

# ETSI TS 132 307 V7.1.0 (2009-01)

*Technical Specification*

**Digital cellular telecommunications system (Phase 2+);  
Universal Mobile Telecommunications System (UMTS);  
LTE;  
Telecommunication management;  
Configuration Management (CM);  
Notification Integration Reference Point (IRP):  
SOAP Solution Set (SS)  
(3GPP TS 32.307 version 7.1.0 Release 7)**



---

Reference

RTS/TSGS-0532307v710

---

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

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2009.  
All rights reserved.

**DECT**<sup>™</sup>, **PLUGTESTS**<sup>™</sup>, **UMTS**<sup>™</sup>, **TIPHON**<sup>™</sup>, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP**<sup>™</sup> is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**LTE**<sup>™</sup> is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM**<sup>®</sup> and the GSM logo are Trade Marks registered and owned by the GSM Association.

---

## 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

|   |    |
|---|----|
| Intellectual Property Rights .....                        | 2  |
| Foreword.....   | 2  |
| Foreword.....   | 5  |
| Introduction .....  | 5  |
| 1 Scope .....   | 7  |
| 2 References .....  | 7  |
| 3 Definitions and abbreviations.....                      | 8  |
| 3.1 Definitions .....                                     | 8  |
| 3.2 Abbreviations .....                                   | 8  |
| 4 Architectural features .....                            | 8  |
| 5 Mapping .....   | 10 |
| 5.1 Operation mapping .....                               | 10 |
| 5.2 Filter language .....                                 | 10 |
| 5.3 Common datatype definition .....                      | 10 |
| 5.3.1 NotificationCategorySetType .....                   | 10 |
| 5.3.2 SubscriptionStateType .....                         | 10 |
| 5.4 Operation parameter mapping .....                     | 11 |
| 5.4.1 Operation subscribe.....                            | 11 |
| 5.4.1.1 Input parameters.....                             | 11 |
| 5.4.1.2 Output parameters .....                           | 11 |
| 5.4.1.3 Fault definition.....                             | 11 |
| 5.4.2 Operation unsubscribe .....                         | 12 |
| 5.4.2.1 Input parameters.....                             | 12 |
| 5.4.2.2 Output parameters .....                           | 12 |
| 5.4.2.3 Fault definition .....                            | 12 |
| 5.4.3 Operation getSubscriptionIds.....                   | 13 |
| 5.4.3.1 Input parameters.....                             | 13 |
| 5.4.3.2 Output parameters .....                           | 13 |
| 5.4.3.3 Fault definition .....                            | 13 |
| 5.4.4 Operation getSubscriptionStatus.....                | 14 |
| 5.4.4.1 Input parameters.....                             | 14 |
| 5.4.4.2 Output parameters .....                           | 14 |
| 5.4.4.3 Fault definition.....                             | 14 |
| 5.4.5 Operation changeSubscriptionFilter .....            | 15 |
| 5.4.5.1 Input parameters.....                             | 15 |
| 5.4.5.2 Output parameters .....                           | 15 |
| 5.4.5.3 Fault definition .....                            | 15 |
| 5.4.6 Operation suspendSubscription .....                 | 15 |
| 5.4.6.1 Input parameters.....                             | 15 |
| 5.4.6.2 Output parameters .....                           | 16 |
| 5.4.6.3 Fault definition .....                            | 16 |
| 5.4.7 Operation resumeSubscription.....                   | 16 |
| 5.4.7.1 Input parameters.....                             | 16 |
| 5.4.7.2 Output parameters .....                           | 16 |
| 5.4.7.3 Fault definition.....                             | 17 |
| 5.4.8 Operation getNotificationCategories .....           | 17 |
| 5.4.8.1 Input parameters.....                             | 17 |
| 5.4.8.2 Output parameters .....                           | 17 |
| 5.4.8.3 Fault definition .....                            | 17 |
| 5.5 Parameter mapping .....                               | 17 |
| 5.6 NotificationIRPNotification Interface definition..... | 18 |
| 5.6.1 Input parameters .....                              | 18 |

|                               |  |           |
|-------------------------------|--|-----------|
| 5.6.2                         | Output parameters.....   | 18        |
| 6                             | IRP Agent's Behaviour.....                                     | 18        |
| 6.1                           | Subscription.....  | 18        |
| <b>Annex A (normative):</b>   | <b>WSDL specifications.....</b>                                | <b>19</b> |
| <b>Annex B (normative):</b>   | <b>Notification WSDL specifications.....</b>                   | <b>27</b> |
| <b>Annex C (informative):</b> | <b>Notification IRP WSDL/XML schema electronic files .....</b> | <b>28</b> |
| <b>Annex D (informative):</b> | <b>Change history .....</b>                                    | <b>29</b> |
| History .....                 |  | 30        |

---

## Foreword

This Technical Specification (TS) has been produced by the 3<sup>rd</sup> 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 3<sup>rd</sup> Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

- 32.301: Configuration Management (CM); Notification Integration Reference Point (IRP): Requirements
- 32.302: Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)
- 32.303: Configuration Management (CM); Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)
- 32.305: Configuration Management (CM); Notification Integration Reference Point (IRP): eXtensible Markup Language (XML) definition
- 32.307: Configuration Management (CM); Notification Integration Reference Point (IRP): SOAP Solution Set (SS)**

Configuration Management (CM), in general, provides the operator with the ability to assure correct and effective operation of the 3G network as it evolves. CM actions have the objective to control and monitor the actual configuration on the Network Elements (NEs) and Network Resources (NRs), and they may be initiated by the operator or by functions in the Operations Systems (OSs) or NEs.

CM actions may be requested as part of an implementation programme (e.g. additions and deletions), as part of an optimisation programme (e.g. modifications), and to maintain the overall Quality of Service (QoS). The CM actions are initiated either as a single action on a NE of the 3G network or as part of a complex procedure involving actions on many NEs.

The Itf-N interface is built up by a number of Integration Reference Points (IRPs) and a related Name Convention, which realise the functional capabilities over this interface. The basic structure of the IRPs is defined in 3GPP TS 32.101 [6] and 3GPP TS 32.102 [7].

Network Elements (NEs) under management and element managers generate notifications of events about occurrences within the network. Different kinds of events carry different kinds of information. For instance a new alarm as specified in 3GPP TS 32.111-2 Alarm IRP: Information Service [5], is one possible kind of event, an object creation as specified in 3GPP TS 32.602 Basic CM IRP: Information Service [15] is another possible kind of event.

