Technical Specification Group Services and System Aspects **_TSGS#13(01)0471_**

Meeting #13, Beijing, China, 24-27 September 2001

**Source:** **SA5**

**Title:** **Rel-4 CR32.304 (Telecommunication Management; Notification Integration Reference Point: CMIP Solution Set)**

**Document for:** **Approval**

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

| Doc-1st-Level | Doc-2nd-Level | Spec | CR | Rev | Phase | Subject | Cat | Version Current | Version -New | Workitem |
|---|---|---|---|---|---|---|---|---|---|---|
| SP-010471 | S5F010128 | 32.304 | 001 | | Rel-4 | SupportedNotificationCategory syntax | F | 4.0.0 | 4.1.0 | OAM-CM |
| SP-010471 | S5F010129 | 32.304 | 002 | | Rel-4 | Introduction of conditional packages | F | 4.0.0 | 4.1.0 | OAM-CM |
| SP-010471 | S5F010127 | 32.304 | 003 | | Rel-4 | OID modified according to TS 32.304 new number | F | 4.0.0 | 4.1.0 | OAM-CM |

**3GPP TSG-SA5 (Telecom Management)**
**Meeting #21, Naperville, IL, USA 16 – 20 July 2001**

**S5F010128**

*CR-Form-v4*

# CHANGE REQUEST

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ⌘ | **32.304** | **CR** | **001** | ⌘ | ev | **-** | ⌘ | Current version: | **4.0.0** ⌘ |

*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 supportedNotificationCategories Syntax | | | |
| ***Source:*** ⌘ | SA5 | | | |
| ***Work item code:*** ⌘ | OAM-FM | | ***Date:*** ⌘ | 20/07/2001 |
| ***Category:*** ⌘ | **F** | | ***Release:*** ⌘ | REL-4 |
| | *Use one of the following categories:*<br>***F*** *(correction)*<br>***A*** *(corresponds to a correction in an earlier release)*<br>***B*** *(addition of feature),*<br>***C*** *(functional modification of feature)*<br>***D*** *(editorial modification)*<br>Detailed explanations of the above categories can be found in 3GPP TR 21.900. | | *Use one of the following releases:*<br>*2 (GSM Phase 2)*<br>*R96 (Release 1996)*<br>*R97 (Release 1997)*<br>*R98 (Release 1998)*<br>*R99 (Release 1999)*<br>*REL-4 (Release 4)*<br>*REL-5 (Release 5)* | |

| | |
|---|---|
| ***Reason for change:*** ⌘ | The ASN1 syntax of supportedNotificationCategories is wrong. According its behaviour, this attribute should be a set of Notificaction Categories and not a single Notification Category. |
| ***Summary of change:*** ⌘ | Change the ASN1 Syntax. |
| ***Consequences if not approved:*** ⌘ | TS 32.304 not consistent with itself and with TS 32.302. |

| | | |
|---|---|---|
| ***Clauses affected:*** ⌘ | 5.4.2 | |
| ***Other specs affected:*** ⌘ | □ Other core specifications ⌘<br>□ Test specifications<br>□ O&M Specifications | |
| ***Other comments:*** ⌘ | | |

**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.

## 5.4.2 supportedNotificationCategories

```
supportedNotificationCategories ATTRIBUTE
    WITH ATTRIBUTE SYNTAX
        TS32-301-4TypeModule.~~NotificationCategory~~NotificationCategoryList;
    MATCHES FOR
        EQUALITY;
    BEHAVIOUR
        supportedNotificationCategoriesBehaviour;
REGISTERED AS { ts32-301NotificationsAttribute 2};
```

supportedNotificationCategoriesBehaviour **BEHAVIOUR**

**DEFINED AS**

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

**3GPP TSG-SA5 (Telecom Management)**
**Meeting #21, Naperville, IL, USA 16 – 20 July 2001**

**S5F010129**

*CR-Form-v4*

# CHANGE REQUEST

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ⌘ | **32.304** | **CR** | **002** | ⌘ | ev | **-** | ⌘ | Current version: | **4.0.0** ⌘ |

