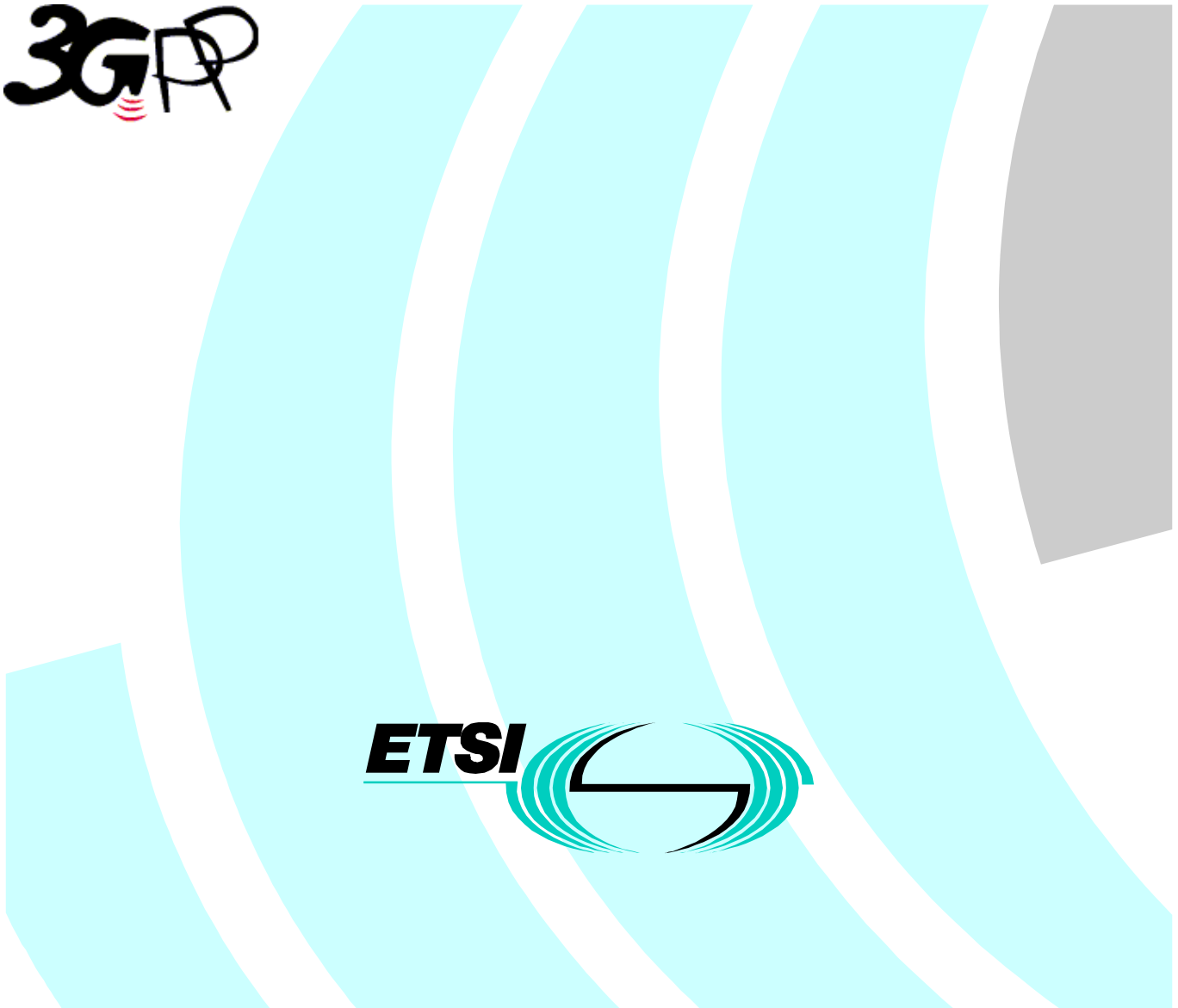
ETSI TS 132 302 V4.0.0 (2001-06)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
Telecommunication Management;
Configuration Management;
Notification Integration Reference Point;
Information Service version 1
(3GPP TS 32.302 version 4.0.0 Release 4)



Reference
DTS/TSGS-0532302Uv4

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

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://www.etsi.org/ipr).

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 the 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 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 [5] and 3GPP TS 32.102 [6].

Network Elements (NEs) under management and element managers generate notifications of events about occurrences within the network. Different kinds of events carry different kinds of information. For instance a new alarm as specified in Alarm IRP: Information Service [1], is one possible kind of event, an object creation as specified in Basic CM IRP : Information Service [8] is another possible kind of event.

Information of an event is carried in notification. An IRPAgent (typically an EM or a NE) emits notifications. IRPManager (typically a network management system) receives notifications. The purpose of Notification IRP is to define an interface through which an IRPManager can subscribe to IRPAgent for receiving notifications.

This IRP bases its design on work captured in ITU-T Recommendation X.734 [2], OMG Notification Service [4]. The central design ideas are:

- Separation of notification Consumers (IRPManagers) from Producers (IRPAgents);
- Notifications are sent to IRPManagers without the need for IRPManagers to periodically check for new notifications.

