3GPP TS 32.336 V11.0.0 (2012-09)

**3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Telecommunication management;
Notification Log (NL) Integration Reference Point (IRP):
Solution Set (SS) definitions
(Release 11)**

Keywords

CORBA, XML, SOAP, management, architecture

3GPP

Postal address

3GPP   3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

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.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Notification Log Integration Reference Point (IRP), as identified below:

32.331 "Notification Log (NL) Integration Reference Point (IRP); Requirements".

32.332 "Notification Log (NL) Integration Reference Point (IRP); Information Service (IS)".

32.336 **"Notification Log (NL) Integration Reference Point (IRP); Solution Set (SS) definitions".**

The present document describes the requirements and information model necessary for Teleco mmunication Management (TM). The TM principles and TM architecture are specified in 3GPP TS 32.101 [2] and 3GPP TS 32.102 [3].

A communications system is composed of a multitude of Network Elements (NE) of various types and, typically, different vendors, which inter-operate in a co-ordinated manner in order to satisfy the network users' communication requirements.

The occurrence of faults in an NE may cause deterioration or loss of this NE's function. Fault Management is the functional area, which allows the operator to detect the occurrence of faults in the network in real-time. Configuration Management and Performance Management are two more functional areas, which require the operator to be alerted to certain conditions in the network.

A standard general-purpose mechanism for the management of logs containing selected notifications from the network is required to provide an ability to perform historical analysis on faults and conditions, which occurred in the network. The TS 32.33x-series, constituting the Notification log IRP, sets forth such a mechanism - and the present document contains the SS definition.

1 Scope

The present document specifies the Solution Sets for the IRP whose semantics is specified in Notification Log (NL) IRP IS (3GPP TS 32.332 [5]).

This Solution Set specification is related to 3GPP TS 32.332 V11.0.X.

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 TR 21.905: "Vocabulary for 3GPP Specifications".

[2]   3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

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

[4]   3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[5]   3GPP TS 32.332: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP); Information Service (IS)".

[6]   3GPP TS 32.342: "Telecommunication management; File Transfer (FT) Integration Reference Point (IRP); Information Service (IS)".

[7]   W3C SOAP 1.1 specification (http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)

[8]   W3C XPath 1.0 specification (http://www.w3.org/TR/1999/REC-xpath-19991116)

[9]   W3C WSDL 1.1 specification (http://www.w3.org/TR/2001/NOTE-wsdl-20010315)

[10]   W3C SOAP 1.2 specification (http://www.w3.org/TR/soap12-part1/)

[11]   3GPP TS 32.306: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Solution Set (SS) definitions".

[12]   OMG TC Document telecom/98-11-01: "OMG Notification Service".
        http://www.omg.org/technology/documents/

[13]   OMG CORBA Services: "Common Object Services Specification, Update: November 22, 1996" (Clause 4 contains the Event Service specification). http://www.omg.org/technology/documents/

[14]   3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirements".

[15]   3GPP TS 32.111-6: "Telecommunication management; Fault Management; Part 5: Alarm Integration Reference Point (IRP): Solution Set (SS) definitions".

[16]   3GPP TS 32.331: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Requirements".

[17]   3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

3   Definitions, symbols and abbreviations

3.1   Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1], 3GPP TS 32.101 [2], 3GPP TS 32.102 [3], 3GPP TS 32.150 [4], 3GPP TS 32.331 [16] and 3GPP TS 32.332 [5] apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

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.

IRP document version number string (or "IRPVersion"): See 3GPP TS 32.311 [14] subclause 3.1.

3.2   Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1], in 3GPP TS 32.101 [2], 3GPP TS 32.102 [3], 3GPP TS 32.150 [4], and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

CORBA   Common Object Request Broker Architecture
EM   Element Manager
FT   File Transfer
IDL   Interface Definition Language
IRP   Integration Reference Point
IS   Information Service
MOC   Managed Object Class
MOI   Managed Object Instance
NE   Network Element
NL   Notification Log
NLIRP   Notification Log Integration Reference Point
NM   Network Manager
NRM   Network Resource Model
UML   Unified Modelling Language
OMG   Object Management Group
XML   eXtensible Markup Language

4   Solution Set Definitions

This specification defines the following 3GPP NL IRP Solution Set Definitions:

Annex A provides the CORBA Solution Set.
Annex B provides the XML Definitions.
Annex C provides the SOAP Solution Set.

Annex A (normative): CORBA Solution Set

This annex contains the CORBA Solution Set for the IRP whose semantics is specified in NL IRP: Information Service (TS 32.332 [5]).

A.1 Architectural Features

The overall architectural feature of Notification Log IRP is specified in 3GPP TS 32.332 [5]. This clause specifies features that are specific to the CORBA SS.

A.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [17].

A.1.2 Notification Services

In implementations of CORBA SS, IRPAgent conveys Notification Log notifications to IRPManager via OMG Notification Service (OMG Notification Service [12]).

OMG Event Service [13] provides event routing and distribution capabilities. OMG Notification Service provides, in addition to Event Service, event filtering and Quality of Service (QoS).

A necessary and sufficient subset of OMG Notification Services shall be used to support Notification Log notifications as specified in 3GPP TS 32.332 [5].

These operation are classified as <<AgentInternal-usage>> in GPP TS 32.332 [5].

A.1.3 Push and Pull Style

OMG Notification Service defines two styles of interaction. One is called push style. In this style, IRPAgent pushes notifications to IRPManager as soon as they are available. The other is called pull style. In this style, IRPAgent keeps the notifications till IRPManager requests for them.

The Notification CORBA SS in 3GPP TS 32.306 [11] specifies that support of Push style is Mandatory (M) and that support of Pull style is Optional (O).

A.1.4 Support multiple notifications in one `push` operation

For efficiency reasons, IRPAgent may send multiple notifications using one single `push` operation. To pack multiple notifications into one `push` operation, IRPAgent may wait and not invoke the `push` operation as soon as notifications are available. To avoid IRPAgent to wait for an extended period of time that is objectionable to IRPManager, IRPAgent shall implement an IRPAgent wide timer configurable by administrator. On expiration of this timer, IRPAgent shall invoke `push` if there is at least one notification to be conveyed to IRPManager. This timer is re-started after each `push` invocation.

A.1.5 NotificationLogIRPNotifications Interface

OMG CORBA Notification push operation is used to realise the notification of `NotificationLogIRPNotifications`. All the notifications in this interface are implemented using this `push_structured_event` method.

A.1.5.1 Method push (M)

```
module CosNotifyComm {
…
Interface SequencePushConsumer : NotifyPublish {
             void push_structured_events(

                     in CosNotification::EventBatch notifications)

        raises ( CosEventComm::Disconnected);

            …

    }; // SequencePushConsumer

    …

}; // CosNotifyComm
```

NOTE 1: The push_structured_events method takes an input parameter of type EventBatch as defined in the OMG CosNotification module (OMG Notification Service [12]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

NOTE 2: The maximum number of events that will be transmitted within a single invocation of this operation is controlled by IRPAgent wide configuration parameter.

NOTE 3: The amount of time the supplier (IRPAgent) of a sequence of Structured Events will accumulate individual events into the sequence before invoking this operation is controlled by IRPAgent wide configuration parameter as well.

NOTE 4: IRPAgent may push EventBatch with only one Structured Event.

A.2 Mapping

A.2.1 Operation and Notification mapping

3GPP TS 32.332 [5] defines semantics of operations and notifications visible across the Notification Log IRP.
The following table indicates the mapping of these operations and notifications to their equivalents defined in this SS.

Table A.2.1: Mapping from IS Notification/Operation to SS equivalents

IS Operation/ notification 3GPP TS 32.332 [5]	SS Method	Qualifier
subscribeLog	subscribe_log	M
unsubscribeLog	unsubscribe_log	M
exportLogRecords	export_log_records	O
getLogRecords	get_log_records	O
getLogSubscriptionIds	get_log_subscription_ids	O
getLogSubscriptionStatus	get_log_subscription_status	O
notifyLogSubscribed	push_structured_event (note 1). See clause A.1.5.	M
notifyLogUnsubscribed	push_structured_event (note 1). See clause A.1.5.	M
notifyOccupancyLevelCrossed	push_structured_event (note 1). See clause A.1.5.	O
notifyLoggingResumed	push_structured_event (note 1). See clause A.1.5.	O
NOTE: OMG Notification Service OMG Notification Service [12] defines this method.		

