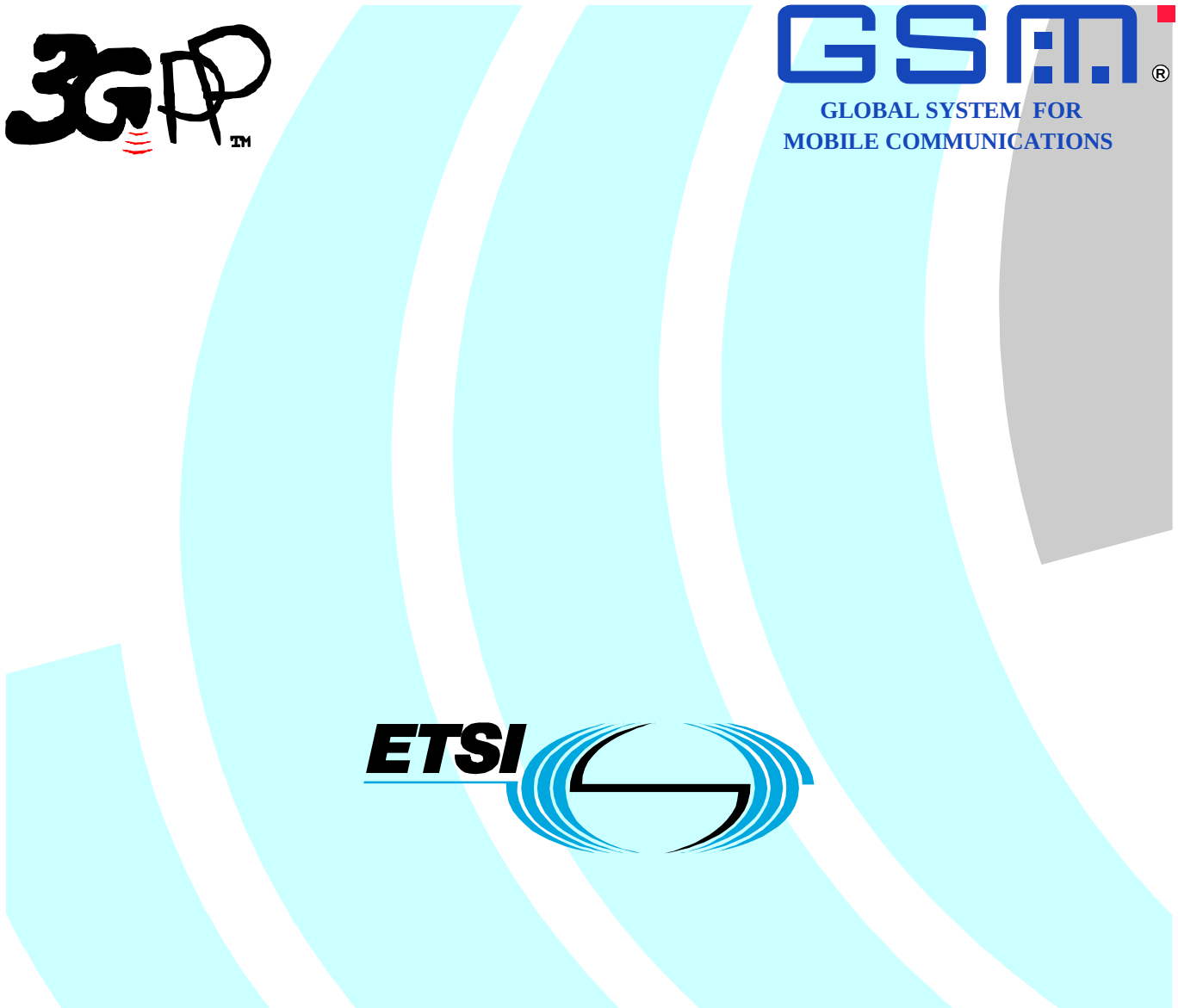
ETSI TS 132 151 V6.0.0 (2004-03)

Technical Specification

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
Telecommunication management;
Integration Reference Point (IRP)
Information Service (IS) template
(3GPP TS 32.151 version 6.0.0 Release 6)



Reference
DTS/TSGS-0532151v600

Keywords
GSM, 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://webapp.etsi.org/IPR/home.asp).

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

Foreword

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

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

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp .

Contents

Foreword

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

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.

1 Scope

The present document contains the template to be used for the production of all Integration Reference Point (IRP) Information Service (IS) specifications within the 3GPP 32-series.

The present document is a member of a TS-family consisting of:

3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

3GPP TS 32.151: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template".