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

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

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

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

---

# 1 Scope

The present document specifies the SOAP Solution Set for the IRP whose semantics is specified in Notification IRP: Information Service (3GPP TS 32.302 [4]).

This Solution Set specification is related to 3GPP TS 32.302 V7.0.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.) 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] ITU-T Recommendation X.736: "Information technology - Open Systems Interconnection - Systems Management: Security alarm reporting function".
- [2] ITU-T Recommendation X.734 (1992): "Information technology - Open Systems Interconnection - Systems management: Event report management function".
- [3] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".
- [4] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".
- [5] 3GPP TS 32.111-2: "Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service (IS)".
- [6] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".
- [7] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [8] 3GPP TS 32.301: "Telecommunication Management; Configuration Management (CM); Notification Integration Reference Point (IRP): Requirements".
- [9] 3GPP TS 32.317: "Telecommunication management; Generic Integration Reference Point (IRP) management; SOAP solution set".
- [10] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirements".
- [11] 3GPP TS 32.172: "Telecommunication management; Subscription Management (SuM) Network Resource Model (NRM) Integration Reference Point (IRP): Information Service".
- [12] W3C SOAP 1.1 specification (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>)
- [13] W3C XPath 1.0 specification (<http://www.w3.org/TR/1999/REC-xpath-19991116>)
- [14] W3C WSDL 1.1 specification (<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>)
- [15] 3GPP TS 32.602: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP): Information Service (IS)".



- [16] 3GPP TS 32.305: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): eXtensible Markup Language (XML) definition".
- [17] WS-I Basic Profile Version 1.1 (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>)
- [18] W3C SOAP 1.2 specification (<http://www.w3.org/TR/soap12-part1/>)

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [6], 3GPP TS 32.102 [7] and 3GPP TS 32.301 [8] and the following apply:

**IRP document version number string (or "IRPVersion"):** See 3GPP TS 32.311 [10].

### 3.2 Abbreviations

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

|      |  |
|------|--|
| CM   | Configuration Management                   |
| EM   | Element Manager                            |
| IS   | Information Service                        |
| NE   | Network Element                            |
| NV   | Name and Value pair                        |
| QoS  | Quality of Service                         |
| SS   | Solution Set                               |
| WSDL | Web Service Description Language           |
| WS-I | Web Services Interoperability Organization |

---

## 4 Architectural features

The overall architectural feature of Notification IRP is specified in 3GPP TS 32.302 [4]. This clause specifies features that are specific to the SOAP solution set.

The SOAP 1.1 specification [12] and WSDL 1.1 specification [14] are supported.

The SOAP 1.2 specification [18] is supported optionally.

This specification uses "document" style in WSDL file.

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

The IRPAgent shall support the push interface model that means IRPAgent sends notifications to IRPManager as soon as new events occur. IRPManager does not need to check ("pull") for events.

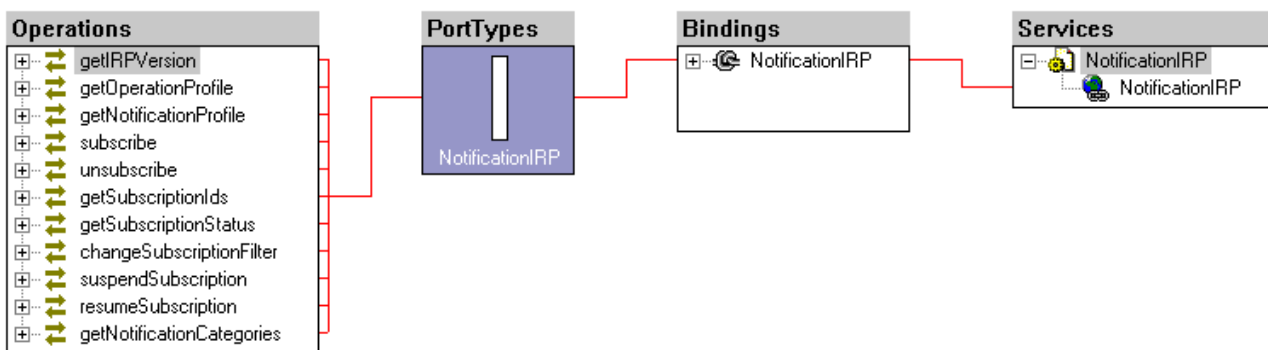
The 'Notification' definition is imported from Notification IRP XML Definition [16].

This specification uses a number of namespace prefixes throughout that are listed in Table 4.1.

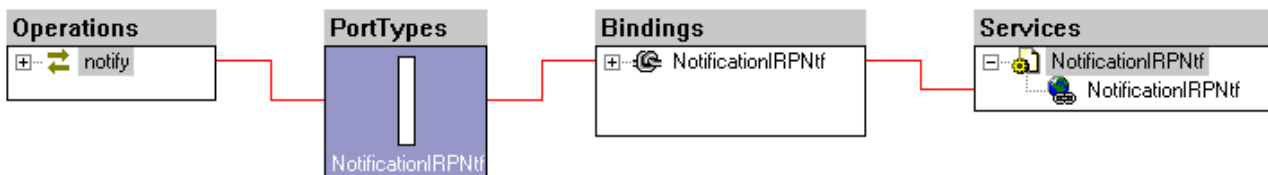
**Table 4.1: Prefixes and Namespaces used in this specification**

| PREFIX              | NAMESPACE   |
|---------------------|---|
| http                | http://schemas.xmlsoap.org/wsdl/http/   |
| soap                | http://schemas.xmlsoap.org/wsdl/soap/   |
| SOAP-ENV            | http://schemas.xmlsoap.org/soap/envelope/   |
| SOAP-ENC or soapenc | http://schemas.xmlsoap.org/soap/encoding/   |
| xs or xsd           | http://www.w3.org/2001/XMLSchema  |
| xsi                 | http://www.w3.org/2001/XMLSchema-instance   |
| ntfIRPSystem        | http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/NotificationIRPSystem                 |
| ntfIRPData          | http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/NotificationIRPData                   |
| ntfIRPNtfSystem     | http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/notification/NotificationIRPNtfSystem |
| ntfIRPNtfData       | http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/notification/NotificationIRPNtfData   |
| genericIRPSystem    | http://www.3gpp.org/ftp/Specs/archive/32_series/32.317/schema/32317-710/GenericIRPSystem                      |

The WSDL structure is like Figure 4.1:



**Figure 4.1: Notification IRP SOAP Solution Set WSDL structure**



**Figure 4.2: Notification IRP SOAP Solution Set Notification WSDL structure**

## 5 Mapping

### 5.1 Operation mapping

Notification IRP: IS (3GPP TS 32.302 [5]) defines semantics of operations visible across this IRP. These operations are the operations of the IOCs defined in 3GPP TS 32.302 [5].

