

# ETSI TS 132 666 V13.0.0 (2016-02)



TECHNICAL SPECIFICATION

**Digital cellular telecommunications system (Phase 2+);  
Universal Mobile Telecommunications System (UMTS);  
LTE;  
Telecommunication management;  
Configuration Management (CM);  
Kernel CM Integration Reference Point (IRP);  
Solution Set (SS) definitions  
(3GPP TS 32.666 version 13.0.0 Release 13)**



---

**Reference**

RTS/TSGS-0532666vd00

---

**Keywords**

GSM,LTE,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**

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (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:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2016.  
All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.  
**3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.  
**GSM®** 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 (<https://ipr.etsi.org>).

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>.

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Contents

Intellectual Property Rights .....	2
Foreword.....	2
Modal verbs terminology.....	2
Foreword.....	5
Introduction .....	5
1 Scope .....	6
2 References .....	6
3 Definitions and abbreviations.....	7
3.1 Definitions .....	7
3.2 Abbreviations .....	7
4 Solution Set definitions .....	7
<b>Annex A (normative): CORBA Solution Set .....</b>	<b>8</b>
A.1 Architectural Features .....	8
A.1.1 Syntax for Distinguished Names and Versions .....	8
A.1.2 Notifications .....	8
A.1.3 Filter language .....	8
A.2 Mapping .....	8
A.2.1 Operation and Notification mapping .....	8
A.2.2 Operation parameter mapping .....	9
A.2.3 Notification attribute mapping .....	9
A.3 Use of OMG Structured Event .....	11
A.4 Rules for NRM extensions .....	15
A.4.1 Extensions not allowed.....	15
A.5 Solution Set definitions .....	15
A.5.1 IDL definition structure .....	15
A.5.2 IDL specification 'KernelCmConstDefs.idl' .....	16
A.5.3 IDL specification 'KernelCmIRPSystem.idl' .....	18
A.5.4 IDL specification 'KernelCmIRPNotifications.idl' .....	19
<b>Annex B (normative): XML definitions .....</b>	<b>22</b>
B.1 Architectural features .....	22
B.1.1 Syntax for Distinguished Names .....	22
B.2 Mapping .....	22
B.3 Solution Set definitions .....	22
B.3.1 XML definition structure.....	22
B.3.2 Graphical Representation .....	22
B.3.3 XML Schema 'kernelNtf.xsd' .....	25
<b>Annex C (normative): SOAP Solution Set.....</b>	<b>28</b>
C.1 Architectural Features .....	28
C.1.1 Syntax for Distinguished Names and versions .....	28
C.1.2 Notifications .....	28
C.1.3 IRP document version number string .....	28
C.2 Mapping .....	29
C.2.1 General mappings.....	29

C.2.2	Operation and Notification mapping .....	29
C.2.3	Operation parameter mapping .....	30
C.2.3.1	Operation getNRMIRPVersion .....	30
C.2.3.1.1	Input parameters .....	30
C.2.3.1.2	Output parameters .....	30
C.2.2.1.3	Fault definition .....	30
C.3	Solution Set definitions .....	31
C.3.1	WSDL definition structure .....	31
C.3.2	Graphical Representation .....	31
C.3.3	WSDL specification 'KernelCMIRPSystem.wsdl' .....	32
<b>Annex D (informative): Change history .....</b>		<b>35</b>
History .....		36

---

# Foreword

This Technical Specification 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:

32.661	Configuration Management (CM); Kernel CM Requirements
32.662	Configuration Management (CM); Kernel CM Information Service (IS)
<b>32.666</b>	<b>Configuration Management (CM); Kernel CM Integration Reference Point (IRP); Solution Set (SS) definitions</b>

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 single actions on single NEs of the 3G network, or as part of a complex procedure involving actions on many resources/objects in one or several NEs.

---

# 1 Scope

The present document specifies the Solution Set definitions for the IRP whose semantics is specified in Kernel CM (Configuration Management) IRP: Information Service 3GPP TS 32.662 [7].

This Solution Set definitions specification is related to 3GPP TS 32.662 V12.0.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 TS 32.101: "Telecommunication management; Principles and high level requirements".
- [2] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [3] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".
- [4] 3GPP TS 32.662: "Telecommunication management; Configuration Management (CM); Kernel CM: Information Service (IS)".
- [5] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [6] Object Management Group 98 (November 1998): "*Notification Service: Joint Revised Submission OMG TC Document telecom/98-11-01*".
- [7] OMG CORBA Services (November 1996): "Common Object Services Specification".
- [8] The Common Object Request Broker: Architecture and Specification (for specification of valid version, see [1]).
- [9] 3GPP TS 32.306: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Solution Set (SS) definitions".
- [10] 3GPP TS 32.111-6: "Telecommunication management; Fault Management; Part 3: Alarm Integration Reference Point (IRP): Solution Set (SS) definitions".
- [11] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management: Information Service (IS)".
- [12] 3GPP TS 32.676: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Solution Set (SS) definitions".
- [13] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management: Requirements".
- [14] 3GPP TS 32.336: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Solution Set (SS) definitions".
- [15] 3GPP TS 32.331: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Requirements".

