

---

**Source:** SA5 (Telecom Management)  
**Title:** 2 Rel-4/5 CRs 32.102 (Telecommunication management;  
Architecture) "Add New Subclause to IS Template for Notification  
Related IOCs"  
**Document for:** Approval  
**Agenda Item:** 7.5.3

---

Doc-1st-Level	Spec	CR	R e v	Phase	Subject	Cat	Ver- Curre nt	Doc-2nd- Level	Workitem
SP-030061	32.102	026	-	Rel-4	<b>Add New Subclause to IS Template for Notification Related IOCs</b>	F	4.2.0	S5-036320	OAM-CM
SP-030061	32.102	027	-	Rel-5	<b>Add New Subclause to IS Template for Notification Related IOCs</b>	A	5.2.0	S5-036118	OAM-NIM

## CHANGE REQUEST

⌘ **32.102 CR 026** ⌘ rev **-** ⌘ Current version: **4.2.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Add New Subclause to IS Template for Notification Related IOCs		
<b>Source:</b>	⌘ S5		
<b>Work item code:</b>	⌘ OAM-CM	<b>Date:</b>	⌘ 28/02/2003
<b>Category:</b>	⌘ <b>F</b>	<b>Release:</b>	⌘ Rel-4
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
	<b>F</b> (correction)		2 (GSM Phase 2)
	<b>A</b> (corresponds to a correction in an earlier release)		R96 (Release 1996)
	<b>B</b> (addition of feature),		R97 (Release 1997)
	<b>C</b> (functional modification of feature)		R98 (Release 1998)
	<b>D</b> (editorial modification)		R99 (Release 1999)
	Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		Rel-4 (Release 4)
			Rel-5 (Release 5)
			Rel-6 (Release 6)

<b>Reason for change:</b>	⌘ The IS template needs to include an additional table for IOCs to identify which notifications apply to an IOC with a common understanding of the semantics to be used.
<b>Summary of change:</b>	⌘ Add new sub-clause X.3.a.6 containing table identifying which notifications apply to an IOC. Define the symatics and usage of the table.
<b>Consequences if not approved:</b>	⌘ The IS specifications do not clearly and consistently identify which notifications apply to IOCs and the semantics that apply.

<b>Clauses affected:</b>	⌘ Annex C										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;">X</td> <td style="text-align: center;"> </td> </tr> </table>	Y	N		X		X	X		Other core specifications	⌘
Y	N										
	X										
	X										
X											
		Test specifications									
		O&M Specifications	32.622, 32.632, 32.642, 32.652.								
<b>Other comments:</b>	⌘										

### How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.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

## Change in Clause Annex C

### Annex C (informative): Information Service template

This annex contains the template to be used for the Information Services documents produced within the 3GPP SA TSG5. This template is based on the latest 3GPP template which **must** be used for any 3GPP Technical Specification.

The introductory clauses of the 3GPP template (from clause 1 to clause 3) are unchanged.

This template is numbered starting with “X” which, in general should correspond to 4 which is the beginning of the main text document. However, if there is a need for a specific IS to introduce additional clauses in the body X may correspond to a number higher than 4. For an NRM only clause X shall be used.

The conclusive clauses/annexes of the 3GPP template are unchanged.

## X Information Object Classes

-- ‘X’ represents a number

### X.1 Information entities imported and local labels

-- this clause identifies a list of information entities (e.g. information object class, information relationship, information attribute) that have been defined in other specifications and that are imported in this specification. This includes information entities from other specifications imported for inheritance purpose. Each element of this list is a pair (label reference, local label). The label reference contains the name of the specification where it is defined, the type of the information entity and its name. The local label of imported information entities can then be used throughout the specification instead of the label reference.

-- this information is provided in a table. An example of such a table is given here below :

Label reference	Local label
32.106-5 [10], information object class, Top	Top

### X.2 Class diagram

#### X.2.1 Attributes and relationships

-- this first diagram represents all information object classes defined in this IS with all their relationships and all their attributes. This diagram shall contain relationship names, role name and role cardinality. This shall be a UML compliant class diagram.

-- Characteristics (attributes, relationships) of imported information object classes need not to be repeated in the diagram. Names of information elements (class, attribute) defined in the IS and which scope is local to this IS must be prefixed by a 3 characters prefix uniquely identifying the IS. Information object classes should be defined using the stereotype <<InformationObjectClass>>. On the class diagram, each attribute in an information object class shall be qualified as “protected” by the addition of a symbol “#” before each attribute.