Common characteristics related to notifications in all other IRPs are gathered in one IRP.

1 Scope

The purpose of Notification IRP is to define an interface through which an IRPManager can subscribe to an IRPAgent for receiving notifications. This document is the « Information Service » of Notification IRP. It defines, for the purpose of subscribing to an IRPAgent for receiving notifications, the information observable and controlled by management system's client and it also specifies the semantics of the interactions used to carry this information. It also defines the information common to all notifications which is called the notificationHeader.

An IRPAgent supporting this IRP IS may emit one or multiple categories of notifications, such as alarms (as specified in Alarm IRP : Information Service [1]) and others. This IRP IS defines a mechanism that IRPManager can use to determine the categories of notifications supported by an IRPAgent. It also defines a mechanism (subscribe and unsubscribe operations) that IRPManager can use to specify the categories of notifications IRPAgent should emit to IRPManager during subscription. It also defines a mechanism (getSubscriptionIds operation) that IRPManager can use to check which categories of notifications it has subscribed to. IRPManager can set and change filter criteria applicable during the life-cycle of a subscription. IRPManager can also exercise flow-control on IRPAgent's emission of notifications (suspendSubscription and resumeSubscription operations).

Using different managerReference, an IRPManager can subscribe several times. It will result in multiple subscriptions. As far as IRPAgent is concerned, notifications are sent to multiple "places".

Using the same managerReference, an IRPManager can subscribe several times specifying different categories of notifications.

This IRP IS does not specify information that is carried in some but not all notifications. That kind of information is specified in other IRP ISs involved. For example, `perceivedSeverity` is a piece of information specific for notifications carrying alarm information. This information is not defined in the present document but in Alarm IRP : Information Service [1].

How IRPManager discovers the IRPAgent's address or reference (so that IRPManager can invoke an operation) is outside the scope of the present document.

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 32.111-2: “Alarm IRP: Information Service”.

[2] ITU-T Recommendation X.734 (09/92): “Information technology - Open Systems Interconnection - Systems management: Event report management function”.

[3] 3GPP TS 32.300: “Name Convention for Managed Objects”.

[4] OMG: “OMG Notification Service”.

[5] 3GPP TS 32.101: "3G Telecom Management principles and high level requirements".

[6] 3GPP TS 32.102: "3G Telecom Management architecture".

[7] 3GPP TS 32.301: “Notification IRP : Requirements”.

[8] 3GPP TS 32.602: “ Basic CM IRP : Information Service”.

[9] 3GPP TS 32.622: “ Generic Network Resources IRP : Network Resource Model”.

[10] 3GPP TS 32.312: “ Generic IRP Management : Information Service”.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply. For terms and definitions not found here, please refer to 3GPP TS 32.101 [5], 3GPP TS 32.102 [6] and 3GPP TS 32.301 [7].

IRPAgent: See 3GPP TS 32.102 [6].

IRPManager: See 3GPP TS 32.102 [6].

Event: It is an occurrence that is of significance to network operators, the NEs under surveillance and network management applications. Events can indicate many types of network management information, such as network alarms, network configuration change information and network performance data.

Notification: It refers to the transport of information regarding events from event producer to consumer (receiver). In this IRP, notification is used to carry information about network events from IRPAgent to IRPManager. Producer sends notifications to consumers as soon as new events occur. Consumer does not need to check ("pull") for events.

IRP : See 3GPP TS 32.102 [6].

Notification Category : It refers to the set of notifications of one 3GPP IRP Information Service specification. A Notification Category is identified by the name of the IRP specification and the IRP specification version number.

Qualifiers: The meaning of qualifiers for operations, parameters and information attributes (whether they are Mandatory(M)/ Conditional(C)/ Optional(O)) defined in the present (Information Service) document is provided in 3GPP TS 32.102 [6]. Moreover, qualifiers of information attributes, when those information attributes are re-used in other IRP ISs, obey to the following rule : Mandatory and Conditional qualifiers of information attributes shall always be the same in other IRPs ISs, Optional qualifiers of information attributes may be set to either Optional or Mandatory in the other IRP ISs.

3.2 Abbreviations

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

CM	Configuration Management
CORBA	Common Object Request Broker Architecture
DN	Distinguished Name
EM	Element Manager
IOC	Information Object Class
IRP	Integration Reference Point
IS	Information Service
ITU-T	International Telecommunication Union, Telecommunication Standardisation Sector
NE	Network Element
NM	Network Manager
NR	Network Resource
NRM	Network Resource Model
OMG	Object Management Group
SS	Solution Set
UML	Unified Modelling Language (OMG)

4 System overview

4.1 System context for Notification

Figure 1 and Figure 2 identify System contexts of Notification IRP in terms of implementations called IRPAgent and IRPManager.