- [16] 3GPP TS 32.172: "Telecommunication management; Subscription Management (SuM) IRP; Network Resources Model (NRM)".
- [17] 3GPP TS 32.316: "Telecommunication management; Generic Integration Reference Point (IRP) management; Solution Set (SS) definitions".
- [18] W3C SOAP 1.1 specification (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>)
- [19] W3C WSDL 1.1 specification (<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>)
- [20] W3C XPath 1.0 specification (<http://www.w3.org/TR/1999/REC-xpath-19991116>)
- [21] WS-I Basic Profile Version 1.1 (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>)
- [22] 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 [1], 3GPP TS 32.102 [2], 3GPP TS 32.600 [3], 3GPP TS 32.662 [4] and 3GPP TS 32.331[15] apply.

**IRP document version number string (or "IRPVersion"):** See 3GPP TS 32.311 [13].

### 3.2 Abbreviations

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

CORBA	Common Object Request Broker Architecture
DN	Distinguished Name
IDL	Interface Definition Language (OMG)
IRP	Integration Reference Point
IS	Information Service
MO	Managed Object
MOC	Managed Object Class
NL	Notification Log
NRM	Network Resource Model
OMG	Object Management Group
SS	Solution Set
UML	Unified Modelling Language
VSE	Vendor Specific Extensions
XML	eXtensible Markup Language

---

## 4 Solution Set definitions

This specification defines the following 3GPP Kernel CM IRP Solution Set definitions:

- 3GPP Kernel CM IRP CORBA SS (Annex A)
- 3GPP Kernel CM IRP XML definitions (Annex B)
- 3GPP Kernel CM IRP SOAP Solution Set (Annex C)



---

## Annex A (normative): CORBA Solution Set

This annex contains the CORBA Solution Set for the IRP whose semantics is specified in Kernel CM IRP: Information Service (TS 32.662 [4]).

---

### A.1 Architectural Features

The overall architectural feature of Kernel Configuration Management IRP is specified in 3GPP TS 32.662 [4].

This clause specifies features that are specific to the CORBA SS.

#### A.1.1 Syntax for Distinguished Names and Versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [5].

The version of this IRP is represented as a string (see also clause 3.1).

#### A.1.2 Notifications

Notifications are sent according to the Notification IRP: CORBA SS (see 3GPP TS 32.303 [9]).

The contents of the Kernel CM IRP notifications are defined in the present document.

#### A.1.3 Filter language

The filter language used in the SS is the Extended Trader Constraint Language (see OMG Notification Service [6]). IRPAgents may throw a FilterComplexityLimit exception when a given filter is too complex.

---

## A.2 Mapping

### A.2.1 Operation and Notification mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) defines semantics of operation and notification visible across the Kernel Configuration Management IRP. The following table in this subclause indicates 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 32.662 [4])	SS Method	Qualifier
getNRMIRPVersion	get_nrm_irp_version	M
notifyObjectCreation	See Notification IRP: CORBA SS [9]	O
notifyObjectDeletion	See Notification IRP: CORBA SS [9]	O
notifyAttributeValueChange	See Notification IRP: CORBA SS [9]	O
notifyStateChange	See Notification IRP: CORBA SS [9]	O
getIRPVersion	get_kernel_cm_irp_versions	M
getOperationProfile	get_kernel_cm_irp_operations_profile	O
getNotificationProfile	get_kernel_cm_irp_notification_profile	O
notifyCMSynchronizationRecommended	See Notification IRP: CORBA SS [9]	O

## A.2.2 Operation parameter mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) defines semantics of parameters carried in operations across the Kernel Configuration Management IRP. The following tables in this subclause indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

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

IS Operation parameter	SS Method parameter	Qualifier
versionNumberList	ManagedGenericRPCConstDefs::VersionNumberSet version_number_list	M
vSEVersionNumberList	ManagedGenericRPCConstDefs::VersionNumberSet vse_version_number_list	M
status	Exceptions: GetNRMIRPVersion	M

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

IS Operation parameter	SS Method parameter	Qualifier
versionNumberList	Return value of type ManagedGenericRPCConstDefs::VersionNumberSet	M
status	exception GetKernelCmIRPVersionsException	M