Table 5.1 maps the operations defined in Notification IRP: IS (3GPP TS 32.302 [5]) to their equivalents (methods) in this Solution Set (SS). Specifically, the table 5.1 maps the operations of the IOCs defined in 3GPP TS 32.302 [5] to their equivalents in this SS. Since one of the IOCs, the NotificationIRP IOC, inherits from the ManagedGenericIRP IOC [4], the table 5.1 also maps the operations of ManagedGenericIRP IOC to their equivalents (methods) in this SS.

Table 5.1 also qualifies if a method is Mandatory (M) or Optional (O).

**Table 5.1: Mapping from IS Operation to SS Equivalents**

| IS Operations in 3GPP TS 32.302 [5]   | SS Operations             | Qualifier |
|---|---------------------------|-----------|
| subscribe   | subscribe                 | M         |
| unsubscribe   | unsubscribe               | M         |
| getSubscriptionIds  | getSubscriptionIds        | O         |
| getSubscriptionStatus   | getSubscriptionStatus     | O         |
| changeSubscriptionFilter  | changeSubscriptionFilter  | O         |
| suspendSubscription   | suspendSubscription       | O         |
| resumeSubscription  | resumeSubscription        | O         |
| getNotificationCategories   | getNotificationCategories | O         |
| getIRPVersion (see note.)   | getIRPVersion             | M         |
| getOperationProfile (see note.)   | getOperationProfile       | O         |
| getNotificationProfile (see note.)  | getNotificationProfile    | O         |
| NOTE: These 3 operations are operations of ManagedGenericIRP IOC specified in 3GPP TS 32.312 [4]. The NotificationIRP IOC of 3GPP TS 32.302 [5] inherits from it. |                           |           |

### 5.2 Filter language

The filter language used in the SS is the XPath Language (see W3C XPath 1.0 specification [13]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

### 5.3 Common datatype definition

#### 5.3.1 NotificationCategorySetType

```
<complexType name="NotificationCategorySetType">
  <complexContent>
    <extension base="ntfIRPData:VersionNumberSetType">
      </extension>
    </complexContent>
  </complexType>
```

#### 5.3.2 SubscriptionStateType

```
<simpleType name="SubscriptionStateType">
  <restriction base="string">
    <enumeration value="Suspended"/>
    <enumeration value="NotSuspended"/>
  </restriction>
</simpleType>
```

## 5.4 Operation parameter mapping

3GPP TS 32.302 [5] defines semantics of parameters carried in operations across the Notification IRP. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

### 5.4.1 Operation subscribe

#### 5.4.1.1 Input parameters

**Table 5.4.1.1: Mapping from IS subscribe input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter   | Qualifier |
|------------------------|---|-----------|
| managerReference       | anyURI managerReference                                       | M         |
| timeTick               | long timeTick   | O         |
| notificationCategories | ntfIRPData:NotificationCategorySetType notificationCategories | O         |
| filter                 | string filter (see note 1)                                    | O         |
|                        | anyURI ntfTransServiceNS (see note 2)                         | M         |

NOTE1: The grammar of the filter string is XPath defined by W3C XPath 1.0 specification [12].

NOTE2: This parameter is used to specify the name space of the Web Notification Transmission Service that will be used for sending notification.  
The 'http://www.3gpp.org/ftp/Specs/archive/32\_series/32307/schema/32307-710/notification' should be supported.

Here is the XML schema fragment of the subscribe request:

```
<!-- subscribe Request -->
<element name="subscribe">
<complexType>
<sequence>
<element name="managerReference" type="anyURI"/>
<element name="timeTick" type="long" minOccurs="0"/>
<element name="notificationCategories" type="ntfIRPData:NotificationCategorySetType"
minOccurs="0"/>
<element name="filter" type="string" minOccurs="0"/>
<element name="ntfTransServiceNS" type="anyURI"/>
</sequence>
</complexType>
</element>
```

#### 5.4.1.2 Output parameters

**Table 5.4.1.2: Mapping from IS subscribe output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter       | Qualifier |
|------------------------|---------------------------|-----------|
| subscriptionId         | string subscriptionId     | M         |
| status                 | ntfIRPData:subscribeFault | M         |

Here is the XML schema fragment of the subscribe response:

```
<!-- subscribe Response -->
<element name="subscribeResponse">
<complexType>
<sequence>
<element name="subscriptionId" type="string"/>
</sequence>
</complexType>
</element>
```

#### 5.4.1.3 Fault definition

```
<!-- subscribe Fault -->
<element name="subscribeFault">
<complexType>
<choice>
```

```

<element name="AlreadySubscribedFault" type="string"/>
<element name="AtLeastOneNotificationCategoryNotSupportedFault" type="string"/>
<element name="subscribeFault" type="string"/>
<element ref="ntfIRPData:InvalidParameterFault"/>
</choice>
</complexType>
</element>

```

## 5.4.2 Operation unsubscribe

### 5.4.2.1 Input parameters

**Table 5.4.2.1: Mapping from IS unsubscribe input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter     | Qualifier |
|------------------------|-------------------------|-----------|
| managerReference       | anyURI managerReference | M         |
| subscriptionId         | string subscriptionId   | O         |

Here is the XML schema fragment of the unsubscribe request:

```

<!-- unsubscribe Request -->
<element name="unsubscribe">
<complexType>
<sequence>
<element name="managerReference" type="anyURI"/>
<element name="subscriptionId" type="string" minOccurs="0"/>
</sequence>
</complexType>
</element>

```

### 5.4.2.2 Output parameters

**Table 5.4.2.2: Mapping from IS unsubscribe output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter         | Qualifier |
|------------------------|-----------------------------|-----------|
| Status                 | ntfIRPData:unsubscribeFault | M         |