*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 ☒ Core Network ☒

| | | |
|---|---|---|
| ***Title:*** | ⌘ | Introduction of Conditional Packages to model “Optional Operations”. |
| ***Source:*** | ⌘ | SA5 |
| ***Work item code:*** | ⌘ | OAM-FM |
| ***Date:*** | ⌘ | 20/07/2001 |
| ***Category:*** | ⌘ | **F** |
| ***Release:*** | ⌘ | REL-4 |

*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 Operations that are qualified “Optional” in TS 32.111-2 are currently modelled as “Mandatory”. |
| ***Summary of change:*** | ⌘ | It is proposed to model them as “Optional” by means of Conditional Packages. |
| ***Consequences if not approved:*** | ⌘ | Inconsistency between TS 32.302 and TS 32.304. |

| | | | |
|---|---|---|---|
| ***Clauses affected:*** | ⌘ | 5.1.1 and 5.2.x | |
| ***Other specs affected:*** | ⌘ | □ Other core specifications<br>□ Test specifications<br>□ O&M Specifications | ⌘ |
| ***Other comments:*** | ⌘ | | |

### 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.

## 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,~~
~~notificationProfilePackage,~~
notificationIRPVersionPackage;
**CONDITIONAL PACKAGES**
notificationControlInfoPackage PRESENT IF "an instance supports it",
notificationProfilePackage PRESENT IF "an instance supports it",
notificationSubscriptionFilterPackage PRESENT IF "an instance supports it",
notificationSubscriptionControlPackage PRESENT IF "an instance supports it ";

**REGISTERED AS** { ts32-301NotificationsObjectClass 1};

## 5.2 Packages

### 5.2.1 notificationControlBasicPackage

notificationControlBasicPackage PACKAGE
BEHAVIOUR
notificationControlBasicPackageBehaviour;
ATTRIBUTES
notificationControlId GET;
ACTIONS
~~changeSubscriptionFilter,~~
~~resumeSubscription,~~
subscribe,
~~suspendSubscription,~~
unsubscribe;
REGISTERED AS { ts32-301NotificationsPackage 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~~provides the Manager with the capability~~,~~ 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 { ts32-301NotificationsPackage 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~~provides the Manager with the capability ~~,~~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 { ts32-301NotificationsPackage 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.”;

## 5.2.4 notificationProfilePackage

```
notificationProfilePackage PACKAGE
    BEHAVIOUR
        notificationProfilePackageBehaviour;
    ACTIONS
        getOperationProfile,
        getNotificationProfile;
```

REGISTERED AS { ts32-301NotificationsPackage 4};

notificationProfilePackageBehaviour BEHAVIOUR

DEFINED AS

“This package has been defined to allow the Manager to get detailed information about the profile of Notification IRP.

The action ‘getOperationProfile’ is invoked by the Manager to get detailed information about the operations supported by Notification IRP.

The action ‘getNotificationProfile’ is invoked by the Manager to get detailed information about the notifications supported by Notification IRP.”;

## 5.2.5 notificationSubscriptionFilterPackage

```
notificationSubscriptionFilterPackage PACKAGE
    BEHAVIOUR
        notificationSubscriptionFilterPackageBehaviour;
    ACTIONS
        changeSubscriptionFilter;
```

REGISTERED AS { ts32-301NotificationsPackage 5};

notificationSubscriptionFilterPackageBehaviour BEHAVIOUR

DEFINED AS

“This Package provides the Manager with the capability to change the subscription filter.

The action 'changeSubscriptionFilter' provides the Manager with the capability to change the active filter for the current subscription.”;

## 5.2.6 notificationSubscriptionControlPackage

```
notificationSubscriptionControlPackage PACKAGE
    BEHAVIOUR
        notificationSubscriptionControlPackageBehaviour;
    ACTIONS
        suspendSubscription,
        resumeSubscription;
```

REGISTERED AS { ts32-301NotificationsPackage 6};

notificationSubscriptionControlPackageBehaviour BEHAVIOUR

DEFINED AS

“This package provides the Manager with the capability to control the subscriptions.