A.2.2 Operation parameter mapping

3GPP TS 32.332 [5] defines semantics of parameters carried in operations across the Notification Log IRP.
The following tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table A.2.2.1: Mapping from IS `subscribeLog` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId	M
loggingEndTIme	GenericIRPManagementConstDefs::IRPTime loggingEndTimeAsked	O
notificationCategories	GenericIRPManagementConstDefs::NotificationCategorySet notificationCategorySet	O
filter	GenericIRPManagementConstDefs::StringOpt filter	O
logManagerToken	NotificationLogIRPConstDefs::LogManagerTokenOpt logManagerToken;	O
loggingEndTime	GenericIRPManagementConstDefs::loggingEndTimeGiven	O
status	GenericIRPManagementConstDefs::Signal Exceptions: SubscribeLog, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::ValueNotSupported, InvalidLogSubscriptionId, UnknownLogManagerToken.	M

Table A.2.2.2: Mapping from IS `unsubscribeLog` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId	M
logManagerToken	NotificationLogIRPConstDefs::LogManagerTokenOpt logManagerToken	O
status	GenericIRPManagementConstDefs::Signal Exceptions: UnsubscribeLog, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ParameterNotSupported, UnknownLogSubscriptionId, UnknownLogManagerToken).	M

Table A.2.2.3: Mapping from IS `exportLogRecords` **parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId	M
notificationCategories	NotificationLogIRPConstDefs::NotificationCategorySetOpt notificationCategorySet	O
filter	GenericIRPManagementConstDefs::StringOpt filter	O
invocationId	string invocationId	M
status	GenericIRPManagementConstDefs::Signal Exceptions: ExportLogRecords, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::OperationNotSupported, UnknownLogSubscriptionId.	M

Table A.2.2.4: Mapping from IS `getLogRecords` **parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId	M
notificationCategories	NotificationLogIRPConstDefs::NotificationCategorySetOpt notificationCategories,	O
filter	GenericIRPManagementConstDefs::StringOpt filter	O
getLogRecordsResult	Some or all of the information contained in getLogRecordsResult will be returned via the return value of type DsLogAdmin::RecordList.	M
status	Exceptions: GetLogRecords, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported, UnknownLogSubscriptionId.	M

Table A.2.2.5: Mapping from IS `getLogSubscriptionIds` **parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionIds	return value of type DsLogAdmin::LogIdList	M
status	Exceptions: GetLogSubscriptionIds, GenericIRPManagementSystem::OperationNotSupported.	M

Table A.2.2.6: Mapping from IS `getLogSubscriptionStatus` **parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId	M
logAttributeList	NotificationLogIRPConstDefs::LogAttributeList logAttributeList	M
status	GenericIRPManagementConstDefs::Signal Exceptions: GetLogSubscriptionStatus, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::OperationNotSupported.	M

A.2.3   Notification parameter mapping

3GPP TS 32.332 [5] defines semantics of parameters carried in notifications. The following tables indicate the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [12]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [12], is:

```
Header
      Fixed Header
            domain_name
            type_name
            event_name
      Variable Header
Body
      filterable_body_fields
      remaining_body
```

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the 3GPP TS 32.332 [5] defined notification parameters.

Table A.2.3.1: Mapping for `notifyLogSubscribed`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS parameter.	domain_name	M	It carries the IRP document version number string. See sub-clause 3.1. It indicates the syntax and semantics of the Structured Event as defined by this specification.
notificationType	type_name	M	This is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: NOTIFY_LOG_SUBSRIBED.
There is no corresponding IS parameter.	event_name	M	Null-string
There is no corresponding IS parameter.	variable Header		
objectClass, objectInstance	One NV (note 1) pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: MANAGED_OBJECT_INSTANCE. Value of NV pair is a string.
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: NOTIFICATION_ID. Value of NV pair is a long.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: EVENT_TIME. Value of NV pair is of type GenericIRPManagementConstDefs::IRPTime.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN Value of NV pair is a string.
logSubscriptionId	One NV pair of remaining_body	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed::LOG_SUBSCRIPTION_ID. Value of NV pair is of type NotificationLogIRPConstDefs::LogSubscriptionId.
loggingEndTime	One NV pair of remaining_body	O	Name of NV pair is the NotificationLogIRPNotifications::NotifyLogSubscribed::LOGGING_END_TIME. Value of NV pair is a string.
notificationCategories	One NV pair of remaining_body	O	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed::NOTIFICATION_CATEGORIES. Value of NV pair is of type GenericIRPManagementConstDefs::VersionNumberSet.
filter	One NV pair of remaining_body	O	Name of NV pair is notificationLogIRPNotifications:: NotifyLogSubscribed:: FILTER Value of NV pair is a string.
NOTE: NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. This note is applicable to all NV-pairs and all mapping tables for all notifications.			

Table A.2.3.2: Mapping for `notifyLogUnsubscribed`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS parameter.	domain_name	M	It carries the IRP document version number string. See sub-clause 3.1. It indicates the syntax and semantics of the Structured Event as defined by this specification.
notificationType	type_name	M	This is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: NOTIFY_LOG_UNSUBSCRIBED.
There is no corresponding IS parameter.	event_name	M	Null-string.
There is no corresponding IS parameter.	variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: MANAGED_OBJECT_INSTANCE. Value of NV pair is a string.
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: NOTIFICATION_ID. Value of NV pair is a long.
eventTime	One NV pair of One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: EVENT_TIME. Value of NV pair is a IRPTime of module GenericIRPManagementConstDefs.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN Value of NV pair is a string.
logSubscriptionId	One NV pair of filterable_body_fields (editor note)	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: LOG_SUBSCRIPTION_ID. Value of NV pair is of type NotificationLogIRPConstDefs::LogSubscriptionId.

Editor note: The placement of this parameter in filterable_body_fields so that it is filterable is not yet aligned with current IS and require further discussion.

Table A.2.3.3: Mapping for `notifyOccupancyLevelCrossed`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS parameter.	domain_name	M	It carries the IRP document version number string. See sub-clause 3.1. It indicates the syntax and semantics of the Structured Event as defined by this specification.
notificationType	type_name	M	This is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: NOTIFY_LOG_OCCUPANCY_LEVEL_CROSSED.
There is no corresponding IS parameter	event_name	M	Null-string
There is no corresponding IS parameter.	variable Header		
objectClass, objectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: MANAGED_OBJECT_INSTANCE. Value of NV pair is a string.
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: NOTIFICATION_ID. Value of NV pair is a long.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: EVENT_TIME. Value of NV pair is a IRPTime of module GenericIRPManagementConstDefs.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN Value of NV pair is a string.
logSubscriptionId	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: LOG_SUBSCRIPTION_ID. Value of NV pair is of type NotificationLogIRPConstDefs::LogSubscriptionId.
currentOccupancyLevel	One NV pair of One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: CURRENT_OCCUPANCY_LEVEL. Value of NV pair is an unsigned short (i.e. DsLogAdmin:: Threshold).
logFullAction	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: LOG_FULL_ACTION. Value of NV pair is of type NotificationLogIRPConstDefs:: LogFullActionType.

Table A.2.3.4: Mapping for `notifyLoggingResumed`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS parameter.	domain_name	M	It carries the IRP document version number string. See sub-clause 3.1. It indicates the syntax and semantics of the Structured Event as defined by this specification.
notificationType	type_name	M	This is the NotificationLogIRPNotifications:: NotifyLogSubscribed:: NOTIFY_LOG_SUBSRIBED.
There is no corresponding IS parameter.	event_name	M	Null-string
There is no corresponding IS parameter.	variable Header		
objectClass, objectInstance	One NV (note 1) pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: MANAGED_OBJECT_INSTANCE. Value of NV pair is a string.
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: NOTIFICATION_ID. Value of NV pair is a long.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: EVENT_TIME. Value of NV pair is of type GenericIRPManagementConstDefs::IRPTime.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN Value of NV pair is a string.
logSubscriptionId	One NV pair of remaining_body	M	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed::LOG_SUBSCRIPTION_ID. Value of NV pair is of type NotificationLogIRPConstDefs::LogSubscriptionId.
loggingEndTime	One NV pair of remaining_body	O	Name of NV pair is the NotificationLogIRPNotifications::NotifyLogSubscribed::LOGGING_END_TIME. Value of NV pair is a string.
notificationCategories	One NV pair of remaining_body	O	Name of NV pair is the NotificationLogIRPNotifications:: NotifyLogSubscribed::NOTIFICATION_CATEGORIES. Value of NV pair is of type GenericIRPManagementConstDefs::VersionNumberSet.
filter	One NV pair of remaining_body	O	Name of NV pair is notificationLogIRPNotifications:: NotifyLogSubscribed:: FILTER Value of NV pair is a string.
NOTE: NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. This note is applicable to all NV-pairs and all mapping tables for all notifications.			

A.3 Solution Set definitions

A.3.1 IDL definition structure

Clause A.3.2 defines the constants and types used by the NL IRP.

Clause A.3.3 defines the operations which are performed by the NL IRP agent.

Clause A.3.4 defines the notifications which are emitted by the NL IRP agent.