Here is the XML schema fragment of the unsubscribe response:

```

<!-- unsubscribe Response -->
<element name="unsubscribeResponse">
</element>

```

### 5.4.2.3 Fault definition

```

<!-- unsubscribe Fault -->
<element name="unsubscribeFault">
<complexType>
<choice>
<element name="unsubscribeFault" type="string"/>
<element ref="ntfIRPData:InvalidParameterFault"/>
</choice>
</complexType>
</element>

```

## 5.4.3 Operation getSubscriptionIds

### 5.4.3.1 Input parameters

**Table 5.4.3.1: Mapping from IS getSubscriptionIds input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter     | Qualifier |
|------------------------|-------------------------|-----------|
| managerReference       | anyURI managerReference | M         |

Here is the XML schema fragment of the getSubscriptionIds request:

```
<!-- getSubscriptionIds Request -->
<element name="getSubscriptionIds">
  <complexType>
    <sequence>
      <element name="managerReference" type="anyURI"/>
    </sequence>
  </complexType>
</element>
```

### 5.4.3.2 Output parameters

**Table 5.4.3.2: Mapping from IS getSubscriptionIds output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter                  | Qualifier |
|------------------------|--------------------------------------|-----------|
| subscriptionIdSet      | sequence of string subscriptionIdSet | M         |
| status                 | ntfIRPData:getSubscriptionIdsFault   | M         |

Here is the XML schema fragment of the getSubscriptionIds response:

```
<!-- getSubscriptionIds Response -->
<element name="getSubscriptionIdsResponse">
  <complexType>
    <sequence>
      <element name="subscriptionIdSet">
        <complexType>
          <sequence>
            <element name="subscriptionId" type="string" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
```

### 5.4.3.3 Fault definition

```
<!-- getSubscriptionIds Fault -->
<element name="getSubscriptionIdsFault">
  <complexType>
    <choice>
      <element name="getSubscriptionIdsFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
```

## 5.4.4 Operation getSubscriptionStatus

### 5.4.4.1 Input parameters

**Table 5.4.4.1: Mapping from IS getSubscriptionStatus input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter   | Qualifier |
|------------------------|-----------------------|-----------|
| subscriptionId         | string subscriptionId | M         |

Here is the XML schema fragment of the getSubscriptionStatus request:

```
<!-- getSubscriptionStatus Request -->
<element name="getSubscriptionStatus">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>
```

### 5.4.4.2 Output parameters

**Table 5.4.4.2: Mapping from IS getSubscriptionStatus output parameters to SS equivalents**

| IS Operation parameter   | SS Method parameter   | Qualifier |
|--|---|-----------|
| notificationCategories   | ntfIRPData:NotificationCategorySetType notificationCategories | C         |
| filterInEffect   | string filterInEffect (see note)                              | O         |
| SubscriptionState  | ntfIRPData:SubscriptionStateType SubscriptionState            | O         |
| timeTick   | long timeTick   | O         |
| status   | ntfIRPData:getSubscriptionStatusFault                         | M         |
| NOTE: The grammar of the filter string is XPath defined by W3C XPath 1.0 specification [12]. |   |           |

Here is the XML schema fragment of the getSubscriptionStatus response:

```
<!-- getSubscriptionStatus Response -->
<element name="getSubscriptionStatusResponse">
  <complexType>
    <sequence>
      <element name="notificationCategories" type="ntfIRPData:NotificationCategorySetType"
        minOccurs="0"/>
      <element name="filterInEffect" type="string" minOccurs="0"/>
      <element name="SubscriptionState" type="ntfIRPData:SubscriptionStateType"
        minOccurs="0"/>
      <element name="timeTick" type="long" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```

### 5.4.4.3 Fault definition

```
<!-- getSubscriptionStatus Fault -->
<element name="getSubscriptionStatusFault">
  <complexType>
    <choice>
      <element name="getSubscriptionStatusFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
```

## 5.4.5 Operation changeSubscriptionFilter

### 5.4.5.1 Input parameters

**Table 5.4.5.1: Mapping from IS changeSubscriptionFilter input parameters to SS equivalents**

| IS Operation parameter   | SS Method parameter      | Qualifier |
|--|--------------------------|-----------|
| subscriptionId   | string subscriptionId    | M         |
| filter   | string filter (see note) | O         |
| NOTE: The grammar of the filter string is XPath defined by W3C XPath 1.0 specification [12]. |                          |           |

Here is the XML schema fragment of the changeSubscriptionFilter request:

```
<!-- changeSubscriptionFilter Request -->
<element name="changeSubscriptionFilter">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
      <element name="filter" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```

### 5.4.5.2 Output parameters

**Table 5.4.5.2: Mapping from IS changeSubscriptionFilter output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter                       | Qualifier |
|------------------------|---|-----------|
| status                 | ntfIRPData: changeSubscriptionFilterFault | M         |

Here is the XML schema fragment of the changeSubscriptionFilter response:

```
<!-- changeSubscriptionFilter Response -->
<element name="changeSubscriptionFilterResponse">
  </element>
```

### 5.4.5.3 Fault definition

```
<!-- changeSubscriptionFilter Fault -->
<element name="changeSubscriptionFilterFault">
  <complexType>
    <choice>
      <element name="changeSubscriptionFilterFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
```

## 5.4.6 Operation suspendSubscription

### 5.4.6.1 Input parameters

**Table 5.4.6.1: Mapping from IS suspendSubscription input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter   | Qualifier |
|------------------------|-----------------------|-----------|
| subscriptionId         | string subscriptionId | M         |

Here is the XML schema fragment of the suspendSubscription request:

```
<!-- suspendSubscription Request -->
<element name="suspendSubscription">
  <complexType>
```



```

    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>

```

### 5.4.6.2 Output parameters

**Table 5.4.6.2: Mapping from IS suspendSubscription output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter                 | Qualifier |
|------------------------|-------------------------------------|-----------|
| status                 | ntfIRPData:suspendSubscriptionFault | M         |

Here is the XML schema fragment of the suspendSubscription response:

```

<!-- suspendSubscription Response -->
<element name="suspendSubscriptionResponse">
</element>

```

### 5.4.6.3 Fault definition

```

<!-- suspendSubscription Fault -->
<element name="suspendSubscriptionFault">
  <complexType>
    <choice>
      <element name="suspendSubscriptionFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>