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

IS Operation parameter	SS Method parameter	Qualifier
irpVersion	ManagedGenericRPCConstDefs::VersionNumber kernel_cm_irp_version	M
operationNameProfile, operationParameterProfile	Return value of type ManagedGenericRPCConstDefs::MethodList	M
status	Exceptions: GetKernelCmIRPOperationsProfileException, ManagedGenericRPSystem::OperationNotSupported, ManagedGenericRPSystem::InvalidParameter	M

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

IS Operation parameter	SS Method parameter	Qualifier
irpVersion	ManagedGenericRPCConstDefs::VersionNumber kernel_cm_irp_version	M
notificationNameProfile, notificationParameterProfile	Return value of type ManagedGenericRPCConstDefs::MethodList	M
status	Exceptions: GetKernelCmIRPNotificationProfileException, ManagedGenericRPSystem::OperationNotSupported, ManagedGenericRPSystem::InvalidParameter	M

## A.2.3 Notification attribute mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) identifies and defines the semantics of attributes for `notifyObjectCreation`, `notifyObjectDeletion`, `notifyAttributeValueChange`, `notifyStateChange` and `notifyCMSynchronizationRecommended` for use for its IRP. The following table in this subclause shows the mapping of the IS notifications to SS equivalents.

**Table A.2.3.1: Mapping from IS notifications to SS equivalents**

IS notifications in 3GPP TS 32.662 [4]	SS notifications	Qualifier
NotifyObjectCreation	push_structured_event	O
NotifyObjectDeletion	push_structured_event	O
NotifyAttributeValueChange	push_structured_event	O
NotifyStateChange	push_structured_event	O
NotifyCMSynchronizationRecommended	push_structured_event	O

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) also qualifies the attributes. The following tables in this subclause show the mapping of these IS attributes to SS equivalents.

**Table A.2.3.2: Mapping from IS Notification Header attributes to SS equivalent**

IS Attribute of Notification Header in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
objectClass, objectInstance	NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE	M
notificationId	NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID	M
eventTime	NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME	M
systemDN	NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN	O
notificationType	--	M

**Table A.2.3.3: Mapping from IS notifyObjectCreation attributes to SS equivalent OBJECT\_CREATION**

IS Attribute of notifyObjectCreation in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table A.2.3.2	M
correlatedNotifications	KernelCmNotifications::MOCreation::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications::MOCreation::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifications::MOCreation::SOURCE_INDICATOR	O
attributeList	KernelCmNotifications::MOCreation::InitialAttributeValues (contained in remainder_of_body)	O

**Table A.2.3.4: Mapping from IS notifyObjectDeletion attributes to SS equivalent OBJECT\_DELETION**

IS Attribute of notifyObjectDeletion in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table A.2.3.2	M
correlatedNotifications	KernelCmNotifications::MODEletion::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications::MODEletion::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifications::MODEletion::SOURCE_INDICATOR	O
attributeList	KernelCmNotifications::MODEletion::AttributeValues (contained in remainder_of_body)	O

**Table A.2.3.5: Mapping from IS notifyAttributeValueChange attributes to SS equivalent ATTRIBUTE\_VALUE\_CHANGE**

IS Attribute of notifyAttributeValueChange in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table A.2.3.2	M
correlatedNotifications	KernelCmNotifications::AttributeValueChange::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications::AttributeValueChange::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifications::AttributeValueChange::SOURCE_INDICATOR	O
attributeValueChangeDefinition	KernelCmNotifications::AttributeValueChange::ModifiedAttributeSet (contained in remainder_of_body)	M

**Table A.2.3.6: Mapping from IS `notifyCMSynchronizationRecommended` attributes to SS equivalent `REQUEST_CM_SYNCHRONIZATION`**

IS Attribute of <code>notifyCMSynchronizationRecommended</code> in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
<code>notificationHeader</code>	See table A.2.3.2	M
<code>baseMOClass</code>	<code>KernelCmNotifications::CMSynchronizationRecommended::BASE_MO_CLASS</code>	M
<code>baseMOInstance</code>	<code>KernelCmNotifications::CMSynchronizationRecommended::BASE_MO_INSTANCE</code>	M
<code>scope</code>	<code>KernelCmNotifications::CMSynchronizationRecommended::SCOPE</code>	M
<code>additionalText</code>	<code>KernelCmNotifications::CMSynchronizationRecommended::ADDITIONAL_TEXT</code>	O

