the conditions to be satisfied by a potential event report before being forwarded by modifying the discriminator construct. The EFD is identified by the M-SET request parameters for the base object class and the base object instance. The new discriminator construct is specified in the M-SET request parameter ‘Modification list’.

Table 7: Parameter mapping of the operation ‘changeSubscriptionFilter’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
subscriptionId	IN	M-SET request parameters ‘Base object class’ and ‘Base object instance’	M
filter	IN	M-SET request parameter ‘Modification list’: attribute identifier and value for the EFD ‘discriminatorConstruct’ attribute	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-SET success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-SET failure confirmation.
			M

4.2.3.6 Parameter mapping of the operation ‘suspendSubscription’

The IRPManager may suspend the transmission of certain notifications by changing the administrative state of the corresponding EFD to locked. The M-SET service is used to request the change of the administrative state. The EFD is identified by the M-SET parameters for the base object class and the base object instance. The attribute to be modified and the new attribute value is specified in the M-SET request parameter ‘Modification list’.

Table 8: Parameter mapping of the operation ‘suspendSubscription’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
subscriptionId	IN	M-SET request parameters ‘Base object class’ and ‘Base object instance’	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-SET success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-SET failure confirmation.
			M

4.2.3.7 Parameter mapping of the operation ‘resumeSubscription’

The IRPManager may resume the emission of certain notifications by changing the administrative state of the corresponding EFD to unlocked. The M-SET service is used to request the change of the administrative state. The EFD is identified by the M-SET request parameters for the base object class and the base object instance. The attribute to be modified and the new attribute value is specified in the M-SET request parameter ‘Modification list’.

Table 9: Parameter mapping of the operation ‘resumeSubscription’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
subscriptionId	IN	M-SET request parameters ‘Base object class’ and ‘Base object instance’	M

status	OUT	status = OperationSucceeded
		The semantics of this status are conveyed by the emission of a M-SET success confirmation.
		status = OperationFailed
		The semantics of this status are conveyed by the emission of a M-SET failure confirmation.
			M

4.2.3.8 Paramter mapping of the operation ‘getNotificationCategories’

Table 10: Parameter mapping of the operation ‘getNotificationCategories’

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
notificationCategoryList	OUT	notificationCategoryList	M
status	OUT	status	M

4.2.3.9 Parameter mapping of the operation 'getIRPVersion'

Table 11: Parameter mapping of the operationParameters of 'getIRPVersion'

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
versionNumberSet	OUT	versionNumberList	M
status	OUT	status	M

4.2.3.10 Parameter mapping of the Operation 'getOperationProfile'

Table 12: Parameter mapping of the operation'getOperationProfile'

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
irpVersion	IN	irpVersionNumber	M
operationNameProfile	OUT	operationNameProfile	M
operationParameterProfile	OUT	operationParameterProfile	M
status	OUT	status	M

4.2.3.11 Parameter mapping of the opeartion 'getNotificationProfile'

Table 13: Parameter mapping of the operation 'getNotificationProfile'

IS Parameter Name	IN/OUT	CMIP SS Equivalent	Qualifier
irpVersion	IN	irpVersionNumber	M
notificationNameProfile	OUT	notificationNameProfile	M
notificationParameterProfile	OUT	notificationParameterProfile	M
status	OUT	status	M

4.2.4 Mapping of the notification header

The following table gives the mapping between the parameters of the notification header specified in TS 32.302 onto the M-EVENT-REPORT request parameters. The notification header contains those parameters that shall be present in every notification.

Table 15: Mapping of common notification parameters

IS Parameters of the Notification Header	M-EVENT-REPORT Request Parameters	Qualifier
(see NOTE 1)	Invoke identifier	M
ManagedObjectClass	Managed object class	M
ManagedObjectInstance	Managed object instance	M
NotificationId	(see NOTE 2)	O
EventTime	Event time	M
SystemDN	(see NOTE 3)	--
NotificationType	Event type	M
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 no explicit M-EVENT-REPORT parameter. Instead, it is included in the M-EVENT-REPORT request parameter 'Event information'. NOTE 3: The common parameter SystemDN is conditional in TS 32.302 [3] 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,
        notificationIRPVersionPackage;
    CONDITIONAL PACKAGES
        notificationControlInfoPackage PRESENT IF "an instance supports it",
        notificationProfilePackage PRESENT IF "an instance supports it";

REGISTERED AS { ts32-304NotificationsObjectClass 1};
```

5.2 Packages

5.2.1 notificationControlBasicPackage

```
notificationControlBasicPackage PACKAGE
    BEHAVIOUR
        notificationControlBasicPackageBehaviour;
    ATTRIBUTES
        notificationControlId;
    REGISTERED AS {ts32-324Package 1};

notificationControlBasicPackageBehaviour BEHAVIOUR
DEFINED AS
```

"An instance of the MOC *notificationControl* is identified by the value of the attribute *notificationControlId*.";

5.2.2 notificationControlInfoPackage