## X.2.2 Inheritance

-- this second diagram represents the inheritance hierarchy of all information object classes defined in this IS. This diagram does not need to contain the complete inheritance hierarchy but shall at least contain the parent information object classes of all information object classes defined in this specification. By default, an information object class inherits from the information object class "top". This shall be a UML compliant class diagram.

-- Characteristics (attributes, relationships) of imported information object classes need not to be repeated in the diagram. Information object classes should be defined using the stereotype <<InformationObjectClass>>.

-- Note : some inheritance relationships presented in X.2.2 can be repeated in X.2.1 to enhance readability.

## X.3 Information object classes definition

-- each information object class is defined using the following structure :

### X.3.a InformationObjectClassName

-- InformationObjectClassName is the name of the information object class

-- 'a' represents a number, starting at 1 and increasing by 1 with each new definition of an information object class

#### X.3.a.1 Definition

-- The <definition> sub-clause is written in natural language. The <definition> sub-clause refers to the information object class itself. The characteristics related to the relationships that the object class can have with other object classes can't be found in the definition. The reader has to refer to relationships definition to find such kind of information. Information related to inheritance shall be precised here.

#### X.3.a.2 Attributes

-- The <attributes> sub-clause presents the list of attributes, which are the manageable properties of the object class . Each element is a pair (attributeName, supportQualifier).The supportQualifier indicates whether the attribute is Mandatory, Optional or Conditional (M, O, C).

-- this information is provided in a table. An example of such a table is given here below :

Attribute name	Support Qualifier
ntfSubscriptionId	M

- Note : this sub-clause does not need to be present when there is no attribute to define.

#### X.3.a.3 Attribute constraints

-- The <attribute constraints> sub-clause presents constraints between attributes that are always held to be true. ). Those properties are always held to be true during the lifetime of the attributes and in particular don't need to be repeated in pre or post conditions of operations or notifications.

- Note : this sub-clause does not need to be present when there is no attribute constraints to define.

#### X.3.a.4 Relationships

-- The <relationship> sub-clause presents the list of relationships in which this class is involved. Each element is a relationshipName.

- Note : this sub-clause is optional and may be avoided since all relationships are represented in the class diagram in clause.X.2.1.

### X.3.a.5 State diagram

-- The <state diagram> sub-clause contains state diagrams. A state diagram of an information object class defines permitted states of this information object class and the transitions between those states. A state is expressed in terms of individual attribute values or a combination of attribute values or involvement in relationships of the information object class being defined. This shall be a UML compliant state diagram.

### X.3.a.6 Notifications

The <notifications> sub-clause presents the list of notifications that can be emitted across the Itf-N, with "object class" and "object instance" parameters of the notification header of these notifications identifying an instance of the IOC defined by the encapsulating sub-clause (i.e. X.3.a). The presence of notifications in the present sub-clause (i.e. X.3.a.6) does not imply nor identify those notifications as being originated from an instance of the IOC defined by the encapsulating sub-clause (i.e. X.3.a).

This information is provided in a table. An example of such a table is given below:

<b>Name</b>	<b>Qualifier</b>	<b>Notes</b>
<a href="#">notifyAckStateChanged</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyAttributeValueChange</a>	O	
<a href="#">notifyChangedAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyClearedAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyNewAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyObjectCreation</a>	O	
<a href="#">notifyObjectDeletion</a>	O	
...	...	

## X.4 Information relationships definition

-- each information relationship is defined using the following structure :

### X.4.a InformationRelationshipName (supportQualifier)

-- InformationRelationshipName is the name of the information relationship followed by a qualifier indicating whether the relationship is Mandatory, Optional or Conditional (M, O, C)

-- 'a' represents a number, starting at 1 and increasing by 1 with each new definition of an information relationship

#### X.4.a.1 Definition

-- The <definition> sub-clause is written in natural language.

#### X.4.a.2 Roles

-- The <roles> sub-clause identifies the roles played in the relationship by object classes.. Each element is a pair (roleName, roleDefinition)

-- this information is provided in a table. An example of such a table is given here below :

<b>Name</b>	<b>Definition</b>
isSubscribedBy	This role represents the one who has subscribed

### X.4.a.3 Constraints

-- The <constraints> sub-clause contains the list of properties specifying the semantic invariants that must be preserved on the relationship. Each element is a pair (propertyName, propertyDefinition). Those properties are always held to be true during the lifetime of the relationship and don't need to be repeated in pre or post conditions of operations or notifications.