A.3.2   IDL specification "NotificationLogIRPConstDefs.idl"

```
// File: NotificationLogIRPConstDefs.idl

#ifndef _NOTIFICATIONLOGIRPCONSTDEFS_IDL_
#define _NOTIFICATIONLOGIRPCONSTDEFS_IDL_

#ifndef _DSLOGADMIN_IDL_
#define _DSLOGADMIN_IDL_
#include <DsLogAdmin.idl>
#endif // _DSLOGADMIN_IDL_

#include "NotificationIRPConstDefs.idl"
#include "GenericIRPManagementConstDefs.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module NotificationLogIRPConstDefs
{

    typedef DsLogAdmin::LogId LogSubscriptionId;
    typedef DsLogAdmin::RecordId RecordId;
    typedef DsLogAdmin::RecordIdList RecordIdList;

    typedef string LogManagerToken;
    /*
    LogManagerTokenOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present; else absent.
    */
    union LogManagerTokenOpt switch (boolean)
    {
        case TRUE:  LogManagerToken value;
    };


    /*
    IRPTimeOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present; else absent.
    */
    union IRPTimeOpt switch (boolean)
    {
        case TRUE: GenericIRPManagementConstDefs::IRPTime value;
    };


    typedef GenericIRPManagementConstDefs::VersionNumberSet
        NotificationCategorySet;
    /*
    NotificationCategorySetOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present; else absent.
    */
    union NotificationCategorySetOpt switch (boolean)
    {
        case TRUE:  NotificationCategorySet value;
    };

    enum LogState { LOGGING, LOGFULL, STOPPED};

    typedef DsLogAdmin::CapacityAlarmThresholdList
        CapacityAlarmThresholdList;

    typedef unsigned short LogFullActionType;
    const LogFullActionType wrap = 0;
    const LogFullActionType halt = 1;


    /*
    IteratorOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present; else absent.
    */
    union IteratorOpt switch (boolean)
    {
        case TRUE:  DsLogAdmin::Iterator value;
    };
```

```
    struct LogAttributes {
        LogSubscriptionId logSubscriptionId;
        GenericIRPManagementConstDefs::IRPTime loggingEndTime;
        unsigned long long maxSize;
        unsigned long long currentSize;
        LogState logState;
        unsigned long long logRecordCount;
        NotificationIRPConstDefs::NotificationCategorySet notificationCategories;
        string filter;
        LogFullActionType logFullAction;
        CapacityAlarmThresholdList occupancyLevels;
        };

    typedef sequence <LogAttributes> LogAttributeList;


};


#endif // _NOTIFICATIONLOGIRPCONSTDEFS_IDL_
```

A.3.3   IDL specification "NotificationLogIRPSystem.idl"

```
// File: NotificationLogIRPSystem.idl

#ifndef _NOTIFICATIONLOGIPRSYSTEM_IDL_
#define _NOTIFICATIONLOGIRPSYSTEM_IDL_

#include "NotificationLogIRPConstDefs.idl"
#include "GenericIRPManagementSystem.idl"
#include <TimeBase.idl>
#ifndef _DSLOGADMIN_IDL_
#define _DSLOGADMIN_IDL_
#include <DsLogAdmin.idl>
#endif // _DSLOGADMIN_IDL_

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module NotificationLogIRPSystem
{
    /*
    System fails to complete the method.  System can provide reason
    to qualify the exception.  The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception SubscribeLog { string reason; };
    exception UnsubscribeLog { string reason; };
    exception ExportLogRecords { string reason; };
    exception GetLogSubscriptionIds { string reason; };
    exception GetLogRecords { string reason; };
    exception GetLogSubscriptionStatus { string reason; };

    exception InvalidLogSubscriptionId { string reason; };
    exception UnknownLogSubscriptionId {};
    exception UnknownLogManagerToken {};
    exception InvalidConstraint { string reason; };


    interface NotificationLogIRP : GenericIRPManagementSystem::
        GenericIRPManagement
    {
        GenericIRPManagementConstDefs::Signal subscribe_log (
            in NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId,
            in NotificationLogIRPConstDefs::IRPTimeOpt loggingEndTimeAsked,
            in NotificationLogIRPConstDefs::NotificationCategorySetOpt
                notificationCategorySet,
            in GenericIRPManagementConstDefs::StringOpt filter,
            out NotificationLogIRPConstDefs::LogSubscriptionId
                logSubscriptionIdOut,
            out NotificationLogIRPConstDefs::LogManagerTokenOpt logManagerToken,
            out NotificationLogIRPConstDefs::IRPTimeOpt loggingEndTimeGiven
        )
        raises ( SubscribeLog,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ParameterNotSupported,
            GenericIRPManagementSystem::ValueNotSupported,
            InvalidLogSubscriptionId, UnknownLogManagerToken
            );

        GenericIRPManagementConstDefs::Signal unsubscribe_log (
            in NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId,
            in NotificationLogIRPConstDefs::LogManagerTokenOpt logManagerToken
        )
        raises ( UnsubscribeLog,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ParameterNotSupported,
            UnknownLogSubscriptionId, UnknownLogManagerToken);

    GenericIRPManagementConstDefs::Signal export_log_records (
        in NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId,
        in NotificationLogIRPConstDefs::NotificationCategorySetOpt
            notificationCategorySet,
        in GenericIRPManagementConstDefs::StringOpt filter,
        out string invocationId
        )
```

```
            raises ( ExportLogRecords,
                GenericIRPManagementSystem::InvalidParameter,
                GenericIRPManagementSystem::ParameterNotSupported,
                GenericIRPManagementSystem::OperationNotSupported,
                UnknownLogSubscriptionId);


        //  If some but not all of the information is returned via the return value
        //  RecordList, then the rest of the information is returned via the
        //  iterator.  Otherwise, the iterator is absent.
        //
        DsLogAdmin::RecordList get_log_records(
            in NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId,
            in NotificationLogIRPConstDefs::NotificationCategorySetOpt
                notificationCategories,
            in GenericIRPManagementConstDefs::StringOpt filter,
            out NotificationLogIRPConstDefs::IteratorOpt iterator
            )
            raises ( GetLogRecords,
                GenericIRPManagementSystem::InvalidParameter,
                GenericIRPManagementSystem::ParameterNotSupported,
                GenericIRPManagementSystem::ValueNotSupported,
                GenericIRPManagementSystem::OperationNotSupported,
                UnknownLogSubscriptionId);


        DsLogAdmin::LogIdList get_log_subscription_ids()
            raises ( GetLogSubscriptionIds,
                 GenericIRPManagementSystem::OperationNotSupported
                 );


        GenericIRPManagementConstDefs::Signal get_log_subscription_status (
            in NotificationLogIRPConstDefs::LogSubscriptionId logSubscriptionId,
            out NotificationLogIRPConstDefs::LogAttributeList logAttributeList
            )
            raises ( GetLogSubscriptionStatus,
                GenericIRPManagementSystem::InvalidParameter,
                GenericIRPManagementSystem::OperationNotSupported
                );

    };
};

#endif // _NOTIFICATIONLOGIPRSYSTEM_IDL_
```

A.3.3   IDL specification "NotificationLogIRPNotifications.idl"

```
// File: NotificationLogIRPNotifications.idl

#ifndef _NOTIFICATIONLOGIRPNOTIFICATIONS_IDL_
#define _NOTIFICATIONLOGIRPNOTIFICATIONS_IDL_

#include "NotificationIRPNotifications.idl"

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module NotificationLogIRPNotifications
{

    interface NotifyLogSubscribed: NotificationIRPNotifications::Notify
    {
        const string NOTIFY_LOG_SUBSRIBED = "x1";
        const string LOG_SUBSCRIPTION_ID = "id";
        const string LOGGING_END_TIME = "loggingEndTime";
        const string NOTIFICATION_CATEGORIES = "categories";
        const string FILTER = "filter";

    };

    interface NotifyLogUnubscribed: NotificationIRPNotifications::Notify
    {
        const string NOTIFY_LOG_UNSUBSCRIBED = "x2";
        const string LOG_SUBSCRIPTION_ID = "id";
    };

    interface NotifyLogOccupancyLevelCrossed:
        NotificationIRPNotifications::Notify
    {
        const string NOTIFY_LOG_OCCUPANCY_LEVEL_CROSSED = "x3";
        const string LOG_SUBSCRIPTION_ID = "id";
        const string CURRENT_OCCUPANCY_LEVEL = "level";
        const string LOG_FULL_ACTION = "fullAction";
    };

    interface NotifyLoggingResumed: NotificationIRPNotifications::Notify
    {
        const string NOTIFY_LOGGING_RESUMED = "x4";
        const string LOG_SUBSCRIPTION_ID = "id";
    };


};

#endif // _NOTIFICATIONLOGIRPNOTIFICATIONS_IDL_
```

Annex B (normative):
XML Definitions

This annex contains the XML Definitions for the Notification Log Integration Reference Point (NL IRP) as it applies to Itf-N, in accordance with NL IRP IS definitions [5].

Apart from being used for the Notification Log, the XML definitions for NL IRP notifications are used by the NL IRP SOAP SS.

B.1 Architectural Features

The overall architectural feature of NL IRP is specified in 3GPP TS 32.332 [5]. This clause specifies features that are specific to the XML definitions.

B.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [17].

B.1.2 Notification Services

This annex defines the XML syntax of NL IRP notifications that is to be used for the NL IRP SOAP Solution Set and NL IRP XML Data File and NL IRP XML Notification Format.

B.1.3 IOC Definitions

This annex defines the XML syntax for the IOC definitions of the NL IRP IS [5], which are used by the XML definitions for the NL IRP notifications and the NL IRP IS operations.

B.2 Mapping

Not present in the current version of this specification.

B.3 Solution Set definitions

B.3.1 XML definition structure

The present clause defines the XML exchanged between an IRPManager and an IRPAgent as part of NL IRP IS (see 3GPP TS 32.332 [5]):

- the NL IRP XML Data File,
- NL IRP notifications.

The use of XML schemas ensures NL XML have the proper structure and to some extent the proper content, and in particular to ensure:

- for a given Notification instance, it is properly named/positioned with regard to the global naming tree;

- for a given Notification instance, only attributes of the corresponding Notification are present;

- for a given Notification attribute, its value is of the proper type.

XML definitions for other application specific notifications (e.g. Alarm IRP notifications, Notification IRP Notification Header) are defined within the respective IRP series within specification 32.xy 6.