"IRPManager" depicts a process that interacts with IRPAgent for the purpose of receiving network Notifications via this IRP. IRPAgent detects network events. IRPAgent sends IRPManagers notifications carrying the events. Examples of IRPManagers can be a process running supporting network Notification logging device or supporting network Notification viewing devices (such as a local craft terminal) or a process running within a Network Manager (NM) as shown in Figure 1 and Figure 2. IRPAgent implements and supports this IRP. IRPAgent can run within one Element Manager (EM) with one or more NEs (see Figure 1) or run within one NE (see Figure 2). In the former case, the interfaces (represented by a thick dotted line) between the EM and the NEs are not subject of this IRP. Whether EM and NE share the same hardware system is not relevant to this IRP either. By observing the interaction across the IRP, one cannot deduce if EM and NE are integrated in a single system or if they run in separate systems.

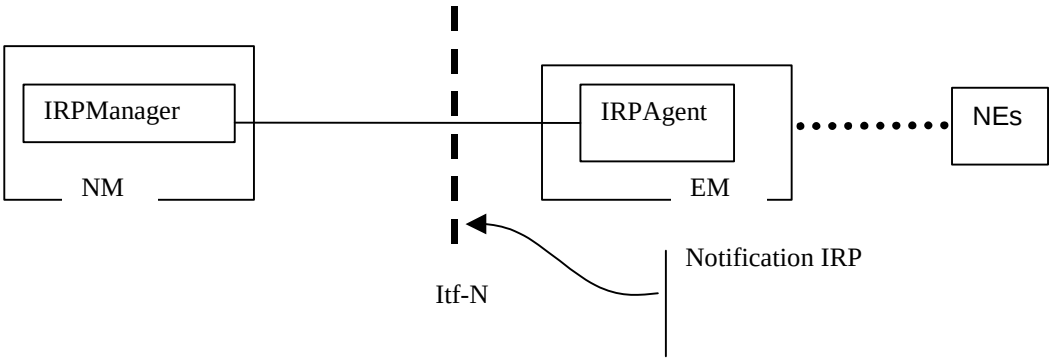


Figure 1: System Context A

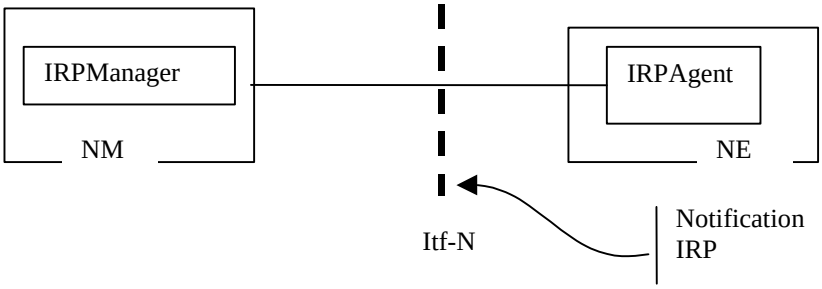


Figure 2: System Context B

5 Information Object Classes

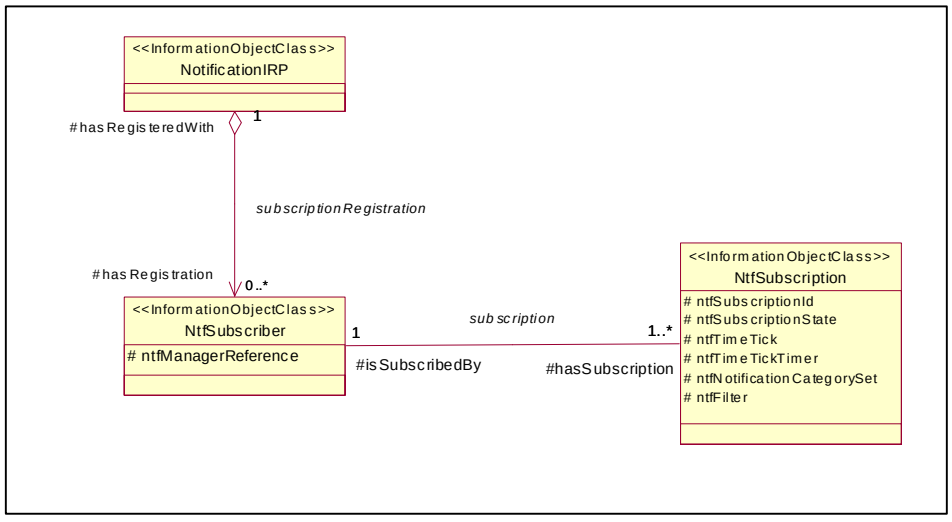
5.1 Information entities imported and local labels

Label reference	Local label
32.622 [9], information object class, Top	Top
32.312 [10], information object class, managedGenericIRP	managedGenericIRP
32.622 [9], information object class, IRPAgent	IRPAgent

5.2 Class Diagram

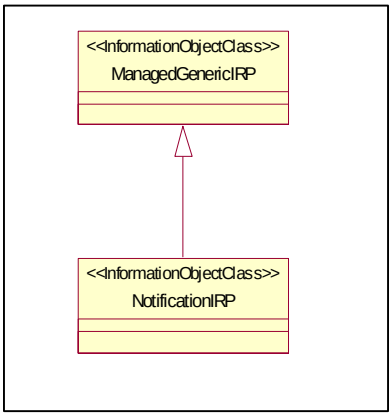
5.2.1 Attributes and relationships

This sub-clause depicts the set of IOCs that encapsulate information within the notification IRP. The intent is to identify the information required for the notification IRP implementation of its operations and notification emission. This sub-clause provides the overview of all information object classes in UML. Subsequent sub-clauses provides more detailed specification of various aspects of these information object classes.