-- this information is provided in a table. An example of such a table is given here below :

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

## X.5 Information attributes definition

-- each information attribute is defined using the following structure :

### X.5.1 Definition and legal values

-- This sub-clause contains for each attribute being defined its name, its definition written in natural language and a list of legal values supported by the attribute.

-- In the case where the legal values can be enumerated, each element is a pair (legalValueName, legalValueDefinition), unless a legalValueDefinition applies to several values in which case the definition is provided only once. When the legal values cannot be enumerated, the list of legal values is defined by a single definition.

-- this information is provided in a table. An example of such a table is given here below :

Attribute Name	Definition	Legal Values
ntfSubscriptionId	It identifies uniquely a subscription	N/A
ntfSubscriptionState	It indicates the activation state of a subscription	“suspended” : the subscription is suspended “notSuspended” : the subscription is active

### X.5.2 Constraints

-- The <constraints> sub-clause indicates whether there are any constraints affecting attributes. Each constraint is defined by a pair (propertyName, propertyDefinition). PropertyDefinitions are expressed in natural language.

-- An example is given here below :

Name	Definition
inv_TimerConstraints	“ntfTimeTickTimer is lower than or equal to ntfTimeTick”

## X.6 Particular information configurations

-- some configurations of information are special or complex enough to justify the usage of a state diagram to clarify them. A state diagram in this clause defines permitted states of the system and the transitions between those states. A state is expressed in terms of a combination of attribute values constraints or involvement in relationships of one or more information object classes.

---

## Y Interface Definition

-- 'Y' represents a number, immediately following 'X'

### Y.1 Class diagram representing interfaces

-- each interface is defined in the diagram. This shall be a UML compliant class diagram.

-- Interfaces are defined using a stereotype <<Interface>>. Each interface contains a set of either operations or notifications which are mandatory or either a single operation or a single notification which is optional. The support of an interface by an information object class is represented by a relationship between the 2 entities with a cardinality (1..1) if all the operations or notifications contained in the interface are mandatory, and (0..1) if the operation or notification contained in the interface is optional. On the class diagram, each operation and notification in an interface shall be qualified as "public" by the addition of a symbol "+" before each operation and notification.

### Y.2 Generic rules

-- the following rules are relevant for all IS. They shall simply be copied as part of the template.

- 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.

### Y.b InterfaceName Interface

-- InterfaceName is the name of the interface

-- 'b' represents a number, starting at 3 and increasing by 1 with each new definition of an interface

-- Each interface is defined by its name and by a sequence of operations or notifications as defined herebelow.

-- each operation is defined using the following structure :

#### Y.b.a Operation OperationName (supportQualifier)

-- OperationName is the name of the operation followed by a qualifier indicating whether the operation is Mandatory, Optional or Conditional (M, O, C)

-- 'a' represents a number, starting at 1 and increasing by 1 with each new definition of an operation

##### Y.b.a.1 Definition

-- The <definition> sub-clause is written in natural language.

### Y.b.a.2 Input parameters

-- list of input parameters of the operation. Each element is a tuple (inputParameterName, supportQualifier, InformationType, inputParameterComment)

-- this information is provided in a table. An example of such a table is given here below :

Parameter Name	Qualifier	Information type	Comment
managerReference	M	ntfSubscriber.ntfManagerReference	It specifies the reference of IRPManager to which notifications shall be sent.

### Y.b.a.3 Output parameters

-- list of output parameters of the operation. Each element is a tuple (outputParameterName, supportQualifier, MatchingInformation, outputParameterComment)

-- this information is provided in a table. An example of such a table is given here below :

Parameter Name	Qualifier	Matching Information	Comment
versionNumberSet	M	notificationIRP.irpversion	It indicates one or more SS version numbers supported by the notificationIRP.

### Y.b.a.4 Pre-condition

-- a pre-condition is a collection of assertions joined by AND, OR, and NOT logical operators. The pre-condition must be held to be true before the operation is invoked .. An example is given here below :

notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed

-- Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the pre-condition are provided in a table. An example of such a table is given here below :

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 ntfSubscription which is involved in a subscription relationship with the ntfSubscriber identified by the managerReference input parameter”

### Y.b.a.5 Post-condition

-- a post-condition is a collection of assertions joined by AND, OR, and NOT logical operators. The post-condition must be held to be true after the completion of the operation. When nothing is said in a post-condition regarding an information entity, the assumption is that this information entity has not changed compared to what is stated in the pre-condition. An example is given here below :



subscriptionDeleted OR allSubscriptionDeleted

-- Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the post-condition are provided in a table. An example of such a table is given here below :

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.”