```
notificationControlInfoPackage PACKAGE
    BEHAVIOUR
        notificationControlInfoPackageBehaviour;
    ATTRIBUTES
        supportedNotificationCategories     GET;
    ACTIONS
        getNotificationCategories;
REGISTERED AS { ts32-304NotificationsPackage 1};

notificationControlInfoPackageBehaviour BEHAVIOUR
DEFINED AS
```

“ The attribute *supportedNotificationCategories* indicates the categories of notifications supported by the current IRPAgent. The action *getNotificationCategories* provides the IRPManager with the capability to query the supported categories of notifications.

”;

5.2.3 notificationIRPVersionPackage

notificationIRPVersionPackage **PACKAGE**
    BEHAVIOUR
        notificationIRPVersionPackageBehaviour;
    ATTRIBUTES
        supportedNotificationIRPVersions   GET;
    ACTIONS
        getNotificationIRPVersion;
REGISTERED AS { ts32-304NotificationsPackage 3};

notificationIRPVersionPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to allow the IRPManager to get information about the Notification IRP versions supported by the IRPAgent.

The attribute *supportedNotificationIRPVersions* indicates all versions of the NotificationIRP currently supported by the IRPAgent.

The action *getNotificationIRPVersion* is invoked by the IRPManager to get information about the NotificationIRP versions supported by the IRPAgent.";

5.2.4 notificationProfilePackage

notificationProfilePackage **PACKAGE**
    BEHAVIOUR
        notificationProfilePackageBehaviour;
    ACTIONS
        getOperationProfile,
        getNotificationProfile;
REGISTERED AS { ts32-304NotificationsPackage 4};

notificationProfilePackageBehaviour **BEHAVIOUR**
DEFINED AS
"This package has been defined to allow the IRPManager to get detailed information about the profile of Notification IRP.

The action *getOperationProfile* is invoked by the IRPManager to get detailed information about the operations supported by Notification IRP.

The action *getNotificationProfile* is invoked by the IRPManager to get detailed information about the notifications supported by Notification IRP.";

5.3 Actions

5.3.2 getNotificationCategories (O)

getNotificationCategories **ACTION**
    BEHAVIOUR
        getNotificationCategoriesBehaviour;
    MODE
        CONFIRMED;
    WITH REPLY SYNTAX
        TS32-304TypeModule.GetNotificationCategoriesReply;
REGISTERED AS { ts32-304NotificationsAction 2};

getNotificationCategoriesBehaviour **BEHAVIOUR**
DEFINED AS

An IRPManager may invoke this action to query the categories of notifications supported by a concerned IRPAgent. This action is irrelevant to any subscriptions. An IRPManagermay invoke this action before or after a subscribtion.

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

- *notificationCategoryList*

This parameter identifies a list of categories of notifications supported by the concerned IRPAgent. A list containing no element, i.e. a NULL list means that the IRPAgent 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 getNotificationIRPVersion (M)

```
getNotificationIRPVersion ACTION
    BEHAVIOUR
        getNotificationIRPVersionBehaviour;
    MODE
        CONFIRMED;
    WITH REPLY SYNTAX
        TS32-304TypeModule.GetNotificationIRPVersionReply;
REGISTERED AS { ts32-304NotificationsAction 3};
```

getNotificationIRPVersionBehaviour **BEHAVIOUR**
DEFINED AS

“An IRPManager invokes this action to enquiry about the version of the Notification IRP the concerned IRPAgent 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 IRPAgent which are backwards compatible. A list containing no element, i.e. a NULL list means that the concerned IRPAgent 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).”;

5.3.4 getNotificationProfile (O)

```
getNotificationProfile ACTION
    BEHAVIOUR
        getNotificationProfileBehaviour;
    MODE
        CONFIRMED;
    WITH INFORMATION SYNTAX
        TS32-304TypeModule.IRPVersionNumber;
    WITH REPLY SYNTAX
        TS32-304TypeModule.GetNotificationProfileReply;
REGISTERED AS { ts32-304NotificationsAction 4};
```

getNotificationProfileBehaviour **BEHAVIOUR**
DEFINED AS

“An IRPManager invokes this action to enquiry about the notification profile (supported notifications and supported parameters) for this specific Notification IRP version.

The 'Action information' contains the following data:

- *irpVersionNumber*

This mandatory parameter identifies a Notification IRP version.

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

- *notificationNameProfile*

It contains a list of notification names, i.e. a NULL list means that the Notification IRP doesn't support any notification.

- *notificationParameterProfile.*

  It contains a set of elements, each element corresponds to a notification name and is composed by a set of parameter names.

- *status*

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

5.3.5 getOperationProfile (O)