5.2.2 Inheritance

This sub-clause depicts the inheritance relationships that exists between information object classes.



5.3 Information object classes definition

5.3.1 NtfSubscriber

5.3.1.1 Definition

This information object represents a Subscriber from a notification IRP perspective : a subscriber is fully identified by a manager reference. An IRPManager using multiple managerReference attributes to subscribe will result in multiple NtfSubscriber instances. It inherits from IOC Top.

5.3.1.2 Attributes

Attribute name	Support Qualifier
ntfManagerReference	M

5.3.2 NtfSubscription

5.3.2.1 Definition

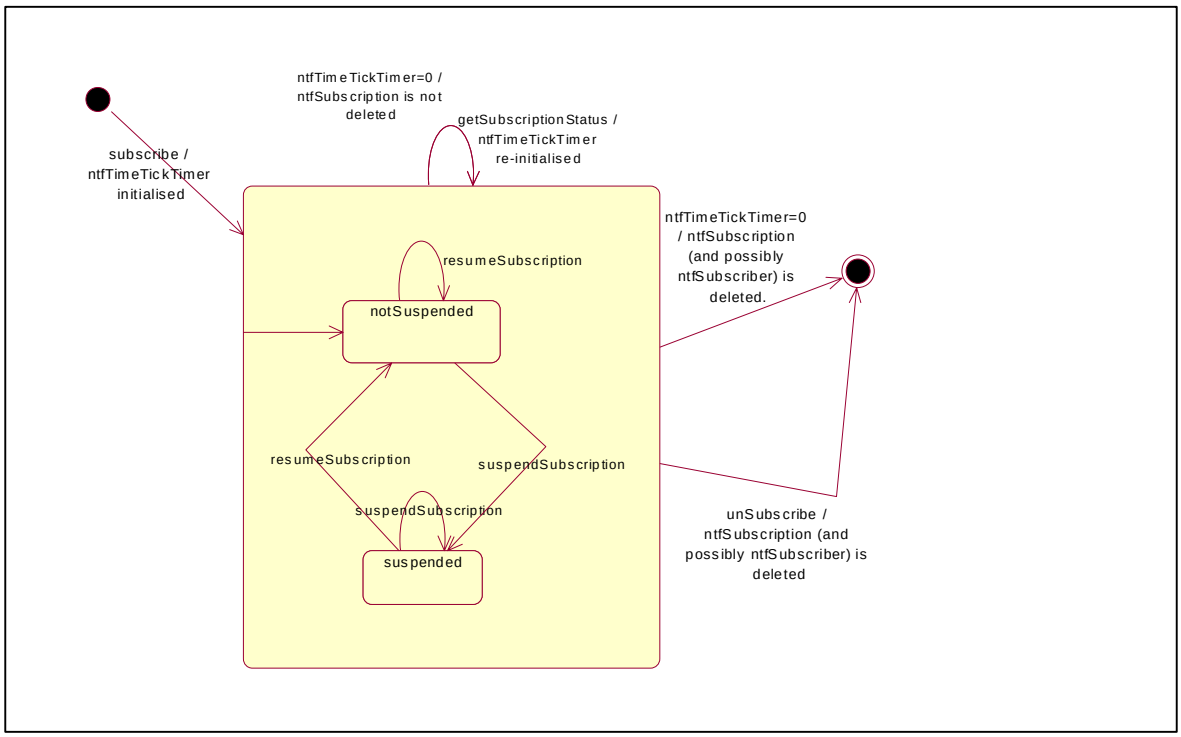
This information object represents a subscription that have been requested by an IRPManager and created. It inherits from IOC Top.

5.3.2.2 Attributes

Attribute name	Support Qualifier
ntfSubscriptionId	M
subscriptionState	M
ntfTimeTick	M
ntfTimeTickTimer	M
ntfNotificationCategorySet	M
ntfFilter	M

5.3.2.3 State diagram

The diagram below depicts states that can be supported by a NtfSubscription.



NotificationIRP can lose the list of `managerReference` that identifies current IRPManagers under subscription. Under this condition, IRPAgent is incapable of sending events to the affected subscriber(s).

This Notification IRP recommends that IRPManager should invoke the `getSubscriptionStatus` operation periodically to confirm that IRPAgent still has the IRPManager's reference in its list. In case getSubscriptionStatus returns the exception operation_failed, IRPManager should assume that IRPAgent has lost the IRPManager's reference.

This IRP does not recommend the frequency IRPManager should use to invoke `getSubscriptionStatus` operation.

5.3.3 NotificationIRP

5.3.3.1 Definition

This information object represents a notification IRP. It inherits from IOC managedGenericIRP.

5.4 Information relationships definition

5.4.1 subscription (M)

5.4.1.1 Definition

This relationship defines the relationship between a NtfSubscriber and its current subscriptions.

