Technical Specification Group Services and System Aspects
Meeting #15, Cheju Island, Korea, 11-14 March 2002

**TSGS#15(02)0031**

**Source:** **SA5 (Telecom Management)**

**Title:** **Rel-5 CR 32.304 (Configuration Management; Notification Integration Reference Point: CMIP SS) Correction of errors in the GDMO and ASN.1 definitions**

**Document for:** **Decision**

**Agenda Item:** **7.5.3**

| Doc-1st-Level | Spec | CR | R | Phase | Subject | Cat | Version-Current | Version-New | Doc-2nd-Level | Workitem |
|---|---|---|---|---|---|---|---|---|---|---|
| SP-020031 | 32.304 | 006 | | Rel-5 | **Correction of errors in the GDMO and ASN.1 definitions** | F | 5.0.0 | 5.1.0 | S5-020157 | OAM-CM |

**3GPP TSG-SA5 (Telecom Management)** **S5-020157**
**Meeting #26, Miami, USA, 25 February – 1 March 2002** **S5C020126r3**

*CR-Form-v5*

# CHANGE REQUEST

⌘ **32.304** **CR** **006** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

***Proposed change affects:*** ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network **X** Core Network **X**

| | | |
|---|---|---|
| ***Title:*** | ⌘ | Correction of errors in the GDMO and ASN.1 definitions |
| ***Source:*** | ⌘ | SA5 |
| ***Work item code:*** | ⌘ | OAM-NIM |
| ***Date:*** | ⌘ | 01/03/2002 |
| ***Category:*** | ⌘ | **F** |
| ***Release:*** | ⌘ | REL-5 |

*Use one of the following categories:*
***F*** *(correction)*
***A*** *(corresponds to a correction in an earlier release)*
***B*** *(addition of feature),*
***C*** *(functional modification of feature)*
***D*** *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

*Use one of the following releases:*
*2 (GSM Phase 2)*
*R96 (Release 1996)*
*R97 (Release 1997)*
*R98 (Release 1998)*
*R99 (Release 1999)*
*REL-4 (Release 4)*
*REL-5 (Release 5)*

| | | |
|---|---|---|
| ***Reason for change:*** | ⌘ | The current GDMO and ASN.1 definitions contain some errors. |
| ***Summary of change:*** | ⌘ | The errors in the GDMO and ASN.1 definitions are removed. |
| ***Consequences if not approved:*** | ⌘ | The GDMO and ASN.1 definitions cannot be compiled. |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 4, 5, 6 |
| ***Other specs affected:*** | ⌘ | ☐ Other core specifications ⌘<br>☐ Test specifications<br>☐ O&M Specifications |
| ***Other comments:*** | ⌘ | None |

**How to create CRs using this form:**
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

# 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 shall be 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 the 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 informations 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)<br>Creation of an EFD | M |
| | unsubscribe | M-DELETE (CMISE)<br>Deletion of an EFD | M |
| SubscriptionSuspendOperations | suspendSubscription | M-SET (CMISE)<br>Modification of the administrative state of the EFD to locked | O |
| | resumeSubscription | M-SET (CMISE)<br>Modification of the administrative state of the EFD to unlocked | O |
| SubscriptionFilterOperations | changeSubscriptionFilter | M-SET (CMISE)<br>Modification of the discriminator construct in the EFD | O |
| SubscriptionStatusOperations | getSubscriptionStatus | M-GET (CMISE)<br>Retrieval of EFD attributes | O |

| | | | |
|---|---|---|---|
| SubscriberManagement | getSubscriptionIds | M-GET (CMISE)<br>Retrieval of the object instances of the EFDs having the specified destination attribute | O |
| IRPManagementOperations | getNotificationCategories | g~~G~~etNotificationCategories | 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 and their equivalence in this CMIP solution set.

~~The input parameters of the operations defined in TS 32.302 are mapped into "Action information" (see GDMO and ASN.1 definitions for more details).~~

~~The output parameters of the operations defined in TS 32.302 are mapped into "Action response" (see GDMO and ASN.1 definitions for more details).~~

### 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<br>The semantics of this status are conveyed by the emission of a M-CREATE success confirmation.<br><br>status = OperationFailed<br>The semantics of this status are conveyed by the emission of | M |

| | | | |
|---|---|---|---|
| | | a M-CREATE failure confirmation. | |

### 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 [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 [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'<br>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<br>The semantics of this status are conveyed by the emission of a M-DELETE success confirmation.<br><br>status = OperationFailed<br>The semantics of this status are conveyed by the emission of a M-DELETE failure confirmation. | M |

### 4.2.3.3 Paramter mapping of the 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'<br>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<br>The semantics of this status are conveyed by the emission of a M-GET success confirmation.<br><br>status = OperationFailed<br>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<br><br>administrativeState<br>locked = suspended<br>unlocked = not suspended/resumed | O |
| timeTick | OUT | -- | -- |
| status | OUT | status = OperationSucceeded<br>The semantics of this status are conveyed by the emission of a M-GETsuccess confirmation.<br><br>status = OperationFailed<br>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<br>The semantics of this status are conveyed by the emission of a M-SET success confirmation.<br><br>status = OperationFailed<br>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<br>The semantics of this status are conveyed by the emission of a M-SET success confirmation.<br><br>status = OperationFailed<br>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<br>The semantics of this status are conveyed by the emission of a M-SET success confirmation.<br><br>status = OperationFailed<br>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 ~~M~~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 ~~common notification parameters~~

The following table gives the mapping between the parameters of the notification header specified in~~common information parameters of~~ TS 32.302 onto the ~~common parameters of~~ 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**

| ~~Common~~ 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].<br>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'.<br>NOTE 3: The common parameter SystemDN is conditional in TS 32.302 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**
~~notificationControlId GET;~~
supportedNotificationCategories GET;
**ACTIONS**
getNotificationCategories;
**REGISTERED AS** { ts32-304NotificationsPackage 1};

notificationControlInfoPackageBehaviour **BEHAVIOUR**
**DEFINED AS**
" ~~This package has been defined to allow the IRPManager to get information about its currently active subscriptions.~~

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**
TS32-304-4TypeModule.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**
TS32-304-4TypeModule.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**
TS32-304-4TypeModule.IRPVersionNumber;
**WITH REPLY SYNTAX**
TS32-304-4TypeModule.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**
TS32-304-4TypeModule.IRPVersionNumber;
**WITH REPLY SYNTAX**
TS32-304-4TypeModule.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
```