### Y.b.a.6 Exceptions

-- list of exceptions that can be raised by the operation. Each element is a tuple (exceptionName, condition, ReturnedInformation, exitState))

#### Y.b.a.6.c exceptionName

-- exceptionName is the name of an exception

-- ‘c’ represents a number, starting at 1 and increasing by 1 with each new definition of an exception

-- this information is provided in a table. An example of such a table is given here below :

Exception Name	Definition
Ope_failed_existing_subscription	<p><b>Condition:</b> (notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed) not verified</p> <p><b>Returned information:</b> output parameter status is set to OperationFailedExistingSubscription</p> <p><b>Exit state:</b> Entry State</p>

-- each notification is defined using the following structure :

### Y.b.a Notification NotificationName (supportQualifier)

-- NotificationName is the name of the notification followed by a qualifier indicating whether the notification is Mandatory, Optional or Conditional (M, O, C).

-- ‘a’ represents a number, starting at 1 and increasing by 1 with each new definition of a notification

#### Y.b.a.1 Definition

-- The <definition> sub-clause is written in natural language.

#### Y.b.a.2 Input parameters

-- list of input parameters of the notification. Each element is a tuple (inputParameterName, supportQualifier and filteringQualifier, matchingInformation, inputParameterComment)

-- the filteringQualifier indicates whether the parameter of the notification can be filtered or not. Values are Yes (Y) or No (N). The matchingInformation refers to information in the state “toState”.

-- this information is provided in a table. The column "Qualifiers" contains the two qualifiers supportQualifier and filteringQualifier separated by a comma. An example of such a table is given here below :

Parameter Name	Qualifiers	Matching Information	Comment
managerReference	M,Y	ntfSubscriber.ntfManagerReference	It specifies the reference of IRPManager to which notifications shall be sent.

### Y.b.a.3 Triggering event

-- the triggering event for the notification to be sent is the transition from the information state defined by the 'from state' sub-clause to the information state defined by the 'to state' sub-clause.

#### Y.b.a.3.1 From state

-- this sub-clause is a collection of assertions joined by AND, OR, and NOT logical operators. An example is given herebelow :

alarmMatched AND alarmInformationNotCleared

-- Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the state "from state" are provided in a table. An example of such a table is given here below :

Assertion Name	Definition
alarmMatched	The newly generated network alarm matches with one AlarmInformation (same values for eventType, probableCause, specificProblem attributes) in AlarmList.
alarmInformationNotCleared	The perceivedSeverity attribute of the matched AlarmInformation is not cleared

#### Y.b.a.3.2 To state

-- this sub-clause is a collection of assertions joined by AND, OR and NOT logical operators. When nothing is said in a to-state regarding an information entity, the assumption is that this information entity has not changed compared to what is stated in the from state. An example is given here below :

resetAcknowledgementInformation AND perceivedSeverityUpdated

-- Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the state "to state" are provided in a table. An example of such a table is given here below :

Assertion Name	Definition
resetAcknowledgementInformation	The matched AlarmInformation identified in inv_alarmMatched in pre-condition has been updated according to the following rule :  ackTime, ackUserId and ackSystemId are updated to contain no information; ackState is updated to "unacknowledged";
perceivedSeverityUpdated	The perceivedSeverity attribute of matched AlarmInformation identified in inv_alarmMatched in pre-condition has been updated.

## Z Scenario

-- 'Z' represents a number, immediately following 'Y'

*-- list of sequence diagrams each describing a possible scenario. This shall be a UML compliant sequence diagram. This is an optional clause.*

**End of Change in Annex C  
End of Document**

## CHANGE REQUEST

⌘ **32.102 CR 027** ⌘ rev **-** ⌘ Current version: **5.2.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Add New Subclause to IS Template for Notification Related IOCs		
<b>Source:</b>	⌘ S5		
<b>Work item code:</b>	⌘ OAM-NIM	<b>Date:</b>	⌘ 28/02/2003
<b>Category:</b>	⌘ <b>A</b>	<b>Release:</b>	⌘ Rel-5
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
	<b>F</b> (correction)	2	(GSM Phase 2)
	<b>A</b> (corresponds to a correction in an earlier release)	R96	(Release 1996)
	<b>B</b> (addition of feature),	R97	(Release 1997)
	<b>C</b> (functional modification of feature)	R98	(Release 1998)
	<b>D</b> (editorial modification)	R99	(Release 1999)
	Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .	Rel-4	(Release 4)
		Rel-5	(Release 5)
		Rel-6	(Release 6)

<b>Reason for change:</b>	⌘ The IS template needs to include an additional table for IOCs to identify which notifications apply to an IOC with a common understanding of the semantics to be used.
<b>Summary of change:</b>	⌘ Add new sub-clause X.3.a.6 containing table identifying which notifications apply to an IOC. Define the symantics and useage of the table.
<b>Consequences if not approved:</b>	⌘ The IS specifications do not clearly and consistently identify which notifications apply to IOCs and the semantics that apply.