```

## 5.4.7 Operation resumeSubscription

### 5.4.7.1 Input parameters

**Table 5.4.7.1: Mapping from IS resumeSubscription input parameters to SS equivalents**

| IS Operation parameter | SS Method parameter   | Qualifier |
|------------------------|-----------------------|-----------|
| subscriptionId         | string subscriptionId | M         |

Here is the XML schema fragment of the resumeSubscription request:

```

<!-- resumeSubscription Request -->
<element name="resumeSubscription">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>

```

### 5.4.7.2 Output parameters

**Table 5.4.7.2: Mapping from IS resumeSubscription output parameters to SS equivalents**

| IS Operation parameter | SS Method parameter                | Qualifier |
|------------------------|------------------------------------|-----------|
| status                 | ntfIRPData:resumeSubscriptionFault | M         |

Here is the XML schema fragment of the resumeSubscription response:

```

<!-- resumeSubscription Response -->
<element name="resumeSubscriptionResponse">

```

```
</element>
```

### 5.4.7.3 Fault definition

```
<!-- resumeSubscription Fault -->
<element name="resumeSubscriptionFault">
  <complexType>
    <choice>
      <element name="resumeSubscriptionFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
```

## 5.4.8 Operation getNotificationCategories

### 5.4.8.1 Input parameters

None.

Here is the XML schema fragment of the getNotificationCategories request:

```
<!-- getNotificationCategories Request -->
<element name="getNotificationCategories">
</element>
```

### 5.4.8.2 Output parameters

**Table 5.4.8.2: Mapping from IS getNotificationCategories output parameters to SS equivalents**

| IS Operation parameter   | SS Method parameter   | Qualifier |
|--------------------------|---|-----------|
| NotificationCategoryList | ntfIRPData:NotificationCategorySetType NotificationCategoryList | M         |
| status                   | ntfIRPData:getNotificationCategoriesFault                       | M         |

Here is the XML schema fragment of the getNotificationCategories response:

```
<!-- getNotificationCategories Response -->
<element name="getNotificationCategoriesResponse">
  <complexType>
    <sequence>
      <element name="NotificationCategoryList" type="ntfIRPData:NotificationCategorySetType"/>
    </sequence>
  </complexType>
</element>
```

### 5.4.8.3 Fault definition

```
<!-- getNotificationCategories Fault -->
<element name="getNotificationCategoriesFault">
  <complexType>
    <choice>
      <element name="getNotificationCategoriesFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
    </choice>
  </complexType>
</element>
```

## 5.5 Parameter mapping

Notification IRP: IS (3GPP TS 32.302 [5]) defines the semantics of common attributes carried in notifications. This SS does not provide the mapping of these attributes to their SOAP SS equivalents. Other IRPs such as Alarm IRP: IS (3GPP TS 32.111-2 [6]) identify and qualify these common attributes for use in their environment. Their corresponding SS documents define the mapping of these attributes to their SS equivalents.

## 5.6 NotificationIRPNotification Interface definition

The operation name is defined as 'notify'.

### 5.6.1 Input parameters

**Table 5.6.1: notify input parameters**

| IS Operation parameter | SS Method parameter          | Qualifier |
|------------------------|------------------------------|-----------|
| -                      | xe:Notification notification | M         |
|                        |                              |           |

Here is the XML schema fragment of the notify request:

```
<!-- notify Request -->
<element name="notify">
  <complexType>
    <sequence>
      <element name="notification" type="xe:Notification"/>
    </sequence>
  </complexType>
</element>
```

### 5.6.2 Output parameters

None.

---

## 6 IRPAgent's Behaviour

This clause describes some IRPAgent's behaviour not captured by WSDL.

### 6.1 Subscription

IRPManager can invoke multiple `subscribe` operation using different `managerReference(s)`. As far as IRPAgent is concerned, the IRPAgent will emit notifications to multiple "places" with their independent filter requirements. IRPAgent will not know if the notifications are going to the same IRPManager.

If IRPManager invokes multiple `subscribe` using the same `managerReference` and with an already subscribed `notificationCategory`, IRPAgent shall raise `AlreadySubscribedFault` fault to all invocations except one.

IRPManager can invoke multiple `subscribe` using the same `managerReference` and with one or more not-yet-subscribed `notificationCategories`. In this case, if IRPAgent supports all the notification categories requested, IRPAgent shall accept the invocation; otherwise, it raises `AtLeastOneNotificationCategoryNotSupportedFault` fault.

## Annex A (normative): WSDL specifications

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:genericIRPSystem="http://www.3gpp.org/ftp/Specs/archive/32_series/32.317/schema/32317-
710/GenericIRPSystem"
xmlns:ntfIRPSystem="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/NotificationIRPSystem"
xmlns:ntfIRPData="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/NotificationIRPData"
  targetNamespace="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/NotificationIRPSystem">
<import namespace="http://www.3gpp.org/ftp/Specs/archive/32_series/32.317/schema/32317-
710/GenericIRPSystem" location="http://www.3gpp.org/ftp/Specs/archive/32_series/32.317/schema/32317-
710-wsdl.zip"/>
  <types>
    <schema
targetNamespace="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/NotificationIRPData" xmlns="http://www.w3.org/2001/XMLSchema">
      <!-- subscribe Request -->
      <element name="subscribe">
        <complexType>
          <sequence>
            <element name="managerReference" type="anyURI"/>
            <element name="timeTick" type="long" minOccurs="0"/>
            <element name="notificationCategories"
type="ntfIRPData:NotificationCategorySetType" minOccurs="0"/>
            <element name="filter" type="string" minOccurs="0"/>
            <element name="ntfTransServiceNS" type="anyURI"/>
          </sequence>
        </complexType>
      </element>
      <!-- subscribe Response -->
      <element name="subscribeResponse">
        <complexType>
          <sequence>
            <element name="subscriptionId" type="string"/>
          </sequence>
        </complexType>
      </element>
      <complexType name="NotificationCategorySetType">
        <complexContent>
          <extension base="ntfIRPData:VersionNumberSetType">
          </extension>
        </complexContent>
      </complexType>
      <!-- subscribe Fault -->
      <element name="subscribeFault">
        <complexType>
          <choice>
            <element name="AlreadySubscribedFault" type="string"/>
            <element name="AtLeastOneNotificationCategoryNotSupportedFault"
type="string"/>
            <element name="subscribeFault" type="string"/>
            <element ref="ntfIRPData:InvalidParameterFault"/>
          </choice>
        </complexType>
      </element>
      <!-- unsubscribe Request -->
      <element name="unsubscribe">
        <complexType>
          <sequence>
            <element name="managerReference" type="anyURI"/>
            <element name="subscriptionId" type="string" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
      <!-- unsubscribe Response -->
      <element name="unsubscribeResponse">
      </element>
    </schema>
  </types>

```