5.4.1.2 Roles

Name	Definition
isSubscribedBy	This role represents the one who has subscribed. It can be played by instances of IOC NtfSubscriber
hasSubscription	This role represents the subscriptions which were made and not unsubscribed. It can be played by instances of IOC NtfSubscription

5.4.1.3 Constraints

Name	Definition
inv_notificationCategoriesAllDistinct	"the notification categories contained in the ntfNotificationCategorySet attribute of NtfSubscription playing the role hasSubscription are all distinct from each other"

5.4.2 subscriptionRegistration (M)

5.4.2.1 Definition

This relationship defines the relationship between the NotificationIRP and the current subscribers of notifications.

5.4.2.2 Roles

Name	Definition
hasRegistration	This role represents the entities to which IRPAgent will notify events. It is played by instances of IOC NtfSubscriber
HasRegisteredWith	This role represents the NotificationIRP to which an IRPManager has subscribed. It is played by instances of IOC NotificationIRP

5.4.2.3 Constraints

Name	Definition
inv_uniqueManagerReference	"all NtfSubscriber involved in the subscriptionRegistration relationship with NotificationIRP are distinguished from each other by their ntfManagerReference Attribute "

5.5 Information attributes definition

This sub-clause defines the semantics of the Attributes used in Information Object Classes.

5.5.1 Definitions and legal values

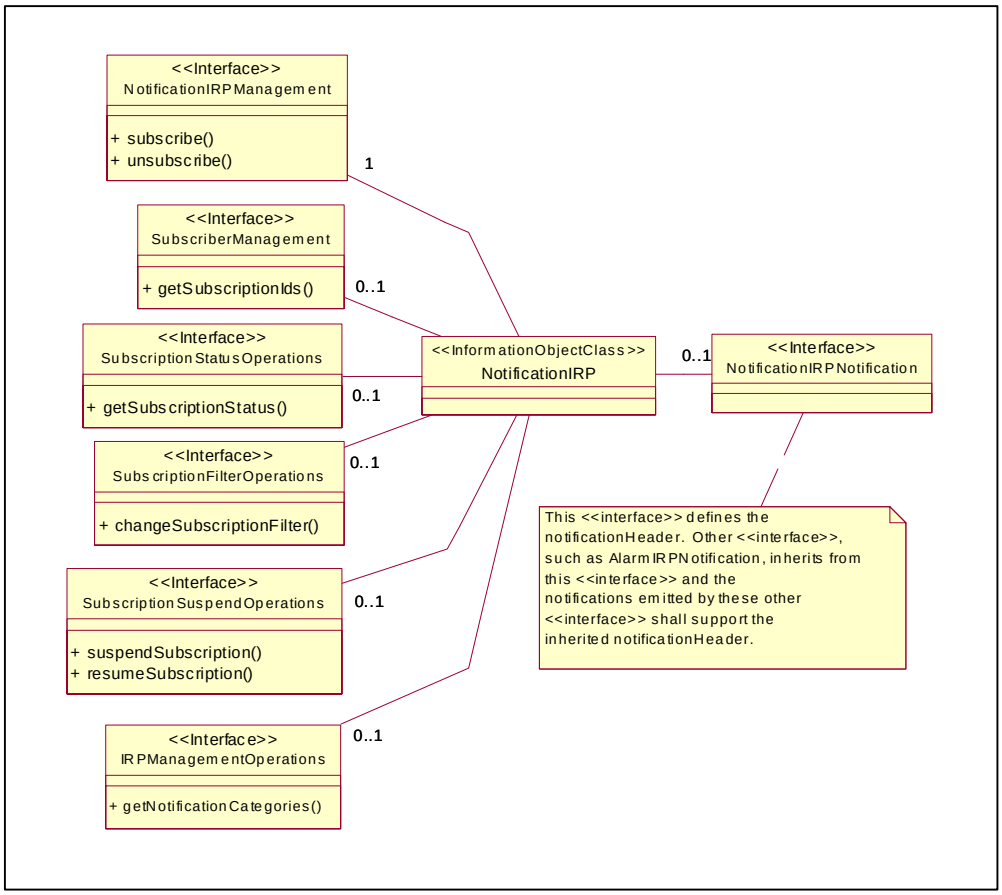
Attribute Name	Definition	Legal Values
ntfSubscriptionId	It identifies uniquely a subscription	N/A
ntfSusbcriptionState	It indicates the activation state of a subscription	"suspended" : the subscription is suspended "notSuspended" : the subscription is active
ntfTimeTick	this Attribute represents the initial value of ntfTimeTickTimer. It is in unit of whole minute. This value defines a time window within which IRPManager intends to invoke `getSubscriptionStatus (`or `subscribe)` operation to confirm its subscription. A special value indicates infinity which is such that timer will never expire and IRPAgent needs other means to decide when to delete resources allocated to the IRPManager	Integer greater or equal to 15, OR special infinite value
ntfTimeTickTimer	this Attribute represents the current value of a timer	integer greater or equal to zero
ntfNotificationCategorySet	this Attribute represents a set of notification categories (see also Definition of notification category in clause 3.1)	
ntfFilter	this Attribute represents the filter of a subscription. The filter can be applied to parameters of notification header (see NotificationIRPNotification interface) and to parameters of notifications defined as filterable in other IRP ISs. IRPAgent shall notify IRPManagers if the event satisfies the filter constraint.	
ntfManagerReference	this Attribute contains the reference of a manager. It uniquely identifies a subscriber	