<b>Clauses affected:</b>	⌘ Annex C										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Y</td> <td style="padding: 2px;">N</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">X</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">X</td> </tr> <tr> <td style="padding: 2px;">X</td> <td style="padding: 2px;"></td> </tr> </table>	Y	N		X		X	X		Other core specifications	⌘ 32.622, 32.632, 32.642, 32.652.
	Y	N									
		X									
		X									
X											
	Test specifications										
	O&M Specifications										
<b>Other comments:</b>	⌘										

### How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.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

## Change in Clause Annex C

### Annex C (informative): Information Service template

This annex contains the template to be used for the Information Services documents produced within the 3GPP SA TSG5. This template is based on the latest 3GPP template which **must** be used for any 3GPP Technical Specification.

The introductory clauses of the 3GPP template (from clause 1 to clause 3) are unchanged.

This template is numbered starting with "X" which, in general should correspond to 4 which is the beginning of the main text document. However, if there is a need for a specific IS to introduce additional clauses in the body X may correspond to a number higher than 4. For an NRM only clause X shall be used.

The conclusive clauses/annexes of the 3GPP template are unchanged.

## X Information Object Classes

*"X" represents a number*

### X.1 Information entities imported and local labels

*This clause identifies a list of information entities (e.g. information object class, information relationship, information attribute) that have been defined in other specifications and that are imported in the present document. This includes information entities from other specifications imported for inheritance purpose. Each element of this list is a pair (label reference, local label). The label reference contains the name of the specification where it is defined, the type of the information entity and its name. The local label of imported information entities can then be used throughout the specification instead of the label reference.*

*This information is provided in a table. An example of such a table is given here below :*

Label reference	Local label
32.106-5 [10], information object class, Top	Top

### X.2 Class diagram

#### X.2.1 Attributes and relationships

*This first diagram represents all information object classes defined in this IS with all their relationships and all their attributes. This diagram shall contain relationship names, role name and role cardinality. This shall be a UML compliant class diagram.*

*Characteristics (attributes, relationships) of imported information object classes need not to be repeated in the diagram. Names of information elements (class, attribute) defined in the IS and which scope is local to this IS must be prefixed by a 3 characters prefix uniquely identifying the IS. Information object classes should be defined using the stereotype <<InformationObjectClass>>. On the class diagram, each attribute in an information object class shall be qualified as "protected" by the addition of a symbol "#" before each attribute.*

## X.2.2 Inheritance

*This second diagram represents the inheritance hierarchy of all information object classes defined in this IS. This diagram does not need to contain the complete inheritance hierarchy but shall at least contain the parent information object classes of all information object classes defined in the present document. By default, an information object class inherits from the information object class "top". This shall be a UML compliant class diagram.*

*Characteristics (attributes, relationships) of imported information object classes need not to be repeated in the diagram. Information object classes should be defined using the stereotype <<InformationObjectClass>>.*

*NOTE: some inheritance relationships presented in X.2.2 can be repeated in X.2.1 to enhance readability.*

## X.3 Information object classes definition

*Each information object class is defined using the following structure.*

### X.3.a InformationObjectClassName

*InformationObjectClassName is the name of the information object class*

*"a" represents a number, starting at 1 and increasing by 1 with each new definition of an information object class*

#### X.3.a.1 Definition

*The <definition> sub-clause is written in natural language. The <definition> sub-clause refers to the information object class itself. The characteristics related to the relationships that the object class can have with other object classes can't be found in the definition. The reader has to refer to relationships definition to find such kind of information. Information related to inheritance shall be precised here.*

#### X.3.a.2 Attributes

*The <attributes> sub-clause presents the list of attributes, which are the manageable properties of the object class . Each element is a tuple (attributeName, visibilityQualifier, supportQualifier, readQualifier, writeQualifier)*