```

<!-- unsubscribe Fault -->
<element name="unsubscribeFault">
  <complexType>
    <choice>
      <element name="unsubscribeFault" type="string"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
<!-- getSubscriptionIds Request -->
<element name="getSubscriptionIds">
  <complexType>
    <sequence>
      <element name="managerReference" type="anyURI"/>
    </sequence>
  </complexType>
</element>
<!-- getSubscriptionIds Response -->
<element name="getSubscriptionIdsResponse">
  <complexType>
    <sequence>
      <element name="subscriptionIdSet">
        <complexType>
          <sequence>
            <element name="subscriptionId" type="string"
maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
<!-- getSubscriptionIds Fault -->
<element name="getSubscriptionIdsFault">
  <complexType>
    <choice>
      <element name="getSubscriptionIdsFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
<!-- getSubscriptionStatus Request -->
<element name="getSubscriptionStatus">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>
<!-- getSubscriptionStatus Response -->
<element name="getSubscriptionStatusResponse">
  <complexType>
    <sequence>
      <element name="notificationCategories"
type="ntfIRPData:NotificationCategorySetType" minOccurs="0"/>
      <element name="filterInEffect" type="string" minOccurs="0"/>
      <element name="SubscriptionState" type="ntfIRPData:SubscriptionStateType"
minOccurs="0"/>
      <element name="timeTick" type="long" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<simpleType name="SubscriptionStateType">
  <restriction base="string">
    <enumeration value="Suspended"/>
    <enumeration value="NotSuspended"/>
  </restriction>
</simpleType>
<!-- getSubscriptionStatus Fault -->
<element name="getSubscriptionStatusFault">
  <complexType>
    <choice>
      <element name="getSubscriptionStatusFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>

```

```

</element>
<!-- changeSubscriptionFilter Request -->
<element name="changeSubscriptionFilter">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
      <element name="filter" type="string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<!-- changeSubscriptionFilter Response -->
<element name="changeSubscriptionFilterResponse">
</element>
<!-- changeSubscriptionFilter Fault -->
<element name="changeSubscriptionFilterFault">
  <complexType>
    <choice>
      <element name="changeSubscriptionFilterFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
<!-- suspendSubscription Request -->
<element name="suspendSubscription">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>
<!-- suspendSubscription Response -->
<element name="suspendSubscriptionResponse">
</element>
<!-- suspendSubscription Fault -->
<element name="suspendSubscriptionFault">
  <complexType>
    <choice>
      <element name="suspendSubscriptionFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
<!-- resumeSubscription Request -->
<element name="resumeSubscription">
  <complexType>
    <sequence>
      <element name="subscriptionId" type="string"/>
    </sequence>
  </complexType>
</element>
<!-- resumeSubscription Response -->
<element name="resumeSubscriptionResponse">
</element>
<!-- resumeSubscription Fault -->
<element name="resumeSubscriptionFault">
  <complexType>
    <choice>
      <element name="resumeSubscriptionFault" type="string"/>
      <element ref="ntfIRPData:OperationNotSupportedFault"/>
      <element ref="ntfIRPData:InvalidParameterFault"/>
    </choice>
  </complexType>
</element>
<!-- getNotificationCategories Request -->
<element name="getNotificationCategories">
</element>
<!-- getNotificationCategories Response -->
<element name="getNotificationCategoriesResponse">
  <complexType>
    <sequence>
      <element name="NotificationCategoryList"
type="ntfIRPData:NotificationCategorySetType"/>
    </sequence>
  </complexType>
</element>
<!-- getNotificationCategories Fault -->

```

```

    <element name="getNotificationCategoriesFault">
      <complexType>
        <choice>
          <element name="getNotificationCategoriesFault" type="string"/>
          <element ref="ntfIRPData:OperationNotSupportedFault"/>
        </choice>
      </complexType>
    </element>

    <element name="OperationNotSupportedFault" type="string"/>
    <element name="InvalidParameterFault" type="string"/>
    <simpleType name="VersionNumberType">
      <restriction base="string"/>
    </simpleType>
    <complexType name="VersionNumberSetType">
      <sequence>
        <element name="versionNumber" type="ntfIRPData:VersionNumberType"
maxOccurs="unbounded"/>
      </sequence>
    </complexType>
    <complexType name="ParameterSetType">
      <sequence>
        <element name="parameterName" type="string" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
    <complexType name="OperationType">
      <sequence>
        <element name="operationName" type="string"/>
        <element name="parameterSet" type="ntfIRPData:ParameterSetType"/>
      </sequence>
    </complexType>
    <complexType name="OperationSetType">
      <sequence>
        <element name="operation" type="ntfIRPData:OperationType"
maxOccurs="unbounded"/>
      </sequence>
    </complexType>
    <complexType name="NotificationType">
      <sequence>
        <element name="notificationName" type="string"/>
        <element name="parameterSet" type="ntfIRPData:ParameterSetType"/>
      </sequence>
    </complexType>
    <complexType name="NotificationSetType">
      <sequence>
        <element name="notification" type="ntfIRPData:NotificationType"
maxOccurs="unbounded"/>
      </sequence>
    </complexType>

  </schema>
</types>
<message name="subscribeRequest">
  <part name="parameter" element="ntfIRPData:subscribe"/>
</message>
<message name="subscribeResponse">
  <part name="parameter" element="ntfIRPData:subscribeResponse"/>
</message>
<message name="subscribeFault">
  <part name="parameter" element="ntfIRPData:subscribeFault"/>
</message>
<message name="unsubscribeRequest">
  <part name="parameter" element="ntfIRPData:unsubscribe"/>
</message>
<message name="unsubscribeResponse">
  <part name="parameter" element="ntfIRPData:unsubscribeResponse"/>
</message>
<message name="unsubscribeFault">
  <part name="parameter" element="ntfIRPData:unsubscribeFault"/>
</message>
<message name="getSubscriptionIdsRequest">
  <part name="parameter" element="ntfIRPData:getSubscriptionIds"/>
</message>
<message name="getSubscriptionIdsResponse">
  <part name="parameter" element="ntfIRPData:getSubscriptionIdsResponse"/>
</message>
<message name="getSubscriptionIdsFault">
  <part name="parameter" element="ntfIRPData:getSubscriptionIdsFault"/>

```