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

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

**3GPP TSG-SA5 (Telecom Management)** **S5F010127**
**Meeting #21, Naperville, IL, USA 16 – 20 July 2001**

CR-Form-v4

# CHANGE REQUEST

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ⌘ | **32.304** | **CR** | **003** | ⌘ | ev | **-** | ⌘ | Current version: | **4.0.0** | ⌘ |

*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 ☒ Core Network ☒

| | | |
|---|---|---|
| ***3. Title:*** | ⌘ | OID corrections according to the new TS 32.304 number |
| ***Source:*** | ⌘ | SA5 |
| ***Work item code:*** | ⌘ | OAM-FM |
| ***Date:*** | ⌘ | 20/07/2001 |
| ***Category:*** | ⌘ | **F** |
| ***Release:*** | ⌘ | REL-4 |

*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)* |

| | | |
|---|---|---|
| ***4. Reason for change:*** | ⌘ | Since the numbering schema of the SA5 TS documents has been changed, also the OID values of the objects defined in such documents need to be changed. |
| ***5. Summary of change:*** | ⌘ | OID values are assigned according to the new number of TS 32.304. |
| ***6. Consequences if not approved:*** | ⌘ | Erroneous OID values within TS 32.304. |

| | | | | |
|---|---|---|---|---|
| ***Clauses affected:*** | ⌘ | 5.x.x | | |
| ***Other specs affected:*** | ⌘ | □ | Other core specifications | ⌘ |
| | | □ | Test specifications | |
| | | □ | O&M Specifications | |
| ***Other comments:*** | ⌘ | | | |

## 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.

# 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",
notificationSubscriptionFilterPackage PRESENT IF "an instance supports it",
notificationSubscriptionControlPackage PRESENT IF "an instance supports it ";

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

~~notificationControl **MANAGED OBJECT CLASS**~~
~~**DERIVED FROM**~~
~~”Rec. X.721 | ISO/IEC 10165-2 : 1992”:top;~~
~~**CHARACTERIZED BY**~~
~~notificationControlBasicPackage,~~
~~notificationControlInfoPackage,~~
~~notificationProfilePackage,~~
~~notificationIRPVersionPackage;~~
~~**REGISTERED AS** { ts32-301NotificationsObjectClass 1};~~

## 5.2 Packages

### 5.2.1 notificationControlBasicPackage

notificationControlBasicPackage PACKAGE
BEHAVIOUR
notificationControlBasicPackageBehaviour;
ATTRIBUTES
notificationControlId GET;
ACTIONS
changeSubscriptionFilter,
resumeSubscription,
subscribe,
suspendSubscription,
unsubscribe;
REGISTERED AS { ts32-~~301~~304NotificationsPackage 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

```
notificationControlBasicPackage PACKAGE
    BEHAVIOUR
        notificationControlBasicPackageBehaviour;
    ATTRIBUTES
        notificationControlId     GET;
    ACTIONS
        subscribe,
        unsubscribe;
REGISTERED AS { ts32-304NotificationsPackage 1};
~~notificationControlInfoPackage PACKAGE~~
~~    BEHAVIOUR~~
~~        notificationControlInfoPackageBehaviour;~~
~~    ATTRIBUTES~~
~~        supportedNotificationCategories         GET;~~
~~    ACTIONS~~
~~        getNotificationCategories,~~
~~        getSubscriptionStatus,~~
~~        getSubscriptionIds;~~
~~REGISTERED AS { ts32-301NotificationsPackage 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-~~301~~304NotificationsPackage 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.”;

### 5.2.4 notificationProfilePackage