- *The visibilityQualifier indicates whether the attribute is public, private or IRPAgent Internal ("+", "—", and "%") respectively). The semantics of public and private are as per the UML specification. The semantic of IRPAgent Internal is defined within the 3GPP UML Repertoire.*
- *The supportQualifier indicates whether the attribute is Mandatory, Optional, Conditional or not supported ("M", "O", "C", or "—", respectively).*
- *The readQualifier indicates whether the attribute shall be readable by the IRPManager. The semantics for readQualifier is identical to supportQualifier, for "M", "O", and "—".*
- *The writeQualifier indicates whether the attribute shall be writeable by the IRPManager. The semantics for writeQualifier is identical to supportQualifier, for "M", "O", and "—".*

*There is a dependency relationship between the supportQualifier and visibilityQualifier, readQualifier, and writeQualifier. The supportQualifier indicates the requirements for the support of the attribute. For any given attribute, regardless of the value of the supportQualifier, at least one of the reqdQualifier or writeQualifier must be "M". The implication of the "O" supportQualifier is that the attribute is optional, however the read and write qualifiers indicate how the optional attribute shall be supported, should the optional attribute be supported. Regardless of the supportQualifier, if an attribute is supported then it shall be supported in accordance with the specified visibilityQualifier.*

*Private or IRPAgent Internal attributes are per definition not readable by the IRPManager. Their readQualifier is hence always "—".*

*Private or IRPAgent Internal attributes are per definition not writable by the IRPManager. Their writeQualifier is hence always "—".*

*The readQualifier and writeQualifier of a supported attribute, that is public, may not be both "—".*

The use of "—" in supportQualifier is reserved for documenting support of attributes defined by an «Archetype» IOC. Attributes with a supportQualifier of "—" are not implemented by the IOC that is realizing a subset of the attributes defined by the «Archetype». The readQualifier and writeQualifier are of no relevance in this case. However, a not supported attribute is neither readable nor writable. For this reason the readQualifier and writeQualifier shall be "—" for unsupported attributes.

For any IOC that uses one or more attributes from an «Archetype», a separate table shall be used to indicate the supported attributes. This table is absent if no «Archetype» attributes are supported. For example, if a particular IOC has defined attributes (i.e. attributes not defined by an «Archetype») and encapsulates attributes from two «Archetype»s, then the totality of the attributes of said IOC will be contained in three separate tables.

This information is provided in a table. An example of such a table is given below:

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
ntfSubscriptionId	+	M	M	O

Another example, where the support qualifier is "O" is given here below:

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
ntfSubscriptionId	+	O	M	O

In this example, the ntfSubscriptionId is an optional attribute. If the implementation chose to support ntfSubscriptionId, then the said implementation is required to support read and may support write.

**NOTE:** This sub-clause does not need to be present when there is no attribute to define.

### X.3.a.3 Attribute constraints

The <attribute constraints> sub-clause presents constraints between attributes that are always held to be true. Those properties are always held to be true during the lifetime of the attributes and in particular don't need to be repeated in pre or post conditions of operations or notifications.

**NOTE:** This sub-clause does not need to be present when there is no attribute constraints to define.

### X.3.a.4 Relationships

The <relationship> sub-clause presents the list of relationships in which this class is involved. Each element is a relationshipName.

**NOTE:** This sub-clause is optional and may be avoided since all relationships are represented in the class diagram in clause.X.2.1.

### X.3.a.5 State diagram

The <state diagram> sub-clause contains state diagrams. A state diagram of an information object class defines permitted states of this information object class and the transitions between those states. A state is expressed in terms of individual attribute values or a combination of attribute values or involvement in relationships of the information object class being defined. This shall be a UML compliant state diagram.

### X.3.a.6 Notifications

The <notifications> sub-clause presents the list of notifications that can be emitted across the Itf-N, with "object class" and "object instance" parameters of the notification header of these notifications identifying an instance of the IOC defined by the encapsulating sub-clause (i.e. X.3.a). The presence of notifications in the present sub-clause (i.e. X.3.a.6) does not imply nor identify those notifications as being originated from an instance of the IOC defined by the encapsulating sub-clause (i.e. X.3.a).

This information is provided in a table. An example of such a table is given below:

Name	Qualifier	Notes
<a href="#">notifyAckStateChanged</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyAttributeValueChange</a>	<a href="#">O</a>	
<a href="#">notifyChangedAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyClearedAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyNewAlarm</a>	<a href="#">See Alarm IRP (3GPP TS 32.111-2 [:])</a>	
<a href="#">notifyObjectCreation</a>	<a href="#">O</a>	
<a href="#">notifyObjectDeletion</a>	<a href="#">O</a>	
...	...	