B.3.1.1   Global structure

Editor's Note: To be added in future version of this specification.

B.3.1.2   XML elements fileHeader and fileFooter

Editor's Note: To be added in future version of this specification.

B.3.1.3   NL IRP XML File Name Conventions

For NL IRP XML File Name Conventions the generic file name definitions as specified by the FT IRP apply (see [6]).

B.3.1.4   NL Interface IRP-specific XML schemas

Interface IRP-specific XML schemas with definition of corresponding XML namespace prefixes are listed by the following table:

Table B.3.1.4: Interface IRP-specific XML schemas, corresponding 3GPP TSs and XML namespace prefixes

IRP	XML schema	3GPP TS no.	XML namespace prefix
Alarm IRP	alarmIRPnotif.xsd	32.111-6 [15]	xa
Notification IRP	notification.xsd	32.306 [11]	xe
Notification Log IRP	notificationLog.xsd	32.336	xl

Each Interface IRP-specific XML schema explicitly declares Interface IRP-specific XML element types for the related Interface IRP.

Additionally, XML schema notification.xsd (see [11]) also provides global XML declarations and definitions for the support of:

- Notification header XML element type declaration

B.3.1.5   NL IRP XML File Content Definitions

B.3.1.5.1   NL IRP XML Data File

The structure of the Notification Log IRP XML Data File is shown in graphical depiction below:
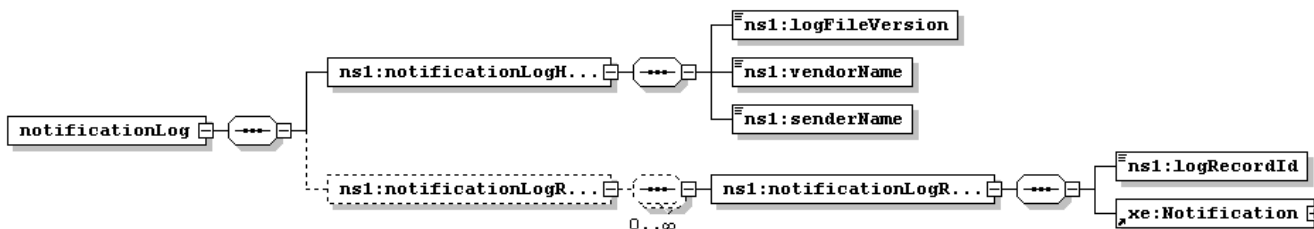
Figure B.3.1.5.1: Structure of the Notification Log IRP XML Data File

The notificationLogRecords element holds the instances of the notification records that are imported from the Notification Format XML schema. Note that Notification Header XML definitions are provided in [11].

The Notification Log IRP XML Data File is a simple instance based construct. The XML schema for the Notification Log IRP XML Data File is defined in Annex B.3.3.

B.3.2 Graphical Representation
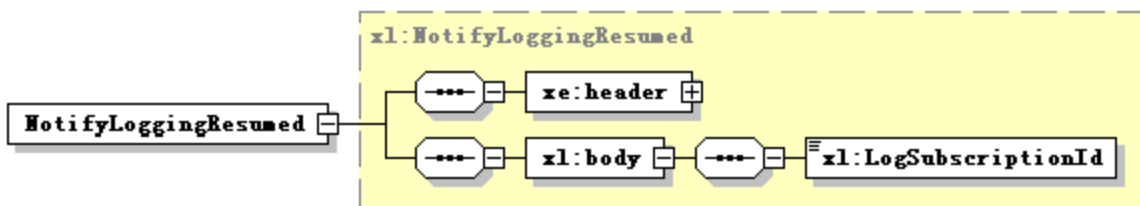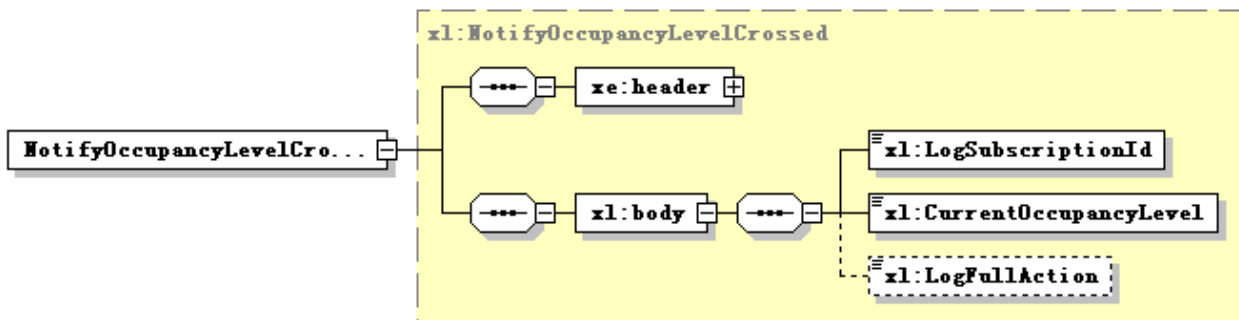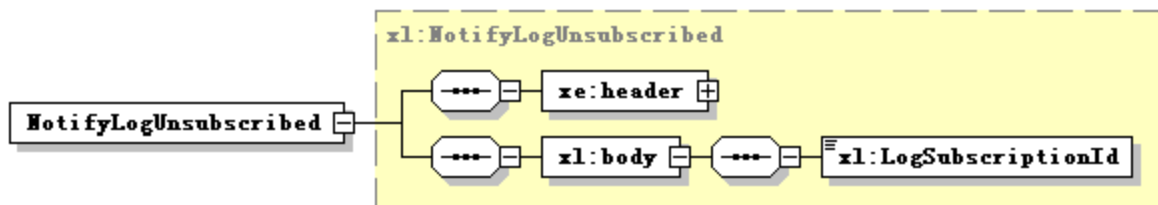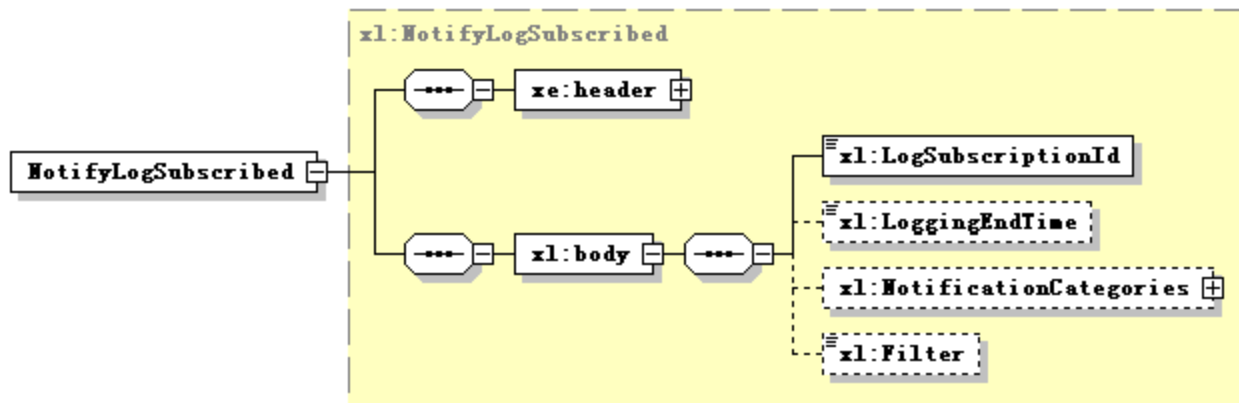
The use of XML schema key word "sequence" to support IS-defined set (not sequence) is for the purpose of XML processor efficiency. This shall not imply the use of "sequence" in other technology.

B.3.3 XML Schema "notificationLog.xsd"

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
    3GPP TS 32.336 Notification Log IRP
    3GPP Notification Log IRP XML Solution Definition, Schema Definition
    notificationLog.xsd
-->
<schema
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#notificationLog"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"
xmlns:xa="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
>
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"/>
 <element name="notificationLog">
        <complexType>
            <sequence>
                <element name="notificationLogHeader" minOccurs="1" maxOccurs="1">
                    <complexType>
                        <sequence>
                            <element name="logFileVersion" type="string" minOccurs="1"
maxOccurs="1"/>
                            <element name="vendorName" type="string" minOccurs="1" maxOccurs="1"/>
                            <element name="senderName" type="string" minOccurs="1" maxOccurs="1"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="notificationLogRecords" minOccurs="0">
                    <complexType>
                        <sequence minOccurs="0" maxOccurs="unbounded">
                            <element name="notificationLogRecord">
                                <complexType>
                                    <sequence>
                                        <element name="logRecordId" type="string"/>
                                        <element ref="xe:Notification"/>
                                    </sequence>
                                </complexType>
                            </element>
                        </sequence>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

B.3.4 XML Schema "ntfLogNotif.xsd"