```

</message>
<message name="getSubscriptionStatusRequest">
  <part name="parameter" element="ntfIRPData:getSubscriptionStatus"/>
</message>
<message name="getSubscriptionStatusResponse">
  <part name="parameter" element="ntfIRPData:getSubscriptionStatusResponse"/>
</message>
<message name="getSubscriptionStatusFault">
  <part name="parameter" element="ntfIRPData:getSubscriptionStatusFault"/>
</message>
<message name="changeSubscriptionFilterRequest">
  <part name="parameter" element="ntfIRPData:changeSubscriptionFilter"/>
</message>
<message name="changeSubscriptionFilterResponse">
  <part name="parameter" element="ntfIRPData:changeSubscriptionFilterResponse"/>
</message>
<message name="changeSubscriptionFilterFault">
  <part name="parameter" element="ntfIRPData:changeSubscriptionFilterFault"/>
</message>
<message name="suspendSubscriptionRequest">
  <part name="parameter" element="ntfIRPData:suspendSubscription"/>
</message>
<message name="suspendSubscriptionResponse">
  <part name="parameter" element="ntfIRPData:suspendSubscriptionResponse"/>
</message>
<message name="suspendSubscriptionFault">
  <part name="parameter" element="ntfIRPData:suspendSubscriptionFault"/>
</message>
<message name="resumeSubscriptionRequest">
  <part name="parameter" element="ntfIRPData:resumeSubscription"/>
</message>
<message name="resumeSubscriptionResponse">
  <part name="parameter" element="ntfIRPData:resumeSubscriptionResponse"/>
</message>
<message name="resumeSubscriptionFault">
  <part name="parameter" element="ntfIRPData:resumeSubscriptionFault"/>
</message>
<message name="getNotificationCategoriesRequest">
  <part name="parameter" element="ntfIRPData:getNotificationCategories"/>
</message>
<message name="getNotificationCategoriesResponse">
  <part name="parameter" element="ntfIRPData:getNotificationCategoriesResponse"/>
</message>
<message name="getNotificationCategoriesFault">
  <part name="parameter" element="ntfIRPData:getNotificationCategoriesFault"/>
</message>

<portType name="NotificationIRP">
  <operation name="getIRPVersion">
    <input message="genericIRPSystem:getIRPVersionRequest"/>
    <output message="genericIRPSystem:getIRPVersionResponse"/>
    <fault name="getIRPVersionFault" message="genericIRPSystem:getIRPVersionFault"/>
  </operation>
  <operation name="getOperationProfile">
    <input message="genericIRPSystem:getOperationProfileRequest"/>
    <output message="genericIRPSystem:getOperationProfileResponse"/>
    <fault name="getOperationProfileFault"
message="genericIRPSystem:getOperationProfileFault"/>
  </operation>
  <operation name="getNotificationProfile">
    <input message="genericIRPSystem:getNotificationProfileRequest"/>
    <output message="genericIRPSystem:getNotificationProfileResponse"/>
    <fault name="getNotificationProfileFault"
message="genericIRPSystem:getNotificationProfileFault"/>
  </operation>
  <operation name="subscribe">
    <input message="ntfIRPSystem:subscribeRequest"/>
    <output message="ntfIRPSystem:subscribeResponse"/>
    <fault name="subscribeFault" message="ntfIRPSystem:subscribeFault"/>
  </operation>
  <operation name="unsubscribe">
    <input message="ntfIRPSystem:unsubscribeRequest"/>
    <output message="ntfIRPSystem:unsubscribeResponse"/>
    <fault name="unsubscribeFault" message="ntfIRPSystem:unsubscribeFault"/>
  </operation>
  <operation name="getSubscriptionIds">
    <input message="ntfIRPSystem:getSubscriptionIdsRequest"/>
    <output message="ntfIRPSystem:getSubscriptionIdsResponse"/>

```



```

        <fault name="getSubscriptionIdsFault" message="ntfIRPSystem:getSubscriptionIdsFault"/>
    </operation>
    <operation name="getSubscriptionStatus">
        <input message="ntfIRPSystem:getSubscriptionStatusRequest"/>
        <output message="ntfIRPSystem:getSubscriptionStatusResponse"/>
        <fault name="getSubscriptionStatusFault"
message="ntfIRPSystem:getSubscriptionStatusFault"/>
    </operation>
    <operation name="changeSubscriptionFilter">
        <input message="ntfIRPSystem:changeSubscriptionFilterRequest"/>
        <output message="ntfIRPSystem:changeSubscriptionFilterResponse"/>
        <fault name="changeSubscriptionFilterFault"
message="ntfIRPSystem:changeSubscriptionFilterFault"/>
    </operation>
    <operation name="suspendSubscription">
        <input message="ntfIRPSystem:suspendSubscriptionRequest"/>
        <output message="ntfIRPSystem:suspendSubscriptionResponse"/>
        <fault name="suspendSubscriptionFault" message="ntfIRPSystem:suspendSubscriptionFault"/>
    </operation>
    <operation name="resumeSubscription">
        <input message="ntfIRPSystem:resumeSubscriptionRequest"/>
        <output message="ntfIRPSystem:resumeSubscriptionResponse"/>
        <fault name="resumeSubscriptionFault" message="ntfIRPSystem:resumeSubscriptionFault"/>
    </operation>
    <operation name="getNotificationCategories">
        <input message="ntfIRPSystem:getNotificationCategoriesRequest"/>
        <output message="ntfIRPSystem:getNotificationCategoriesResponse"/>
        <fault name="getNotificationCategoriesFault"
message="ntfIRPSystem:getNotificationCategoriesFault"/>
    </operation>