5.5.2 Constraints

- "ntfTimeTickTimer is lower or equal to ntfTimeTick"

6 Interface Definition

6.1 Class diagram representing interfaces



6.2 Generic rules

- rule 1 : each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

- rule 2 : Each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

- rule 3 : each operation shall support a generic exception operation_failed_internal_problem which is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

6.3 notificationIRPManagement Interface

6.3.1 Operation `subscribe` (M)

6.3.1.1 Definition

IRPManager invokes this operation to establish subscription to receive network events via notifications, under the filter constraint specified in this operation.

6.3.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`managerReference`	M	NtfSubscriber.ntfManagerReference	It specifies the reference of IRPManager to which notifications shall be sent.
`timeTick`	O	NtfSubscription.ntfTimeTick	It specifies the value of a timer hold by NotificationIRP for the subject IRPManager. The value is in unit of whole minute. A special infinite value is assumed when parameter is absent or present but equal to zero.
`notification Categories`	O	SET OF (name of IRP, version of IRP)	It identifies one or more Notification Category (see also Definition in subclause 3.1)
`filter`	O	NtfSubscription.ntfFilter Filter constraint grammar is SS dependent	It specifies a filter constraint that IRPAgent shall use to filter notification of the category specified in `notificationCategories` parameter. If this parameter is absent, then no filter constraint shall be applied.

6.3.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`subscriptionId`	M	NtfSubscription.ntfSubscriptionId	It holds an unambiguous identity of this subscription.
`status`	M	ENUM (OperationSucceeded, OperationFailedExistingSubscription, OperationFailed)	If subscriptionCreated is true, status = OperationSuceeded. If operation_failed_existing_subscription is true, status = OperationFailedExistingSubscription If operation_failed is true, status = OperationFailed.

6.3.1.4 Pre-condition

notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed

Assertion Name	Definition
notificationCategoriesNotAllSubscribed	"at least one notificationCategory identified in the notificationCategories input parameter is supported by IRPAgent and is not a member of the ntfNotificationCategorySet attribute of an NtfSubscription which is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter".
notificationCategoriesParameterAbsentAndNotAllSubscribed	" notificationCategories input parameter is absent and at least one notificationCategory supported by IRPAgent is not a member of the ntfNotificationCategorySet attribute of an ntfSsubscription which is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter"

6.3.1.5 Post-condition

subscriberPossiblyCreated AND subscriptionCreated

Assertion Name	Definition
subscriberPossiblyCreated	"an NtfSubscriber with a ntfManagerReference attribute equal to the value of the managerReference input parameter is involved in a subscriptionRegistration relationship with NotificationIRP"
subscriptionCreated	"an NtfSubscription has been created according to the following rules : - subscriptionState attribute value has been set to "notSuspended", - ntfTimeTick attribute value has been set to the value of the timeTick input parameter if this value was higher or equal to 15, or set to 15 if this parameter value was between 1 and 15, or set to a special infinite value if the parameter value was lower or equal to 0 or if parameter was absent, - ntfTimeTickTimer has been reset with the value of timeTick attribute, - ntfFilter attribute value has been set to the value of the filter input parameter if present, - NtfSubscription is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter, - attribute ntfNotificationCategorySet of NtfSubscription contains EITHER the notification categories identified by the notificationCategories input parameter that were not already contained in the ntfNotificationCategorySet attribute of other NtfSubscription of the same NtfSubscriber identified by the managerReference input parameter OR if notificationCategories input parameter is absent, all notification categories supported by IRPAgent that were not already contained in the ntfNotificationCategorySet attribute of other subscriptions of the same NtfSubscriber identified by the managerReference input parameter ".

6.3.1.6 Exceptions

Name	Definition
operation_failed_existing_subscription	**Condition:** (notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed) not true **Returned Information:** The output parameter status. **Exit state:** Entry State
Operation_failed	**Condition:** Post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.3.2 Operation `unsubscribe` (M)

6.3.2.1 Definition

The IRPManager invokes this operation to cancel subscriptions. The IRPManager can cancel one subscription made with a managerReference by providing the corresponding subscriptionId or all subscriptions made with the same managerReference by leaving the subscriptionId parameter absent.

6.3.2.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`managerReference`	M	NtfSubscriber.ntfManagerReference	It specifies the reference of an IRPManager.
`subscriptionId`	O	NtfSubscription.ntfSubscriptionId	It holds a `subscriptionId` carried as the output parameter in the `subscribe` operation.