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
    3GPP TS 32.336 NotificationLogIRP Notification XML Schema
    ntfLogIRPNotif.xsd
-->
<schema xmlns:xl="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRPNotif"
xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRPNotif"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"/>
    <simpleType name="LogSubscriptionId">
        <restriction base="string">
            <minLength value="1"/>
        </restriction>
    </simpleType>
    <simpleType name="CurrentOccupancyLevel">
        <restriction base="short">
            <maxInclusive value="100"/>
            <minInclusive value="0"/>
        </restriction>
    </simpleType>
    <simpleType name="LogFullAction">
        <restriction base="string">
            <enumeration value="Wrap"/>
            <enumeration value="Halt"/>
        </restriction>
    </simpleType>
    <complexType name="NotificationCategories">
        <sequence>
            <element name="IRPVersion" type="string" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="NotifyLogSubscribed">
        <complexContent>
            <extension base="xe:Notification">
                <sequence>
                    <element name="body">
                        <complexType>
                            <sequence>
                                <element name="LogSubscriptionId" type="xl:LogSubscriptionId"/>
                                <element name="LoggingEndTime" type="dateTime" minOccurs="0"/>
                                <element name="NotificationCategories"
type="xl:NotificationCategories" minOccurs="0"/>
                                <element name="Filter" type="string" minOccurs="0"/>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="NotifyLogUnsubscribed">
        <complexContent>
            <extension base="xe:Notification">
                <sequence>
                    <element name="body">
                        <complexType>
                            <sequence>
                                <element name="LogSubscriptionId" type="xl:LogSubscriptionId"/>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="NotifyOccupancyLevelCrossed">
        <complexContent>
            <extension base="xe:Notification">
                <sequence>
                    <element name="body">
                        <complexType>
                            <sequence>
```

```
                                <element name="LogSubscriptionId" type="xl:LogSubscriptionId"/>
                                <element name="CurrentOccupancyLevel"
type="xl:CurrentOccupancyLevel"/>
                                <element name="LogFullAction" type="xl:LogFullAction"
minOccurs="0"/>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="NotifyLoggingResumed">
        <complexContent>
            <extension base="xe:Notification">
                <sequence>
                    <element name="body">
                        <complexType>
                            <sequence>
                                <element name="LogSubscriptionId" type="xl:LogSubscriptionId"/>
                            </sequence>
                        </complexType>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="NotifyLogSubscribed" type="xl:NotifyLogSubscribed"/>
    <element name="NotifyLogUnsubscribed" type="xl:NotifyLogUnsubscribed"/>
    <element name="NotifyOccupancyLevelCrossed" type="xl:NotifyOccupancyLevelCrossed"/>
    <element name="NotifyLoggingResumed" type="xl:NotifyLoggingResumed"/>
</schema>
```

B.3.5   NL IRP XML Example Notifications

This annex provides an XML example notification, demonstrating the use of and the embedding into the NL IRP XML Notification Format. This example makes use of the Alarm IRP XML Schema for Notification.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<notificationLog xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#notificationLog"
xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#notificationLog
NotificationLog.xsd" >
    <notificationLogHeader>
        <logFileVersion>32.336 V10.0</logFileVersion>
        <vendorName>Text</vendorName>
        <senderName>Text</senderName>
    </notificationLogHeader>
    <notificationLogRecords>
        <notificationLogRecord>
            <logRecordId>1001</logRecordId>
            <xe:Notification xsi:type="NotifyNewSecurityAlarm"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
xsi:schemaLocation="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif
alarmIRPNotif.xsd">
                <xe:header>
                    <xe:objectInstance>String</xe:objectInstance>
                    <xe:notificationId>2147483647</xe:notificationId>
                    <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                    <xe:systemDN>String</xe:systemDN>
                    <xe:notificationType>String</xe:notificationType>
                </xe:header>
                <body>
                    <probableCause>Indeterminate</probableCause>
                    <perceivedSeverity>Critical</perceivedSeverity>
                    <alarmType>Communications Alarm</alarmType>
                    <correlatedNotifications>
                        <correlatedNotification>
                            <source>String</source>
                            <notificationIdSet>
                                <notificationId>2147483647</notificationId>
                            </notificationIdSet>
                        </correlatedNotification>
                    </correlatedNotifications>
                    <additionalText>String</additionalText>
                    <additionalInformation>String</additionalInformation>
                </body>
            </xe:Notification>
        </notificationLogRecord>
        <notificationLogRecord>
            <logRecordId>String</logRecordId>
            <xe:Notification xsi:type="NotifyNewAlarm"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
                <xe:header>
                    <xe:objectInstance>String</xe:objectInstance>
                    <xe:notificationId>2147483647</xe:notificationId>
                    <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                    <xe:systemDN>String</xe:systemDN>
                    <xe:notificationType>String</xe:notificationType>
                </xe:header>
                <body>
                    <probableCause>Indeterminate</probableCause>
                    <perceivedSeverity>Critical</perceivedSeverity>
                    <alarmType>Communications Alarm</alarmType>
                    <alarmId>String</alarmId>
                    <specificProblem>String</specificProblem>
                    <correlatedNotifications>
                        <correlatedNotification>
                            <source>String</source>
                            <notificationIdSet>
                                <notificationId>2147483647</notificationId>
                            </notificationIdSet>
                        </correlatedNotification>
                    </correlatedNotifications>
                    <backedUpStatus>true</backedUpStatus>
                    <backupObject>String</backupObject>
                    <trendIndication>Less Severe</trendIndication>
                    <thresholdInfo>
```

```
                    <attributeId>String</attributeId>
                    <observedValue>3.14159E0</observedValue>
                    <thresholdLevel>
                        <indication>Up</indication>
                        <low>3.14159E0</low>
                        <high>3.14159E0</high>
                    </thresholdLevel>
                    <armTime>2001-12-17T09:30:47.0Z</armTime>
                </thresholdInfo>
                <stateChangeDefinition>
                    <stateValueChange>
                        <attributeName>Administrative State</attributeName>
                        <oldValue>
                            <administrativeState>locked</administrativeState>
                        </oldValue>
                        <newValue>
                            <administrativeState>unlocked</administrativeState>
                        </newValue>
                    </stateValueChange>
                    <stateValueChange>
                        <attributeName>Usage State</attributeName>
                        <oldValue>
                            <usageState>idle</usageState>
                        </oldValue>
                        <newValue>
                            <usageState>busy</usageState>
                        </newValue>                          </stateValueChange>
                </stateChangeDefinition>
                <monitoredAttributes>
                    <attribueValue>
                        <attributeName>String</attributeName>
                        <attributeValue>Text</attributeValue>
                    </attribueValue>
                </monitoredAttributes>
                <proposedRepairActions>String</proposedRepairActions>
                <additionalText>String</additionalText>
                <additionalInformation>String</additionalInformation>
            </body>
        </xe:Notification>
    </notificationLogRecord>
    <notificationLogRecord>
        <logRecordId/>
        <xe:Notification xsi:type="NotifyAckStateChanged"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
        <xe:header>
            <xe:objectInstance>String</xe:objectInstance>
            <xe:notificationId>2147483647</xe:notificationId>
            <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
            <xe:systemDN>String</xe:systemDN>
            <xe:notificationType>String</xe:notificationType>
        </xe:header>
        <body>
            <probableCause>Indeterminate</probableCause>
            <perceivedSeverity>Critical</perceivedSeverity>
            <alarmType>Communications Alarm</alarmType>
            <alarmId>String</alarmId>
            <ackState>Acknowledged</ackState>
            <ackUserId>String</ackUserId>
            <ackSystemId>String</ackSystemId>
        </body>
        </xe:Notification>
    </notificationLogRecord>
    <notificationLogRecord>
        <logRecordId/>
        <xe:Notification xsi:type="NotifyAlarmListRebuilt"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
        <xe:header>
            <xe:objectInstance>String</xe:objectInstance>
            <xe:notificationId>2147483647</xe:notificationId>
            <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
            <xe:systemDN>String</xe:systemDN>
            <xe:notificationType>String</xe:notificationType>
        </xe:header>
        <body>
            <reason>
                <predefined>Agent NE Communications Error</predefined>
            </reason>
```

```
                        <alarmListAlignmentRequirement>Alignment
Required</alarmListAlignmentRequirement>
                    </body>
                </xe:Notification>
            </notificationLogRecord>
            <notificationLogRecord>
                <logRecordId/>
                <xe:Notification xsi:type="NotifyChangedAlarm"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
                    <xe:header>
                        <xe:objectInstance>String</xe:objectInstance>
                        <xe:notificationId>2147483647</xe:notificationId>
                        <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                        <xe:systemDN>String</xe:systemDN>
                        <xe:notificationType>String</xe:notificationType>
                    </xe:header>
                    <body>
                        <probableCause>Indeterminate</probableCause>
                        <perceivedSeverity>Critical</perceivedSeverity>
                        <alarmType>Communications Alarm</alarmType>
                        <alarmId>String</alarmId>
                    </body>
                </xe:Notification>
            </notificationLogRecord>
            <notificationLogRecord>
                <logRecordId/>
                <xe:Notification xsi:type="NotifyClearedAlarm"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
                    <xe:header>
                        <xe:objectInstance>String</xe:objectInstance>
                        <xe:notificationId>2147483647</xe:notificationId>
                        <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                        <xe:systemDN>String</xe:systemDN>
                        <xe:notificationType>String</xe:notificationType>
                    </xe:header>
                    <body>
                        <probableCause>Indeterminate</probableCause>
                        <perceivedSeverity>Critical</perceivedSeverity>
                        <alarmType>Communications Alarm</alarmType>
                        <alarmId>String</alarmId>
                        <correlatedNotifications>
                            <correlatedNotification>
                                <source>String</source>
                                <notificationIdSet>
                                    <notificationId>2147483647</notificationId>
                                    <notificationId>2147483647</notificationId>
                                </notificationIdSet>
                            </correlatedNotification>
                            <correlatedNotification>
                                <source>String</source>
                                <notificationIdSet>
                                    <notificationId>2147483647</notificationId>
                                    <notificationId>2147483647</notificationId>
                                </notificationIdSet>
                            </correlatedNotification>
                        </correlatedNotifications>
                        <clearUserId>String</clearUserId>
                        <clearSystemId>String</clearSystemId>
                    </body>
                </xe:Notification>
            </notificationLogRecord>
            <notificationLogRecord>
                <logRecordId/>
                <xe:Notification xsi:type="NotifyComments"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
                    <xe:header>
                        <xe:objectInstance>String</xe:objectInstance>
                        <xe:notificationId>2147483647</xe:notificationId>
                        <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                        <xe:systemDN>String</xe:systemDN>
                        <xe:notificationType>String</xe:notificationType>
                    </xe:header>
                    <body>
                        <alarmType>Communications Alarm</alarmType>
                        <probableCause>Indeterminate</probableCause>
                        <perceivedSeverity>Critical</perceivedSeverity>
                        <comments>
                            <comment>
```

```
                                    <commentDate>2001-12-17T09:30:47.0Z</commentDate>
                                    <commentText>String</commentText>
                                    <userId>String</userId>
                                    <systemId>String</systemId>
                                </comment>
                            </comments>
                            <alarmId>String</alarmId>
                        </body>
                    </xe:Notification>
            </notificationLogRecord>
            <notificationLogRecord>
                    <logRecordId/>
                    <xe:Notification xsi:type="NotifyPotentialFaultyAlarmList"
xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.111-6#alarmIRPNotif"
                        <xe:header>
                            <xe:objectInstance>String</xe:objectInstance>
                            <xe:notificationId>2147483647</xe:notificationId>
                            <xe:eventTime>2001-12-17T09:30:47.0Z</xe:eventTime>
                            <xe:systemDN>String</xe:systemDN>
                            <xe:notificationType>String</xe:notificationType>
                        </xe:header>
                        <body>
                            <reason>
                                <predefined>Agent NE Communications Error</predefined>
                            </reason>
                        </body>
                    </xe:Notification>
            </notificationLogRecord>
    </notificationLogRecords>
</notificationLog>
```