</portType>
<binding name="NotificationIRP" type="ntfIRPSystem:NotificationIRP">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getIRPVersion">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/getIRPVersion"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="getIRPVersionFault">
            <soap:fault name="getIRPVersionFault" use="literal"/>
        </fault>
    </operation>
    <operation name="getOperationProfile">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/getOperationProfile"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="getOperationProfileFault">
            <soap:fault name="getOperationProfileFault" use="literal"/>
        </fault>
    </operation>
    <operation name="getNotificationProfile">
        <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/getNotificationProfile"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="getNotificationProfileFault">
            <soap:fault name="getNotificationProfileFault" use="literal"/>
        </fault>
    </operation>
    <operation name="subscribe">

```

```

    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/subscribe"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="subscribeFault">
      <soap:fault name="subscribeFault" use="literal"/>
    </fault>
  </operation>
  <operation name="unsubscribe">
    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/unsubscribe"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="unsubscribeFault">
      <soap:fault name="unsubscribeFault" use="literal"/>
    </fault>
  </operation>
  <operation name="getSubscriptionIds">
    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/getSubscriptionIds"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="getSubscriptionIdsFault">
      <soap:fault name="getSubscriptionIdsFault" use="literal"/>
    </fault>
  </operation>
  <operation name="getSubscriptionStatus">
    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/getSubscriptionStatus"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="getSubscriptionStatusFault">
      <soap:fault name="getSubscriptionStatusFault" use="literal"/>
    </fault>
  </operation>
  <operation name="changeSubscriptionFilter">
    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/changeSubscriptionFilter"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
    <fault name="changeSubscriptionFilterFault">
      <soap:fault name="changeSubscriptionFilterFault" use="literal"/>
    </fault>
  </operation>
  <operation name="suspendSubscription">
    <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-710/suspendSubscription"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>

```

```
        <fault name="suspendSubscriptionFault">
          <soap:fault name="suspendSubscriptionFault" use="literal"/>
        </fault>
      </operation>
    <operation name="resumeSubscription">
      <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/resumeSubscription"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
      <fault name="resumeSubscriptionFault">
        <soap:fault name="resumeSubscriptionFault" use="literal"/>
      </fault>
    </operation>
    <operation name="getNotificationCategories">
      <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/getNotificationCategories"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
      <fault name="getNotificationCategoriesFault">
        <soap:fault name="getNotificationCategoriesFault" use="literal"/>
      </fault>
    </operation>
  </binding>
  <service name="NotificationIRP">
    <port name="NotificationIRP" binding="ntfIRPSys:NotificationIRP">
      <soap:address location="To be defined."/>
    </port>
  </service>
</definitions>
```

## Annex B (normative): Notification WSDL specifications

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ntfIRPNtfSystem="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/notification/NotificationIRPNtfSystem"
xmlns:ntfIRPNtfData="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/notification/NotificationIRPNtfData"
xmlns:xe="http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/32305#notification"
targetNamespace="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/notification/NotificationIRPNtfSystem">
  <types>
    <schema
targetNamespace="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/notification/NotificationIRPNtfData" xmlns="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://www.3gpp.org/ftp/specs/latest/rel-
6/32_series/32305#notification"
schemaLocation="http://www.3gpp.org/ftp/specs/archive/32_series/32.305/schema/32305-610-
XMLSchema.zip"/>

      <!-- notify Request -->
      <element name="notify">
        <complexType>
          <sequence>
            <element name="notification" type="xe:Notification"/>
          </sequence>
        </complexType>
      </element>

    </schema>
  </types>

  <message name="notifyRequest">
    <part name="parameter" element="ntfIRPNtfData:notify"/>
  </message>

  <portType name="NotificationIRPNtf">
    <operation name="notify">
      <input message="ntfIRPNtfSystem:notifyRequest"/>
    </operation>
  </portType>
  <binding name="NotificationIRPNtf" type="ntfIRPNtfSystem:NotificationIRPNtf">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="notify">
      <soap:operation
soapAction="http://www.3gpp.org/ftp/Specs/archive/32_series/32.307/schema/32307-
710/notification/notify"/>
      <input>
        <soap:body use="literal"/>
      </input>
    </operation>
  </binding>
  <service name="NotificationIRPNtf">
    <port name="NotificationIRPNtf" binding="ntfIRPNtfSystem:NotificationIRPNtf">
      <soap:address location="To be defined."/>
    </port>
  </service>
</definitions>

```

---

## Annex C (informative): Notification IRP WSDL/XML schema electronic files

The electronic files corresponding to the normative WSDL/XML schema defined in the present document are available in native form in the following archive:

[http://www.3gpp.org/ftp/Specs/archive/32\\_series/32307/schema/32307-710-wsdl.zip](http://www.3gpp.org/ftp/Specs/archive/32_series/32307/schema/32307-710-wsdl.zip)

## Annex D (informative): Change history

| Change history |       |           |      |    |  |     |       |       |
|----------------|-------|-----------|------|----|--|-----|-------|-------|
| Date           | TSG # | TSG Doc.  | CR   | R  | Subject/Comment  | Cat | Old   | New   |
| Mar 2006       | SA_31 | SP-060096 | --   | -- | Submitted to TSG SA#31 for Information   | --  | 1.0.0 |       |
| Dec 2006       | SA_34 | SP-060740 | --   | -- | Submitted to TSG SA#34 for Approval  | --  | 2.0.0 | 7.0.0 |
| Mar 2007       | --    | --        | --   | -- | Delete reference to the 32.304 CMIP SS. Reason: SA#35 endorsed the SA5 decision to not propagate the CMIP Solution Sets to Rel-7 (TS 32.3x4, TS 32.4x4, TS 32.6x4) | --  | 7.0.0 | 7.0.1 |
| Jun 2007       | --    | --        | --   | -- | Changed TS Title. Reason: SOAP does not stand anymore for "Simple Object Access Protocol"  | --  | 7.0.1 | 7.0.2 |
| Dec 2008       | SA_42 | SP-080845 | 0001 | -  | Incorrect URI specified in TS 32.307 (both word document and corresponding wsdl file)  | F   | 7.0.2 | 7.1.0 |

---

## History

| <b>Document history</b> |              |             |
|-------------------------|--------------|-------------|
| V7.0.1                  | March 2007   | Publication |
| V7.0.2                  | June 2007    | Publication |
| V7.1.0                  | January 2009 | Publication |
|                         |              |             |
|                         |              |             |