6.3.2.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`status`	M	ENUM (OperationSucceeded, OperationFailed)	If (subscriptionDeleted OR allSubscriptionDeleted) is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

6.3.2.4 Pre-condition

validSubscriptionId&ManagerReference OR SubscriptionIdAbsent&ValidManagerReference

Assertion Name	Definition
validSubscriptionId&ManagerReference	"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter"
SubscriptionIdAbsent&ValidManagerReference	"subscriptionId input parameter is absent and the NtfSubscriber identified by the managerReference input parameter exists"

6.3.2.5 Post-condition

subscriptionDeleted OR allSubscriptionDeleted

Assertion Name	Definition
subscriptionDeleted	"the NtfSubscription identified by subscriptionId input parameter is no more involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter and has been deleted. If this NtfSubscriber has no more NtfSubscription, it is deleted as well."
allSubscriptionDeleted	"in the case subscriptionId input parameter was absent, the NtfSubscriber identified by the managerReference input parameter is no more involved in any subscription relationship and is deleted, the corresponding NtfSubscription have been deleted as well."

6.3.2.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.4 subscriberManagement Interface

6.4.1 Operation `getSubscriptionIds` (O)

6.4.1.1 Definition

IRPManager invokes this operation to get the values of all still valid (not unsubscribed or removed by IRPAgent) `subscriptionIds` assigned by NotificationIRP as result of previously `subscribe` operations performed by this IRPManager.

6.4.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`managerReference`	M	NtfSubscriber.ntfManagerReference	It specifies the reference of IRPManager that requests the set of identifiers of active subscriptions related to this IRPManager.

6.4.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`subscriptionIdSet`	M	SET OF NtfSubscription.ntfSubscriptionId where NtfSubscription is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter	It holds a set of the `subscriptionId`, each assigned as output parameter in previous `subscribe` operations invoked by the current IRPManager. This value should contain no information if the IRPManager did not yet subscribed to that System or System lost all subscription related information.
`status`	M	ENUM (Operation succeeded, Operation failed)	If validSubscriptionIdSet is true, status = OperationSuceeded. If operation_failed is true, status = OperationFailed.

6.4.1.4 Pre-condition

validManagerReference

Assertion Name	Definition
validManagerReference	"the NtfSubscriber identified by the managerReference input parameter exists"

6.4.1.5 Post-condition

None specific

6.4.1.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.5 subscriptionStatusOperations Interface

6.5.1 Operation getSubscriptionStatus (O)

6.5.1.1 Definition

IRPManager invokes this operation to query the subscription status of a particular subscription. IRPManager can use getSubscriptionStatus operation to know about the filter constraint in effect, the state of subscription (i.e., if subscription is suspended/inactive or resumed/active), the timeTick value that may be set at subscribe invocation time and the notificationCategory currently in used in the subscription.

6.5.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`subscriptionId`	M	NtfSubscription.ntfSubscriptionId	It holds the `subscriptionId` carried as the output parameter in the `subscribe` operation.

6.5.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`notificationCategorySet`	M	NtfSubscription.ntfNotificationCategorySet	It identifies the notification Category(ies) supported in this subscription.
`filterInEffect`	O	NtfSubscription.ntfFilter	It contains the filter constraint currently set.
`SubscriptionState`	O	NtfSubscription.ntfSubscriptionState	
`timeTick`	O	NtfSubscription.ntfTimeTick	It carries the same value as the one in `subscribe` operation
`status`	M	ENUM (Operation succeeded, Operation failed)	If (timeTickReset) is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

6.5.1.4 Pre-condition

validSubscriptionId

Assertion Name	Definition
validSubscriptionId	"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship"

6.5.1.5 Post-condition

timeTickReset

Assertion Name	Definition
timeTickReset	"the ntfTimeTickTimer attribute of NtfSubscription identified as input parameter has been reset with the value of ntfTimeTick attribute of the same NtfSubscription "

6.5.1.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.6 subscriptionFilterOperations Interface

6.6.1 Operation changeSubscriptionFilter (O)

6.6.1.1 Definition

IRPManager invokes this operation to replace the present filter constraint with a new one.

6.6.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`subscriptionId`	M	NtfSubscription.ntfSubscriptionId	It carries the `subscriptionId` carried as the output parameter in the `subscribe` operation.
`filter`	M	NtfSubscription.ntfFilter	It specifies a filter constraint

6.6.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`status`	M	ENUM (Operation succeeded, Operation failed)	If filterUpdated is true, status = OperationSuceeded. If operation_failed is true, status = OperationFailed.

6.6.1.4 Pre-condition

validNtfSubscriptionId

Assertion Name	Definition
validNtfSubscriptionId	"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship"

6.6.1.5 Post-condition

filterUpdated

Assertion Name	Definition
filterUpdated	"ntfFilter attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to the value of the filter input parameter"

6.6.1.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.7 subscriptionSuspendOperations Interface

6.7.1 Operation `suspendSubscription` (O)

6.7.1.1 Definition

IRPManager invokes this operation to request IRPAgent to stop emission of notifications. IRPAgent may lose notification(s) if subscription is suspended.

6.7.1.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`subscriptionId`	M	NtfSubscription.ntfSubscriptionId	It carries the `subscriptionId` carried as the output parameter in the `subscribe` operation.

6.7.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`status`	M	ENUM (Operation succeeded, Operation failed)	If subscriptionStateSuspended is true, status = OperationSuceeded. If operation_failed is true, status = OperationFailed.