Annex C (normative):
SOAP Solution Set

This annex specifies the SOAP Solution Set for the IRP whose semantics are specified in NL IRP: Information Service (3GPP TS 32.332 [5]).

C.1 Architectural features

The overall architectural feature of the NL IRP is specified in 3GPP TS 32.332 [5]. This clause specifies features that are specific to the SOAP Solution Set.

C.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [17].

C.1.2 Notification Services

The NL IRP SOAP SS uses the Notification IRP SOAP SS of 3GPP TS 32.306 [11]. The IRPAgent shall support the push interface model, which means that the IRPAgent sends notifications to the IRPManager as soon as new events occur. The IRPManager does not need to check ("pull") for events.

Relevant definitions are imported from the NL IRP XML definitions in Annex B.

C.1.3 Supported W3C specifications

The SOAP 1.1 specification [7] and WSDL 1.1 specification [9] are supported.

The SOAP 1.2 specification [10] is supported optionally.

This specification uses "document" style in WSDL file.

This specification uses "literal" encoding style in WSDL file.

The filter language used in the SS is the XPath Language (see W3C XPath 1.0 specification [8]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

C.1.4 Prefixes and namespaces

This specification uses a number of namespace prefixes throughout that are listed in Table C.1.4.

Table C.1.4: Prefixes and Namespaces used in this specification

PREFIX	NAMESPACE
(no prefix)	http://schemas.xmlsoap.org/wsdl/
soap	http://schemas.xmlsoap.org/wsdl/soap/
nlRPSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLRPSystem
nlIRPData	http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLIRPData
xl	http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRPNotif
genericIRPSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRPSystem
ntfIRPNtfSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.306#NotificationIRPNtfSystem

C.2 Mapping

C.2.1 Operation and notification mapping

The Notification Log (NL) IRP IS (3GPP TS 32.332 [5]) defines semantics of operation and notification visible across the Itf-N. The following tables indicate mapping of these operations and notifications to their equivalents defined in this SS.

Table C.2.1: Mapping from IS Operation to SS Equivalents

IS Operations in 3GPP TS 32.332 [5]	SS Operations	SS Port	Qualifier
subscribeLog	subscribeLog	NLIRPOperations1Port	M
unsubscribeLog	unsubscribeLog	NLIRPOperations1Port	M
exportLogRecords	exportLogRecords	NLIRPOperations1Port	O
getLogRecords	getLogRecords	NLIRPOperations1Port	O
getLogSubscriptionIds	getLogSubscriptionIds	NLIRPOperations2Port	O
getLogSubscriptionStatus	getLogSubscriptionStatus	NLIRPOperations2Port	M
notifyLogSubscribed	notify (note 1)	NotificationIRPNtfPort	M
notifyLogUnsubscribed	notify (note 1)	NotificationIRPNtfPort	M
notifyOccupancyLevelCrossed	notify (note 1)	NotificationIRPNtfPort	M
notifyLoggingResumed	notify (note 1)	NotificationIRPNtfPort	M
NOTE 1: The IS equivalent maps to an XML definition specified in Annex B, and this being an input parameter to the operation notify under the port type ntfIRPNtfSystem:NotificationIRPNtf and under the binding ntfIRPNtfSystem:NotificationIRPNtf of 3GPP TS 32.306 [11].			

C.2.2 Operation parameter mapping

The Notification Log (NL) IRP IS (3GPP TS 32.332 [5]) defines semantics of parameters carried in the operations. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table C.2.2.1: Mapping from IS `subscribeLog` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	logSubscriptionId	M
loggingEndTIme	loggingEndTIme	O
notificationCategories	notificationCategories	O
filter	filter	O
logManagerToken	logManagerToken	O
loggingEndTime	loggingEndTime	O
status	status	M

Table C.2.2.2: Mapping from IS `unsubscribeLog` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	logSubscriptionId	M
logManagerToken	logManagerToken	O
status	status	M

Table C.2.2.3: Mapping from IS `exportLogRecords` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	logSubscriptionId	M
notificationCategories	notificationCategories	O
filter	filter	O
invocationId	invocationId	M
status	status	M

Table C.2.2.4: Mapping from IS `getLogRecords` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	logSubscriptionId	M
notificationCategories	notificationCategories	O
filter	filter	O
getLogRecordsResult	getLogRecordsResult	M
status	status	M

Table C.2.2.5: Mapping from IS `getLogSubscriptionIds` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionIds	logSubscriptionIds	M
status	status	M

Table C.2.2.6: Mapping from IS `getLogSubscriptionStatus` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
logSubscriptionId	logSubscriptionId	M
logAttributeList	logAttributeList	M
status	status	M

C.2.3 Notification parameter mapping

The Notification Log (NL) IRP Notifications are defined in Annex B.

C.3 Solution Set definitions

C.3.1 WSDL definition structure

Clause C.3.2 provides a graphical representation of the NL IRP service.

Clause C.3.3 defines the services which are supported the NL IRP agent.

C.3.2 Graphical Representation

The WSDL structure is depicted in Figure C.3.2 below, depicting port type, binding and service. The port type contains port type operations, which again contains input, output and fault messages. The binding contains binding operations, which have the same name as the port type operations. The binding connects to a port inside the service.
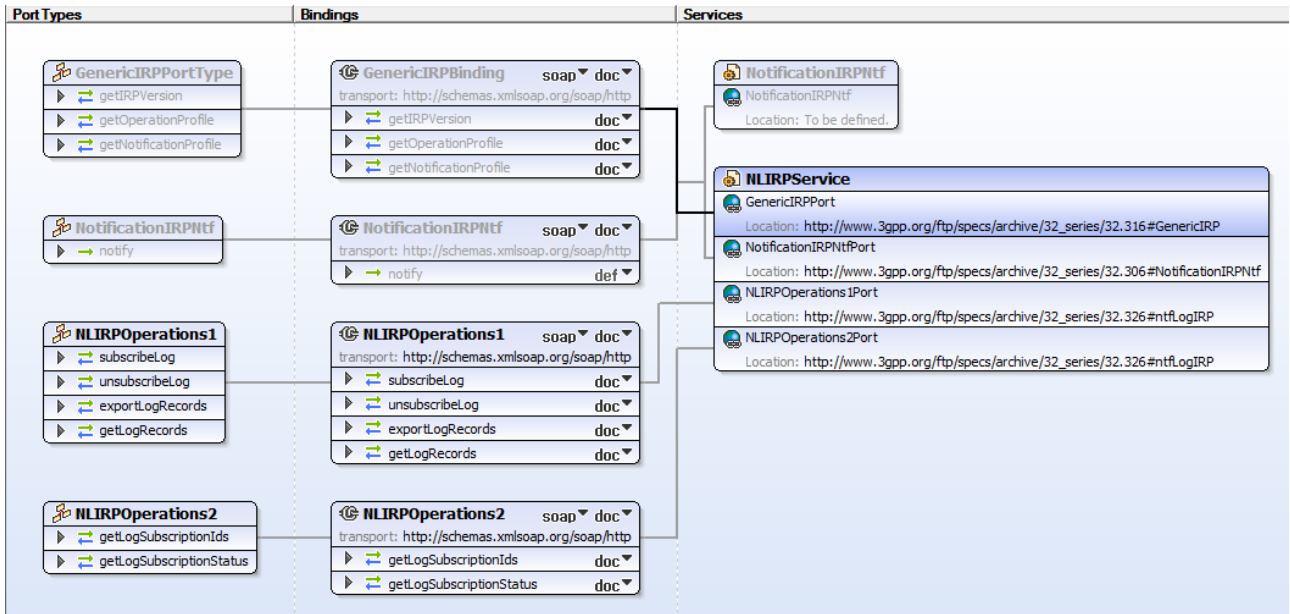
Figure C.3.2: Notification Log (NL) IRP SOAP Solution Set WSDL structure