```
notificationProfilePackage PACKAGE
    BEHAVIOUR
        notificationProfilePackageBehaviour;
    ACTIONS
        getOperationProfile,
        getNotificationProfile;
```

REGISTERED AS { ts32-~~301~~304NotificationsPackage 4};

notificationProfilePackageBehaviour BEHAVIOUR

DEFINED AS

“This package has been defined to allow the Manager to get detailed information about the profile of Notification IRP.

The action ‘getOperationProfile’ is invoked by the Manager to get detailed information about the operations supported by Notification IRP.

The action ‘getNotificationProfile’ is invoked by the Manager to get detailed information about the notifications supported by Notification IRP.”;

### 5.2.5 notificationSubscriptionFilterPackage

```
notificationSubscriptionFilterPackage PACKAGE
    BEHAVIOUR
        notificationSubscriptionFilterPackageBehaviour;
    ACTIONS
        changeSubscriptionFilter;
```

REGISTERED AS { ts32-304NotificationsPackage 5};

notificationSubscriptionFilterPackageBehaviour BEHAVIOUR

DEFINED AS

“This Package provides the Manager with the capability to change the subscription filter.

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

### 5.2.6 notificationSubscriptionControlPackage

```
notificationSubscriptionControlPackage PACKAGE
    BEHAVIOUR
        notificationSubscriptionControlPackageBehaviour;
    ACTIONS
        suspendSubscription,
        resumeSubscription;
```

REGISTERED AS { ts32-304NotificationsPackage 6};

notificationSubscriptionControlPackage BEHAVIOUR

DEFINED AS

“This package provides the Manager with the capability to control the subscriptions.

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

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

## 5.3 Actions

### 5.3.1 changeSubscriptionFilter (O)

changeSubscriptionFilter **ACTION**
**BEHAVIOUR**
changeSubscriptionFilterBehaviour;
**MODE**
CONFIRMED;
**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.ChangeSubscriptionFilter;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.ChangeSubscriptionFilterReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 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 *notificatioCategories* (as specified in the *subscribe* action)

  – 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:

- *status*

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

### 5.3.2 getNotificationCategories (O)

getNotificationCategories **ACTION**
**BEHAVIOUR**
getNotificationCategoriesBehaviour;
**MODE**
CONFIRMED;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.GetNotificationCategoriesReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 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 subscribtion.

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 getNotificationIRPVersion (M)

getNotificationIRPVersion **ACTION**
**BEHAVIOUR**
getNotificationIRPVersionBehaviour;
**MODE**
CONFIRMED;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.GetNotificationIRPVersionReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 3};

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).”;

### 5.3.4 getNotificationProfile (O)

```
getNotificationProfile ACTION
    BEHAVIOUR
        getNotificationProfileBehaviour;
    MODE
        CONFIRMED;
    WITH INFORMATION SYNTAX
        TS32-~~301~~304~~-4~~TypeModule.IRPVersionNumber;
    WITH REPLY SYNTAX
        TS32-~~301~~304~~-4~~TypeModule.GetNotificationProfileReply;
REGISTERED AS { ts32-~~301~~304NotificationsAction 4};
```

getNotificationProfileBehaviour **BEHAVIOUR**

**DEFINED AS**

“A Manager 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-~~301~~304~~-4~~TypeModule.IRPVersionNumber;
    WITH REPLY SYNTAX
        TS32-~~301~~304~~-4~~TypeModule.GetOperationProfileReply;
REGISTERED AS { ts32-~~301~~304NotificationsAction 5};
```

getOperationProfileBehaviour **BEHAVIOUR**

**DEFINED AS**

“A Manager 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).”;

### 5.3.6 getSubscriptionIds (O)

getSubscriptionIds **ACTION**
**BEHAVIOUR**
getSubscriptionIdsBehaviour;
**MODE**
CONFIRMED;
**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.GetSubscriptionIds;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.GetSubscriptionIdsReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 6};

getSubscriptionIdsBehaviour **BEHAVIOUR**

**DEFINED AS**

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

The 'Action information' field contains the following data:

- *managerReference*

  This parameter identifies unambiguously the Manager invoking the current operation.