## X.4 Information relationships definition

Each information relationship is defined using the following structure :

### X.4.a InformationRelationshipName (supportQualifier)

*InformationRelationshipName* is the name of the information relationship followed by a qualifier indicating whether the relationship is Mandatory, Optional or Conditional (M, O, C)

"a" represents a number, starting at 1 and increasing by 1 with each new definition of an information relationship

#### X.4.a.1 Definition

The <definition> sub-clause is written in natural language.

#### X.4.a.2 Roles

The <roles> sub-clause identifies the roles played in the relationship by object classes. Each element is a pair (roleName, roleDefinition)

This information is provided in a table. An example of such a table is given here below :

Name	Definition
isSubscribedBy	This role represents the one who has subscribed

#### X.4.a.3 Constraints

The <constraints> sub-clause contains the list of properties specifying the semantic invariants that must be preserved on the relationship. Each element is a pair (propertyName, propertyDefinition). Those properties are always held to be true during the lifetime of the relationship and don't need to be repeated in pre or post conditions of operations or notifications.

This information is provided in a table. An example of such a table is given here below :

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

## X.5 Information attributes definition

Each information attribute is defined using the following structure :



## X.5.1 Definition and legal values

*This sub-clause contains for each attribute being defined its name, its definition written in natural language and a list of legal values supported by the attribute.*

*In the case where the legal values can be enumerated, each element is a pair (legalValueName, legalValueDefinition), unless a legalValueDefinition applies to several values in which case the definition is provided only once. When the legal values cannot be enumerated, the list of legal values is defined by a single definition.*

*This information is provided in a table. An example of such a table is given here below :*

Attribute Name	Definition	Legal Values
ntfSubscriptionId	It identifies uniquely a subscription	N/A
ntfSubscriptionState	It indicates the activation state of a subscription	"suspended" : the subscription is suspended "notSuspended" : the subscription is active

## X.5.2 Constraints

*The <constraints> sub-clause indicates whether there are any constraints affecting attributes. Each constraint is defined by a pair (propertyName, propertyDefinition). PropertyDefinitions are expressed in natural language.*

*An example is given here below :*

Name	Definition
inv_TimerConstraints	"ntfTimeTickTimer is lower than or equal to ntfTimeTick"

## X.6 Particular information configurations

*Some configurations of information are special or complex enough to justify the usage of a state diagram to clarify them. A state diagram in this clause defines permitted states of the system and the transitions between those states. A state is expressed in terms of a combination of attribute values constraints or involvement in relationships of one or more information object classes.*

---

## Y Interface Definition

*"Y" represents a number, immediately following "X"*

### Y.1 Class diagram representing interfaces

*Each interface is defined in the diagram. This shall be a UML compliant class diagram.*

*Interfaces are defined using a stereotype <<Interface>>. Each interface contains a set of either operations or notifications which are mandatory or either a single operation or a single notification which is optional. The support of an interface by an information object class is represented by a relationship between the 2 entities with a cardinality (1..1) if all the operations or notifications contained in the interface are mandatory, and (0..1) if the operation or notification contained in the interface is optional. On the class diagram, each operation and notification in an interface shall be qualified as "public" by the addition of a symbol "+" before each operation and notification.*

### Y.2 Generic rules

*The following rules are relevant for all IS. They shall simply be copied as part of the template.*

*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.

## Y.b InterfaceName Interface

*InterfaceName* is the name of the interface

"b" represents a number, starting at 3 and increasing by 1 with each new definition of an interface

Each interface is defined by its name and by a sequence of operations or notifications as defined here below.

Each operation is defined using the following structure.

### Y.b.a Operation OperationName (supportQualifier)

*OperationName* is the name of the operation followed by a qualifier indicating whether the operation is Mandatory, Optional or Conditional (M, O, C)

"a" represents a number, starting at 1 and increasing by 1 with each new definition of an operation

#### Y.b.a.1 Definition

The <definition> sub-clause is written in natural language.

#### Y.b.a.2 Input parameters

List of input parameters of the operation. Each element is a tuple (*inputParameterName*, *supportQualifier*, *InformationType*, *inputParameterComment*)

This information is provided in a table. An example of such a table is given here below :

Parameter Name	Qualifier	Information type	Comment
managerReference	M	ntfSubscriber.ntfManagerReference	It specifies the reference of IRPManager to which notifications shall be sent.

#### Y.b.a.3 Output parameters

List of output parameters of the operation. Each element is a tuple (*outputParameterName*, *supportQualifier*, *MatchingInformation*, *outputParameterComment*)

This information is provided in a table. An example of such a table is given here below :

Parameter Name	Qualifier	Matching Information	Comment
versionNumberSet	M	notificationIRP.irpversion	It indicates one or more SS version numbers supported by the notificationIRP.