C.3.3   WSDL specification "NLSystem.wsdl"

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   3GPP TS 32.336 Notification Log IRP SOAP Solution Set
-->
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:nlIRPSystem="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLIRPSystem"
xmlns:nlIRPData="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLIRPData"
xmlns:genericIRPSystem="http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRPSystem"
xmlns:ntfIRPNtfSystem="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#NotificationIRPNtfSyst
em" xmlns:xl="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRPNotif"
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLIRPSystem">
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRPSystem"/>
  <import
namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#NotificationIRPNtfSystem"/>
  <types>
    <schema targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#NLIRPData"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRPNotif"/>
      <!--  loggingEndTime Type -->
      <complexType name="loggingEndTimeType">
        <choice>
          <element name="dateTime" type="dateTime"/>
          <element name="indefinite"/>
        </choice>
      </complexType>
      <!-- subscribeLog Request -->
      <element name="subscribeLogRequest">
        <complexType>
          <sequence>
            <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
            <element name="loggingEndTime" type="nlIRPData:loggingEndTimeType" minOccurs="0"/>
            <element name="notificationCategories" type="xl:NotificationCategories" minOccurs="0"/>
            <element name="filter" type="string" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
      <!-- subscribeLog Response -->
      <element name="subscribeLogResponse">
        <complexType>
          <sequence>
            <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
            <element name="logManagerToken" type="string" minOccurs="0"/>
            <element name="loggingEndTime" type="nlIRPData:loggingEndTimeType" minOccurs="0"/>
            <element name="status">
              <simpleType>
                <restriction base="string">
                  <enumeration value="OperationSucceeded"/>
                  <enumeration value="OperationFailed"/>
                </restriction>
              </simpleType>
            </element>
            <element name="failureReason" minOccurs="0">
              <simpleType>
                <restriction base="string">
                  <enumeration value="logSubscriptionId_required"/>
                  <enumeration value="maxLogs_reached"/>
                  <enumeration value="operation_failed"/>
                  <enumeration value="operation_failed_invalid_input_parameter"/>
                  <enumeration
value="operation_failed_unsupported_optional_input_parameter_loggingEndTime"/>
                  <enumeration
value="operation_failed_unsupported_optional_input_parameter_notificationCategories"/>
                  <enumeration
value="operation_failed_unsupported_optional_input_parameter_filter"/>
                  <enumeration value="operation_failed_internal_problem"/>
                </restriction>
              </simpleType>
            </element>
          </sequence>
        </complexType>
      </element>
      <!-- subscribeLog Fault -->
      <element name="subscribeLogFault">
```

```xml
        <simpleType>
         <restriction base="string">
           <enumeration value="OperationFailed"/>
         </restriction>
        </simpleType>
      </element>
      <!-- unsubscribeLog Request -->
      <element name="unsubscribeLogRequest">
        <complexType>
         <sequence>
           <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
           <element name="logManagerToken" type="string" minOccurs="0"/>
         </sequence>
        </complexType>
      </element>
      <!-- unsubscribeLog Response -->
      <element name="unsubscribeLogResponse">
        <complexType>
         <sequence>
           <element name="status">
             <simpleType>
               <restriction base="string">
                 <enumeration value="OperationSucceeded"/>
                 <enumeration value="OperationFailed"/>
               </restriction>
             </simpleType>
           </element>
           <element name="failureReason" minOccurs="0">
             <simpleType>
               <restriction base="string">
                 <enumeration value="operation_failed"/>
                 <enumeration value="operation_failed_invalid_input_parameter"/>
                 <enumeration
value="operation_failed_unsupported_optional_input_parameter_logManagerToken"/>
                 <enumeration value="operation_failed_internal_problem"/>
               </restriction>
             </simpleType>
           </element>
         </sequence>
        </complexType>
      </element>
      <!-- unsubscribeLog Fault -->
      <element name="unsubscribeLogFault">
        <simpleType>
         <restriction base="string">
           <enumeration value="OperationFailed"/>
         </restriction>
        </simpleType>
      </element>
      <!-- exportLogRecords Request -->
      <element name="exportLogRecordsRequest">
        <complexType>
         <sequence>
           <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
           <element name="notificationCategories" type="xl:NotificationCategories" minOccurs="0"/>
           <element name="filter" type="string" minOccurs="0"/>
         </sequence>
        </complexType>
      </element>
      <!-- exportLogRecords Response -->
      <element name="exportLogRecordsResponse">
        <complexType>
         <sequence>
           <element name="invocationId" type="string"/>
           <element name="status">
             <simpleType>
               <restriction base="string">
                 <enumeration value="OperationSucceeded"/>
                 <enumeration value="OperationFailed"/>
               </restriction>
             </simpleType>
           </element>
           <element name="failureReason" minOccurs="0">
             <simpleType>
               <restriction base="string">
                 <enumeration value="export_failed"/>
                 <enumeration value="operation_failed"/>
                 <enumeration value="operation_failed_invalid_input_parameter"/>
```

```
                <enumeration
value="operation_failed_unsupported_optional_input_parameter_notificationCategories"/>
                <enumeration
value="operation_failed_unsupported_optional_input_parameter_filter"/>
                <enumeration value="operation_failed_internal_problem"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </complexType>
    </element>
    <!-- exportLogRecords Fault -->
    <element name="exportLogRecordsFault">
      <simpleType>
        <restriction base="string">
          <enumeration value="OperationFailed"/>
        </restriction>
      </simpleType>
    </element>
    <!-- getLogRecords Request -->
    <element name="getLogRecordsRequest">
      <complexType>
        <sequence>
          <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
          <element name="notificationCategories" type="xl:NotificationCategories" minOccurs="0"/>
          <element name="filter" type="string" minOccurs="0"/>
        </sequence>
      </complexType>
    </element>
    <!-- getLogRecords Response -->
    <element name="getLogRecordsResponse">
      <complexType>
        <sequence>
          <element name="getLogRecordsResult">
            <complexType>
              <sequence>
                <element name="LogRecord" maxOccurs="unbounded">
                  <complexType>
                    <sequence>
                      <element name="logRecordId" type="string"/>
                      <element name="logRecordContent" type="string" minOccurs="0"/>
                    </sequence>
                  </complexType>
                </element>
              </sequence>
            </complexType>
          </element>
          <element name="status">
            <simpleType>
              <restriction base="string">
                <enumeration value="OperationSucceeded"/>
                <enumeration value="OperationFailed"/>
              </restriction>
            </simpleType>
          </element>
          <element name="failureReason" minOccurs="0">
            <simpleType>
              <restriction base="string">
                <enumeration value="operation_failed"/>
                <enumeration value="operation_failed_invalid_input_parameter"/>
                <enumeration
value="operation_failed_unsupported_optional_input_parameter_notificationCategories"/>
                <enumeration
value="operation_failed_unsupported_optional_input_parameter_filter"/>
                <enumeration value="operation_failed_internal_problem"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </complexType>
    </element>
    <!-- getLogRecords Fault -->
    <element name="getLogRecordsFault">
      <simpleType>
        <restriction base="string">
          <enumeration value="OperationFailed"/>
        </restriction>
      </simpleType>
```

```
        </element>
        <!-- getLogSubscriptionIds Request -->
        <element name="getLogSubscriptionIdsRequest">
          </element>
        <!-- getLogSubscriptionIds Response -->
        <element name="getLogSubscriptionIdsResponse">
          <complexType>
            <sequence>
              <element name="logSubscriptionIds">
                <complexType>
                  <sequence>
                    <element name="logSubscriptionId" type="xl:LogSubscriptionId" minOccurs="0"
maxOccurs="unbounded"/>
                  </sequence>
                </complexType>
              </element>
              <element name="status">
                <simpleType>
                  <restriction base="string">
                    <enumeration value="OperationSucceeded"/>
                    <enumeration value="OperationFailed"/>
                  </restriction>
                </simpleType>
              </element>
            </sequence>
          </complexType>
        </element>
        <!-- getLogSubscriptionIds Fault -->
        <element name="getLogSubscriptionIdsFault">
          <simpleType>
            <restriction base="string">
              <enumeration value="OperationFailed"/>
            </restriction>
          </simpleType>
        </element>
        <!-- getLogSubscriptionStatus Request -->
        <element name="getLogSubscriptionStatusRequest">
          <complexType>
            <sequence>
              <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
            </sequence>
          </complexType>
        </element>
        <!-- getLogSubscriptionStatus Response -->
        <element name="getLogSubscriptionStatusResponse">
          <complexType>
            <sequence>
              <element name="logAttributeList">
                <complexType>
                  <sequence>
                    <element name="logSubscriptionId" type="xl:LogSubscriptionId"/>
                    <element name="loggingEndTime" type="nlIRPData:loggingEndTimeType" minOccurs="0"/>
                    <element name="maxSize" type="nonNegativeInteger" minOccurs="0"/>
                    <element name="currentSize" type="nonNegativeInteger" minOccurs="0"/>
                    <element name="creationTime" type="dateTime" minOccurs="0"/>
                    <element name="logState">
                      <simpleType>
                        <restriction base="string">
                          <enumeration value="logging"/>
                          <enumeration value="logFull"/>
                          <enumeration value="stopped"/>
                        </restriction>
                      </simpleType>
                    </element>
                    <element name="logRecordCount" type="nonNegativeInteger" minOccurs="0"/>
                    <element name="notificationCategories" type="xl:NotificationCategories"
minOccurs="0"/>
                    <element name="filter" type="string" minOccurs="0"/>
                    <element name="logFullAction" type="xl:LogFullAction"/>
                    <element name="occupancyLevels">
                      <complexType>
                        <choice>
                          <element name="level" type="nonNegativeInteger"/>
                          <sequence>
                            <element name="level1" type="nonNegativeInteger"/>
                            <element name="level2" type="nonNegativeInteger"/>
                            <element name="level3" type="nonNegativeInteger"/>
                          </sequence>
```