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

- *subscriptionIdList*

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

- *status*

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

### 5.3.7 getSubscriptionStatus (O)

getSubscriptionStatus **ACTION**
**BEHAVIOUR**
getSubscriptionStatusBehaviour;
**MODE**
CONFIRMED;
**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.GetSubscriptionStatus;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule. GetSubscriptionStatusReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 7};

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).”;

### 5.3.8 resumeSubscription (O)

resumeSubscription **ACTION**
**BEHAVIOUR**
resumeSubscriptionBehaviour;
**MODE**
CONFIRMED;

**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.ResumeSubscription;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.ResumeSubscriptionReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 8};

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).”;

### 5.3.9 subscribe (M)

subscribe **ACTION**
**BEHAVIOUR**
subscribeBehaviour;
**MODE**
CONFIRMED;
**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.Subscribe;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.SubscribeReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 9};

subscribeBehaviour **BEHAVIOUR**

**DEFINED AS**

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

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

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

The 'Action information'  contains the following data:

- *managerReference*

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

- *destination*

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

- *filter*

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

- *timeTick*

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

- *notificationCategoryList*

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

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

- Parameter *filter*
- Parameter *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).";

## 5.3.10 suspendSubscription (O)

suspendSubscription **ACTION**

**BEHAVIOUR**

suspendSubscriptionBehaviour;

**MODE**

CONFIRMED;

**WITH INFORMATION SYNTAX**

TS32-~~301~~304~~-4~~TypeModule.SuspendSubscription;

**WITH REPLY SYNTAX**

TS32-~~301~~304~~-4~~TypeModule.SuspendSubscriptionReply;

**REGISTERED AS** { ts32-~~301~~304NotificationsAction 10};

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).”;

### 5.3.11 unsubscribe (M)

unsubscribe **ACTION**
**BEHAVIOUR**
unsubscribeBehaviour;
**MODE**
CONFIRMED;
**WITH INFORMATION SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.Unsubscribe;
**WITH REPLY SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.UnsubscribeReply;
**REGISTERED AS** { ts32-~~301~~304NotificationsAction 11};

unsubscribeBehaviour **BEHAVIOUR**

**DEFINED AS**

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

The 'Action information' contains the following data:

- *managerReference*

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

- *subscriptionId*

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

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

- *status*

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

## 5.4 Attributes

### 5.4.1 notificationControlId

notificationControlId **ATTRIBUTE**
**WITH ATTRIBUTE SYNTAX**

TS32-~~301~~304~~-4~~TypeModule.GeneralObjectId;
**MATCHES FOR** EQUALITY;
**BEHAVIOUR** notificationControlIdBehaviour;
**REGISTERED AS** { ts32-~~301~~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-~~301~~304~~-4~~TypeModule.NotificationCategory;
**MATCHES FOR**
EQUALITY;
**BEHAVIOUR**
supportedNotificationCategoriesBehaviour;
**REGISTERED AS** { ts32-~~301~~304NotificationsAttribute 2};

supportedNotificationCategoriesBehaviour **BEHAVIOUR**

**DEFINED AS**

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

### 5.4.3 supportedNotificationIRPVersions

supportedNotificationIRPVersions **ATTRIBUTE**
**WITH ATTRIBUTE SYNTAX**
TS32-~~301~~304~~-4~~TypeModule.SupportedNotificationIRPVersions;
**MATCHES FOR**
EQUALITY;
**BEHAVIOUR**
supportedNotificationIRPVersionsBehaviour;
**REGISTERED AS** { ts32-~~301~~304NotificationsAttribute 3};