#### Y.b.a.4 Pre-condition

A pre-condition is a collection of assertions joined by AND, OR, and NOT logical operators. The pre-condition must be held to be true before the operation is invoked . An example is given here below :

*notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed*

Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the pre-condition are provided in a table. An example of such a table is given here below :

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 ntfSubscription which is involved in a subscription relationship with the ntfSubscriber identified by the managerReference input parameter"

#### Y.b.a.5 Post-condition

A post-condition is a collection of assertions joined by AND, OR, and NOT logical operators. The post-condition must be held to be true after the completion of the operation. When nothing is said in a post-condition regarding an information entity, the assumption is that this information entity has not changed compared to what is stated in the pre-condition. An example is given here below :

*subscriptionDeleted OR allSubscriptionDeleted*

Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the post-condition are provided in a table. An example of such a table is given here below :

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."

#### Y.b.a.6 Exceptions

List of exceptions that can be raised by the operation. Each element is a tuple (exceptionName, condition, ReturnedInformation, exitState))

##### Y.b.a.6.c exceptionName

ExceptionName is the name of an exception

"c" represents a number, starting at 1 and increasing by 1 with each new definition of an exception

This information is provided in a table. An example of such a table is given here below :

Exception Name	Definition
Ope_failed_existing_subscription	<b>Condition:</b> (notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed) not verified <b>Returned information:</b> output parameter status is set to OperationFailedExistingSubscription <b>Exit state:</b> Entry State

Each notification is defined using the following structure.

## Y.b.a Notification NotificationName (supportQualifier)

*NotificationName is the name of the notification followed by a qualifier indicating whether the notification is Mandatory, Optional or Conditional (M, O, C).*

*"a" represents a number, starting at 1 and increasing by 1 with each new definition of a notification*

### Y.b.a.1 Definition

*The <definition> sub-clause is written in natural language.*

### Y.b.a.2 Input parameters

*List of input parameters of the notification. Each element is a tuple (inputParameterName, supportQualifier and filteringQualifier, matchingInformation, inputParameterComment)*

*The filteringQualifier indicates whether the parameter of the notification can be filtered or not. Values are Yes (Y) or No (N). The matchingInformation refers to information in the state "toState".*

*This information is provided in a table. The column "Qualifiers" contains the two qualifiers supportQualifier and filteringQualifier separated by a comma. An example of such a table is given here below :*

Parameter Name	Qualifiers	Matching Information	Comment
managerReference	M,Y	ntfSubscriber.ntfManagerReference	It specifies the reference of IRPManager to which notifications shall be sent.

### Y.b.a.3 Triggering event

*The triggering event for the notification to be sent is the transition from the information state defined by the "from state" sub-clause to the information state defined by the "to state" sub-clause.*

#### Y.b.a.3.1 From state

*This sub-clause is a collection of assertions joined by AND, OR, and NOT logical operators. An example is given here below :*

*alarmMatched AND alarmInformationNotCleared*

*Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the state "from state" are provided in a table. An example of such a table is given here below :*

Assertion Name	Definition
alarmMatched	The newly generated network alarm matches with one AlarmInformation (same values for eventType, probableCause, specificProblem attributes) in AlarmList.
alarmInformationNotCleared	The perceivedSeverity attribute of the matched AlarmInformation is not cleared

#### Y.b.a.3.2 To state

*This sub-clause is a collection of assertions joined by AND, OR and NOT logical operators. When nothing is said in a to-state regarding an information entity, the assumption is that this information entity has not changed compared to what is stated in the from state. An example is given here below :*

*resetAcknowledgementInformation AND perceivedSeverityUpdated*

*Each assertion is defined by a pair (propertyName, propertyDefinition). All assertions constituting the state "to state" are provided in a table. An example of such a table is given here below :*

Assertion Name	Definition
resetAcknowledgementInformation	The matched AlarmInformation identified in inv_alarmMatched in pre-condition has been updated according to the following rule : ackTime, ackUserId and ackSystemId are updated to contain no information; ackState is updated to "unacknowledged";
perceivedSeverityUpdated	The perceivedSeverity attribute of matched AlarmInformation identified in inv_alarmMatched in pre-condition has been updated.

---

## Z Scenario

*"Z" represents a number, immediately following "Y"*

*List of sequence diagrams each describing a possible scenario. This shall be a UML compliant sequence diagram. This is an optional clause.*

<b>End of Change in Annex C</b> <b>End of Document</b>
---