```
getOperationProfile ACTION
    BEHAVIOUR
        getOperationProfileBehaviour;
    MODE
        CONFIRMED;
    WITH INFORMATION SYNTAX
        TS32-304TypeModule.IRPVersionNumber;
    WITH REPLY SYNTAX
        TS32-304TypeModule.GetOperationProfileReply;
REGISTERED AS { ts32-304NotificationsAction 5};
```

getOperationProfileBehaviour **BEHAVIOUR**
DEFINED AS

"An IRPManager invokes this action to enquiry about the operation profile (supported operations and supported parameters) for this specific Notification IRP version.

The 'Action information' contains the following data:

- *irpVersionNumber*

This mandatory parameter identifies a Notification IRP version.

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

- *operationNameProfile*

It contains a list of operation names.

- *operationParameterProfile.*

It contains a set of elements, each element corresponds to an operation name and is composed by a set of parameter names.

- *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
        TS32-304TypeModule.GeneralObjectId;
    MATCHES FOR    EQUALITY;
    BEHAVIOUR    notificationControlIdBehaviour;
REGISTERED AS { ts32-304NotificationsAttribute 1};
```

```
notificationControlIdBehaviour BEHAVIOUR
DEFINED AS
```

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

5.4.2 supportedNotificationCategories

```
supportedNotificationCategories ATTRIBUTE
    WITH ATTRIBUTE SYNTAX
        TS32-304TypeModule. NotificationCategoryList;
    MATCHES FOR
        EQUALITY;
    BEHAVIOUR
        supportedNotificationCategoriesBehaviour;
REGISTERED AS { ts32-304NotificationsAttribute 2};
```

```
supportedNotificationCategoriesBehaviour BEHAVIOUR
DEFINED AS
```

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

5.4.3 supportedNotificationIRPVersions

```
supportedNotificationIRPVersions ATTRIBUTE
    WITH ATTRIBUTE SYNTAX
        TS32-304TypeModule.SupportedNotificationIRPVersions;
    MATCHES FOR
        EQUALITY;
    BEHAVIOUR

        supportedNotificationIRPVersionsBehaviour;
REGISTERED AS { ts32-304NotificationsAttribute 3};
```

```
supportedNotificationIRPVersionsBehaviour BEHAVIOUR
DEFINED AS
```

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

6 ASN.1 definitions

```
TS32-304TypeModule {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-Operation-Maintenance(3)

            ts-32-304(304) 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)};

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

ts32-304Prefix          OBJECT IDENTIFIER ::= { baseNodeUMTS ts-32-304(304)}
ts32-304InfoModel       OBJECT IDENTIFIER ::= { ts32-304Prefix informationModel(0)}

ts32-304NotificationsObjectClass  OBJECT IDENTIFIER ::= { ts32-304InfoModel managedObjectClass(3)}
ts32-304NotificationsPackage      OBJECT IDENTIFIER ::= { ts32-304InfoModel package(4)}
ts32-304NotificationsAttribute    OBJECT IDENTIFIER ::= { ts32-304InfoModel attribute(7)}
ts32-304NotificationsAction       OBJECT IDENTIFIER ::= { ts32-304InfoModel action(9)}



    -- Start of 3GPP SA5 own definitions

    ErrorCauses ::= ENUMERATED
        {
        noError (0),                        -- operation / notification successfully performed
        notificationIRPVersionNotSupported (3),        -- Notification IRP version requested by NM not supported by
                                            IRPAgent
        wrongFilter (4),                    -- the value of the filter parameter is not valid
        wrongDestination (5),               -- the value of the destination parameter is not valid
        unspecifiedErrorReason (255)        -- operation failed, specific error unknown
        }


    GeneralObjectId ::= INTEGER

    GetNotificationCategoriesReply ::= SEQUENCE
        {
        notificationCategoryList     NotificationCategoryList,
        status                       ErrorCauses
        }

    GetNotificationIRPVersionReply ::= SEQUENCE
        {
        versionNumbersList     SupportedNotificationIRPVersions,
        status                 ErrorCauses
        }

    GetNotificationProfileReply ::= SEQUENCE
        {
        notificationNameProfile        NotificationList,
        notificationParameterProfile   ParameterListOfList,
        status                         ErrorCauses
        }

    GetOperationProfileReply ::= SEQUENCE
        {
        operationNameProfile           OperationList,
        operationParameterProfile      ParameterListOfList,
        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
        bulkCM              (3) --the notification category defined in the bulk CM IRP
        }

    NotificationCategoryList ::= SET OF NotificationCategory

    NotificationList ::= SET OF NotificationName

    NotificationName ::= GraphicString

    OperationList ::= SET OF OperationName

    OperationName ::= GraphicString

    ParameterList ::= SET OF ParameterName

    ParameterListOfList ::= SET OF ParameterList

    ParameterName ::= GraphicString

    SupportedNotificationIRPVersions ::= SET OF IRPVersionNumber

    END -- of TS32-304TypeModule
```

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2001	S_12	SP-010283	--	--	Approved at TSG SA #12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	S_13	SP-010471	001	--	SupportedNotificationCategory syntax	4.0.0	4.1.0
Sep 2001	S_13	SP-010471	002	--	Introduction of conditional packages	4.0.0	4.1.0
Sep 2001	S_13	SP-010471	003	--	OID modified according to TS 32.304 new number	4.0.0	4.1.0
Dec 2001	S_14	SP-010653	004	--	Maximize the reuse of ITU-T CMIP event report management functions	4.1.0	5.0.0
Mar 2002	S_15	SP-020031	006	--	Correction of errors in the GDMO and ASN.1 definitions	5.0.0	5.1.0

History

Document history		
V5.1.0	March 2002	Publication