3GPP TS 32.152: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire".

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. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document.*

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[4] 3GPP TS 32.152: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire".

[5] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[6] 3GPP TS 32.111-2: "Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.150 [3] and the following apply:

IRPAgent: See 3GPP TS 32.102 [2].

IRPManager: See 3GPP TS 32.102 [2].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.150 [3] and the following apply:

IOC	Information Object Class
IRP	Integration Reference Point
IS	Information Service
OMG	Object Management Group
UML	Unified Modelling Language (OMG)

4 Information Service (IS) template

The present document contains the template to be used for the production of all Integration Reference Point (IRP) Information Service (IS) specifications within the 3GPP 32-series.

The clauses in this template that shall be used in the IS specifications are numbered starting with "X", which in general should correspond to clause 4 that is the beginning of the main part of the TS. However, if there is a need in a specific IS to introduce additional clauses in the body, X may correspond to a number higher than 4. For a Network Resource Model (NRM) IRP IS, only clause X shall be used.

The introductory clauses (from clause 1 to clause 3) for the IS should be taken from the standard 3GPP TS template (i.e. not this IRP IS template).

Usage of fonts shall be according to the following table.

Item	Font
Class names	Courier
Attribute names	Courier
Operation names	Courier
Parameter names	Courier
Assertion names	Courier
Notification names	Courier
Exception names	Courier
State names	Arial
Enumerated values	Arial

X Information Object Classes

"X" represents a clause number in the actual Information Service TS.

X.1 Imported information entities 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
3GPP TS 32.622 [5], 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 (see also 3GPP TS 32.152 [4]).

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>>.

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 clause X.2.2 can be repeated in clause X.2.1 to enhance readability.

X.3 Information object class definitions

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> subclause is written in natural language. The <definition> subclause 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> subclause 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 (3GPP TS 32.152 [4]).*

- *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 readQualifier 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 subclause does not need to be present when there is no attribute to define.

X.3.a.3 Attribute constraints

The <attribute constraints> subclause 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 subclause does not need to be present when there are no attribute constraints to define.

X.3.a.4 Relationships

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

NOTE: This subclause 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> subclause 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.

NOTE: This subclause does not need to be present when there is no state diagram to define.

X.3.a.6 Notifications

The <notifications> subclause 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 subclause (i.e. clause X.3.a). The presence of notifications in the present subclause (i.e. clause X.3.a.6) does not imply nor identify those notifications as being originated from an instance of the IOC defined by the encapsulating subclause (i.e. clause X.3.a).

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

Name	Qualifier	Notes
notifyAckStateChanged	See Alarm IRP (3GPP TS 32.111-2 [6])	
notifyAttributeValueChange	O	
notifyChangedAlarm	See Alarm IRP (3GPP TS 32.111-2 [6])	
notifyClearedAlarm	See Alarm IRP (3GPP TS 32.111-2 [6])	
notifyNewAlarm	See Alarm IRP (3GPP TS 32.111-2 [6])	
notifyObjectCreation	O	
notifyObjectDeletion	O	
…	…	

NOTE: This subclause does not need to be present when there is no notification to define.

X.4 Information relationship definitions

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> subclause is written in natural language.

X.4.a.2 Roles

The <roles> subclause 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> subclause 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 attribute definitions

Each information attribute is defined using the following structure:

X.5.1 Definition and legal values

This subclause 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
ntfSusbcriptionState	It indicates the activation state of a subscription	"suspended"*:* the subscription is suspended. "notSuspended"*:* the subscription is active.

X.5.2 Constraints

The <constraints> subclause 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	The 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 (see also 3GPP TS 32.152 [4]).

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> subclause 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	The 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.

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	**Condition:** (notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed) not verified. **Returned information:** output parameter status is set to OperationFailedExistingSubscription. **Exit state:** 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> subclause 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 column "Qualifiers" contains the two qualifiers, supportQualifier and filteringQualifier, separated by a comma. The supportQualifier indicates whether the attribute is Mandatory, Optional or Conditional ("M","O", or "C", respectively). 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. 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" subclause to the information state defined by the "to state" subclause.

Y.b.a.3.1 From state

This subclause 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 subclause 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".

This clause contains one or more sequence diagrams, each describing a possible scenario. This shall be a UML compliant sequence diagram. This is an optional clause.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Dec 2003	S_22	SP-030614	--	--	Submitted to TSG SA#22 for Information	1.0.0	
Mar 2004	S_23	SP-040114	--	--	Submitted to TSG SA#23 for Approval	2.0.0	6.0.0

History

Document history		
V6.0.0	March 2004	Publication