6.7.1.4 Pre-condition

validSubscriptionId

Assertion Name	Definition
validSubscriptionId	"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship"

6.7.1.5 Post-condition

subscriptionStateSuspended

Assertion Name	Definition
subscriptionStateSuspended	"ntfSubscriptionState attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to or kept as "suspended""

6.7.1.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.7.2 Operation `resumeSubscription` (O)

6.7.2.1 Definition

IRPManager invokes this operation to request IRPAgent to resume emission of notifications.

6.7.2.2 Input parameters

Parameter Name	Qualifier	Information Type	Comment
`subscriptionId`	M	NtfSubscription.ntfSubscriptionId	It carries the `subscriptionId` carried as the output parameter in the `subscribe` operation.

6.7.2.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`status`	M	ENUM (Operation succeeded, Operation failed)	If subscriptionStateNotSuspended is true, status = OperationSuceeded. If operation_failed is true, status = OperationFailed.

6.7.2.4 Pre-condition

validSubscriptionId

Assertion Name	Definition
validSubscriptionId	"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship"

6.7.2.5 Post-condition

subscriptionStateNotSuspended

Assertion Name	Definition
subscriptionStateNotSuspended	"ntfSubscriptionState attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to or kept as "notSuspended""

6.7.2.6 Exceptions

Name	Definition
Operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry State

6.8 IRPManagementOperations Interface

6.8.1 Operation getNotificationCategories (O)

6.8.1.1 Definition

IRPManager invokes this operation to query the categories of notification supported by IRPAgent. IRPManager does not need to be in subscription to invoke this operation.

6.8.1.2 Input parameters

None

6.8.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`NotificationCategoryList`	M	SET OF (name of IRP, version of IRP) where each IRP is contained by IRPAgent and attribute notificationNameProfile of the IRP is not empty	
`status`	M	ENUM (Operation succeeded, Operation failed)	OperationFailed only if operation_failed_internal_problem

6.8.1.4 Pre-condition

None specific

6.8.1.5 Post-condition

None specific

6.8.1.6 Exceptions

None specific

6.9 NotificationIRPNotification Interface

IRPAgent notifies the subscribed IRPManager that an event has occurred and that the event has satisfied the filter constraints used for this subscription. One event example is the notification defined in Alarm IRP: IS (3GPP TS 32.111-2 [1]).

It should be possible to pack multiple notifications together for sending to NM. This provides more efficient use of data communication resources. In order to pack multiple notifications, an EM/NE configurable parameter defines the maximum number of notifications to be packed together. Additionally an EM/NE configurable parameter defines the maximum time delay before the notifications have to be sent.

Under normal operations, an IRPAgent shall send, to each IRPManager, notifications in the same order they were generated, i.e. in the First-In, First-Out order. There shall not be any priority given to types of notifications.

This interface doesn't define any specific notification but instead defines information that is commonly found in notifications defined by other IRPs. This information is called notificationHeader. Notification interfaces defined in other IRPs, such as Alarm IRP: IS (3GPP TS 32.111-2 [1]), shall inherit from this interface and define their notifications by :

- Identifying and qualifying the Notification Header attributes for their use;
- Specify additional attributes specific to their use;

Despite the fact that the semantic of notifications is defined by other IRP ISs, it is notification IRP and not those IRP that is responsible for the emission of those notifications.

The Notification Header is defined here below :

Attribute Name	Qualifier	Comment
objectClass	M	It specifies the class name of the IOC. A network event has occurred in an instance of this class.
objectInstance	M	It specifies the instance of the above IOC in which the network event occurred by carrying the Distinguish Name (DN) of the Information Class
notificationId	O	This is an identifier for the notification, which may be used to correlate notifications. The identifier of the notification shall be chosen to be unique across all notifications of a particular managed object throughout the time that correlation is significant, it uniquely identifies the notification from other notifications generated by the subject Information Object. If IRPManager receives notifications from one IRPAgent, IRPManager shall use the identifier of the notification and the objectInstance to uniquely identify all received notifications. If IRPManager receives notifications from multiple IRPAgents and notifications of each Information Object are reported at most through one IRPAgent, IRPManager shall use the identifier of the notification and objectInstance to uniquely identify all received notifications. If IRPManager receives notifications from multiple IRPAgents and notifications of one or more Information Objects are reported through two or more IRPAgents, IRPManager shall use the identifier of the notification together with objectInstance and the identity of IRPAgent (systemDN), to uniquely identify all received notifications. If the information systemDN is absent, IRPManager needs other means, which are outside the scope of this IRP, to determine the identity of IRPAgent. How identifiers of notifications are re-used to correlate notifications is outside of the scope of this recommendation.
eventTime	M	It indicates the event occurrence time. The semantics of Generalised Time specified by ITU-T shall be used here.
systemDN	C	It carries the Distinguished Name (DN) of IRPAgent that detects the network event and generates the notification. See “Name Convention for Managed Objects” [3] for name convention regarding DN.
notificationType	M	the type of notification which is reported by the notification

All those parameters (except notificationId) shall be filterable

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

History

Document history		
V4.0.0	June 2001	Publication