```xml
                    </choice>
                  </complexType>
                </element>
              </sequence>
            </complexType>
          </element>
          <element name="status">
            <simpleType>
              <restriction base="string">
                <enumeration value="OperationSucceeded"/>
                <enumeration value="OperationFailed"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </complexType>
    </element>
    <!-- getLogSubscriptionStatus Fault -->
    <element name="getLogSubscriptionStatusFault">
      <simpleType>
        <restriction base="string">
          <enumeration value="OperationFailed"/>
        </restriction>
      </simpleType>
    </element>
  </schema>
</types>
<message name="subscribeLogRequest">
  <part name="parameter" element="nlIRPData:subscribeLogRequest"/>
</message>
<message name="subscribeLogResponse">
  <part name="parameter" element="nlIRPData:subscribeLogResponse"/>
</message>
<message name="subscribeLogFault">
  <part name="parameter" element="nlIRPData:subscribeLogFault"/>
</message>
<message name="unsubscribeLogRequest">
  <part name="parameter" element="nlIRPData:unsubscribeLogRequest"/>
</message>
<message name="unsubscribeLogResponse">
  <part name="parameter" element="nlIRPData:unsubscribeLogResponse"/>
</message>
<message name="unsubscribeLogFault">
  <part name="parameter" element="nlIRPData:unsubscribeLogFault"/>
</message>
<message name="exportLogRecordsRequest">
  <part name="parameter" element="nlIRPData:exportLogRecordsRequest"/>
</message>
<message name="exportLogRecordsResponse">
  <part name="parameter" element="nlIRPData:exportLogRecordsResponse"/>
</message>
<message name="exportLogRecordsFault">
  <part name="parameter" element="nlIRPData:exportLogRecordsFault"/>
</message>
<message name="getLogRecordsRequest">
  <part name="parameter" element="nlIRPData:getLogRecordsRequest"/>
</message>
<message name="getLogRecordsResponse">
  <part name="parameter" element="nlIRPData:getLogRecordsResponse"/>
</message>
<message name="getLogRecordsFault">
  <part name="parameter" element="nlIRPData:getLogRecordsFault"/>
</message>
<message name="getLogSubscriptionIdsRequest">
  <part name="parameter" element="nlIRPData:getLogSubscriptionIdsRequest"/>
</message>
<message name="getLogSubscriptionIdsResponse">
  <part name="parameter" element="nlIRPData:getLogSubscriptionIdsResponse"/>
</message>
<message name="getLogSubscriptionIdsFault">
  <part name="parameter" element="nlIRPData:getLogSubscriptionIdsFault"/>
</message>
<message name="getLogSubscriptionStatusRequest">
  <part name="parameter" element="nlIRPData:getLogSubscriptionStatusRequest"/>
</message>
<message name="getLogSubscriptionStatusResponse">
  <part name="parameter" element="nlIRPData:getLogSubscriptionStatusResponse"/>
</message>
```

```
    <message name="getLogSubscriptionStatusFault">
      <part name="parameter" element="nlIRPData:getLogSubscriptionStatusFault"/>
    </message>
    <portType name="NLIRPOperations1">
      <operation name="subscribeLog">
        <input message="nlIRPSystem:subscribeLogRequest"/>
        <output message="nlIRPSystem:subscribeLogResponse"/>
        <fault name="subscribeLogFault" message="nlIRPSystem:subscribeLogFault"/>
      </operation>
      <operation name="unsubscribeLog">
        <input message="nlIRPSystem:unsubscribeLogRequest"/>
        <output message="nlIRPSystem:unsubscribeLogResponse"/>
        <fault name="unsubscribeLogFault" message="nlIRPSystem:unsubscribeLogFault"/>
      </operation>
      <operation name="exportLogRecords">
        <input message="nlIRPSystem:exportLogRecordsRequest"/>
        <output message="nlIRPSystem:exportLogRecordsResponse"/>
        <fault name="exportLogRecordsFault" message="nlIRPSystem:exportLogRecordsFault"/>
      </operation>
      <operation name="getLogRecords">
        <input message="nlIRPSystem:getLogRecordsRequest"/>
        <output message="nlIRPSystem:getLogRecordsResponse"/>
        <fault name="getLogRecordsFault" message="nlIRPSystem:getLogRecordsFault"/>
      </operation>
    </portType>
    <portType name="NLIRPOperations2">
      <operation name="getLogSubscriptionIds">
        <input message="nlIRPSystem:getLogSubscriptionIdsRequest"/>
        <output message="nlIRPSystem:getLogSubscriptionIdsResponse"/>
        <fault name="getLogSubscriptionIdsFault" message="nlIRPSystem:getLogSubscriptionIdsFault"/>
      </operation>
      <operation name="getLogSubscriptionStatus">
        <input message="nlIRPSystem:getLogSubscriptionStatusRequest"/>
        <output message="nlIRPSystem:getLogSubscriptionStatusResponse"/>
        <fault name="getLogSubscriptionStatusFault"
message="nlIRPSystem:getLogSubscriptionStatusFault"/>
      </operation>
    </portType>
    <binding name="NLIRPOperations1" type="nlIRPSystem:NLIRPOperations1">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="subscribeLog">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#subscribeLog" style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
        <fault name="subscribeLogFault">
          <soap:fault name="subscribeLogFault" use="literal"/>
        </fault>
      </operation>
      <operation name="unsubscribeLog">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#unsubscribeLog"
style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
        <fault name="unsubscribeLogFault">
          <soap:fault name="unsubscribeLogFault" use="literal"/>
        </fault>
      </operation>
      <operation name="exportLogRecords">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#exportLogRecords"
style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
        <fault name="exportLogRecordsFault">
```

```
            <soap:fault name="exportLogRecordsFault" use="literal"/>
          </fault>
        </operation>
        <operation name="getLogRecords">
          <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#getLogRecords" style="document"/>
          <input>
            <soap:body use="literal"/>
          </input>
          <output>
            <soap:body use="literal"/>
          </output>
          <fault name="getLogRecordsFault">
            <soap:fault name="getLogRecordsFault" use="literal"/>
          </fault>
        </operation>
      </binding>
      <binding name="NLIRPOperations2" type="nlIRPSystem:NLIRPOperations2">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="getLogSubscriptionIds">
          <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#getLogSubscriptionIds"
style="document"/>
          <input>
            <soap:body use="literal"/>
          </input>
          <output>
            <soap:body use="literal"/>
          </output>
          <fault name="getLogSubscriptionIdsFault">
            <soap:fault name="getLogSubscriptionIdsFault" use="literal"/>
          </fault>
        </operation>
        <operation name="getLogSubscriptionStatus">
          <soap:operation
soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#getLogSubscriptionStatus"
style="document"/>
          <input>
            <soap:body use="literal"/>
          </input>
          <output>
            <soap:body use="literal"/>
          </output>
          <fault name="getLogSubscriptionStatusFault">
            <soap:fault name="getLogSubscriptionStatusFault" use="literal"/>
          </fault>
        </operation>
      </binding>
      <service name="NLIRPService">
        <port name="NLIRPOperations1Port" binding="nlIRPSystem:NLIRPOperations1">
          <soap:address location="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRP"/>
        </port>
        <port name="NLIRPOperations2Port" binding="nlIRPSystem:NLIRPOperations2">
          <soap:address location="http://www.3gpp.org/ftp/specs/archive/32_series/32.336#ntfLogIRP"/>
        </port>
        <port name="GenericIRPPort" binding="genericIRPSystem:GenericIRPBinding">
          <soap:address location="http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRP"/>
        </port>
        <port name="NotificationIRPNtfPort" binding="ntfIRPNtfSystem:NotificationIRPNtf">
          <soap:address
location="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#NotificationIRPNtf"/>
        </port>
      </service>
</definitions>
```

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2010-09	SA#49	SP-100515	--	--	Presentation to SA for Information and Approval	---	1.0.0
2010-10	--	--	--	--	Publication	1.0.0	10.0.0
2012-09-	SA#57	-	-	-	Automatic upgrade from previous Release version 10.0.0	10.0.0	11.0.0