supportedNotificationIRPVersionsBehaviour **BEHAVIOUR**

**DEFINED AS**

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

# 6 ASN.1 definitions

**TS32-304~~1-4~~TypeModule {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-Operation-Maintenance(3)**

**ts-32-304~~1~~(304~~1~~)~~ part4(4)~~ informationModel(0) asn1Module(2) version1(1)}**

```
    DEFINITIONS IMPLICIT TAGS ::=
    BEGIN

    --EXPORTS everything
    IMPORTS
    Destination, DiscriminatorConstruct
    FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}
    CMISFilter
    FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)};

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

ts32-304~~1~~Prefix              OBJECT IDENTIFIER ::= { baseNodeUMTS ts-32-304~~1~~(304~~1~~)}
~~ts32-301Part4               OBJECT IDENTIFIER ::= { ts32-301Prefix part4(4)}~~
ts32-304~~1-4~~InfoModel         OBJECT IDENTIFIER ::= { ts32-304~~1Part4~~Prefix informationModel(0)}

ts32-304~~1~~NotificationsObjectClass OBJECT IDENTIFIER ::= { ts32-304~~1-4~~InfoModel managedObjectClass(3)}
ts32-304~~1~~NotificationsPackage     OBJECT IDENTIFIER ::= { ts32-304~~1-4~~InfoModel package(4)}
ts32-304~~1~~NotificationsAttribute      OBJECT IDENTIFIER ::= { ts32-304~~1-4~~InfoModel attribute(7)}
ts32-304~~1~~NotificationsAction      OBJECT IDENTIFIER ::= { ts32-304~~1-4~~InfoModel action(9)}



    -- Start of 3GPP SA5 own definitions

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

    ChangeSubscriptionFilter ::= SEQUENCE
        {
        subscriptionId          GraphicString,
        filter                  CMISFilter    -- ITU-T X.711
        }

    ChangeSubscriptionFilterReply ::= SEQUENCE
        {
        status                  ErrorCauses
        }
```

```
GeneralObjectId ::= INTEGER

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

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

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

GetOperationProfileReply ::= SEQUENCE
    {
    operationNameProfile             OperationList,
    operationParameterProfile        ParameterListOfList,
    status                           ErrorCauses
    }


GetSubscriptionStatus ::= SEQUENCE
    {
    subscriptionId               GraphicString
    }

GetSubscriptionStatusReply ::= SEQUENCE
    {
    notificationCategoryList     NotificationCategoryList,
    filterInEffect               CMISFilter,   -- ITU-T X.711
    subscriptionState            SubscriptionState OPTIONAL,
    timeTick                     INTEGER OPTIONAL,
    status                       ErrorCauses
    }

GetSubscriptionIds ::= SEQUENCE
    {
    managerReference    INTEGER
    }

GetSubscriptionIdsReply ::= SEQUENCE
    {
    subscriptionIdList     SubscriptionIdList,
    status                 ErrorCauses
    }

IRPVersionNumber ::= GraphicString

NotificationCategory ::= ENUMERATED
    {
    alarm                  (1),--the notification category defined in the alarm IRP
    basicCM                (2) --the notification category defined in the basic CM IRP

    }
```

```
NotificationCategoryList ::= SET OF NotificationCategory

NotificationList ::= SET OF NotificationName

NotificationName ::= GraphicString

OperationList ::= SET OF OperationName

OperationName ::= GraphicString

ParameterList ::= SET OF ParameterName

ParameterListOfList ::= SET OF ParameterList

ParameterName ::= GraphicString

ResumeSubscription ::= SEQUENCE
    {
    subscriptionId              GraphicString
    }

ResumeSubscriptionReply ::= SEQUENCE
    {
    status                      ErrorCauses
    }

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

SubscribeReply ::= SEQUENCE
    {
    subscriptionId          GraphicString,
    status                  ErrorCauses
    }

SubscriptionIdList ::= SET OF GraphicString

SubscriptionState ::= ENUMERATED
    {
    suspended           (0),
    notSuspended        (1)
    }

SupportedNotificationIRPVersions ::= SET OF IRPVersionNumber

SuspendSubscription ::= SEQUENCE
    {
    subscriptionId              GraphicString
    }

SuspendSubscriptionReply ::= SEQUENCE
    {
    status                      ErrorCauses
    }
```

```
Unsubscribe ::= SEQUENCE
    {
    managerReference    INTEGER,
    subscriptionId              GraphicString
    }

UnsubscribeReply ::= SEQUENCE
    {
    status            ErrorCauses
    }

END -- of TS32-304TypeModule ~~module TS32-301-NotificationsAsn1TypeModule~~
```