**Table A.2.3.7: Mapping from IS `notifyStateChange` attributes to SS equivalent `STATE_CHANGE`**

IS Attribute of <code>notifyStateChange</code> in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
<code>notificationHeader</code>	See table A.2.3.2	M
<code>stateChange</code>	<code>StateManagementIRPConstDefs::AttributeNameValue</code> (see note)	M
<code>correlatedNotifications</code>	<code>KernelCmNotifications::StateChange::CORRELATED_NOTIFICATIONS</code>	O
<code>additionalText</code>	<code>KernelCmNotifications::StateChange::ADDITIONAL_TEXT</code>	O
<code>sourceIndicator</code>	<code>KernelCmNotifications::StateChange::SOURCE_INDICATOR</code>	O
NOTE:	The <code>stateChange</code> attribute is mapped into name-value pairs that contain the state identifier in the name and the new and optional old state values in the attribute field (See TS 32.676 [12] <code>StateManagementIRPConstDefs</code> IDL <code>&lt;State Name&gt;OldNewValue</code> structures).	

## A.3 Use of OMG Structured Event

In CORBA SS, OMG defined `StructuredEvent` (see OMG Notification Service [6]) is used to carry notifications. This clause identifies the OMG defined `StructuredEvent` attributes that carry the attributes of notifications defined in 3GPP TS 32.662 [4].

The composition of OMG Structured Event, as defined in OMG Notification Service [6], is:

```

Header
  Fixed Header
    domain_name
    type_name
    event_name
  Variable Header
Body
  filterable_body_fields
  remainder_of_body

```

The following table in this clause lists all OMG Structured Event attributes in its leftmost column. The second column identifies the SS attributes, if any, that shall be carried there.

Attributes that are denoted as "optional" may be absent from the OMG Structured Event. As an example, if the optional `additionalText` attribute is not used for a particular notification, then the IRP Agent may exclude `additionalText` from the filterable body fields for that particular notification. Individual notifications from the same IRP Agent may include or exclude the same optional attribute.

TableA.3.1: Use of OMG Structured Event

IS Parameters	OMG CORBA Structured Event attribute	Comment
There is no corresponding IS parameter	domain_name	It contains the supported SS document version. This version is defined by the string constant KernelCmIRPSystem::VERSION defined in this specification.
notificationType	type_name	It is an attribute of notificationHeader. It shall indicate one of the following: Object Creation, Object Deletion, Attribute Value Change, State Change and CM Synchronization Recommended. It is a string. Its value is either defined by KernelCmNotifications::MOCreation::EVENT_TYPE, KernelCmNotifications::MODeletion::EVENT_TYPE, KernelCmNotifications::AttributeValueChange::EVENT_TYPE, KernelCmNotifications::StateChange::EVENT_TYPE or KernelCmNotifications::CMSynchronizationRecommended::EVENT_TYPE
There is no corresponding IS parameter	event_name	It shall be set to an empty string.
There is no corresponding IS parameter	variable Header	
objectClass, objectInstance	One NV pair of filterable_body_fields	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 this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.306 [9]).
notificationId	One NV pair of remainder_of_body	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.306 [9]).
eventTime	One NV pair of filterable_body_fields	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.306 [9]).
systemDN	One NV pair of filterable_body_fields	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS [9].
correlatedNotifications	One NV pair of remainder_of_body	It is an attribute of the Object Creation, Object Deletion and Attribute Value Change notifications. Name of NV pair is the KernelCmNotifications::NotificationCommon::CORRELATED_NOTIFICATIONS. Value of NV pair is a NotificationIRPConstDefs::CorrelatedNotificationSet defined in 3GPP TS 32.306 [9].
additionalText	One NV pair of remainder_of_body	It is an attribute of the Object Creation, Object Deletion, Attribute Value Change and CM Synchronization Recommended notifications. Name of NV pair is the KernelCmNotifications::NotificationCommon::ADDITIONAL_TEXT. Value of NV pair is a string.

sourceIndicator	One NV pair of remainder_of_body	<p>It is an attribute of the Object Creation, Object Deletion and Attribute Value Change notifications.</p> <p>Name of NV pair is the KernelCmConstDefs::AttributeNameValue::SOURCE_INDICATOR.</p> <p>Value of NV pair is the KernelCmNotifications::NotificationCommon::RESOURCE_OPERATION or MANAGEMENT_OPERATION or UNKNOWN_OPERATION.</p>
attributeList	remainder_of_non_filterable_body (see 3GPP TS 32.306 [9])	<p>It is used to transport attribute information.</p> <p>For Object Creation notification, this is defined by KernelCmNotifications::MOCreation::InitialAttributeValues. For Object Deletion notification, this is defined by KernelCmNotifications::MODeletion::AttributeValues.</p> <p>The name component of InitialAttributeValues and AttributeValues will be set to attribute names defined in KernelCmNRMDefs.</p>
attributeValueChangeDefinition	remainder_of_non_filterable_body (see 3GPP TS 32.306 [9])	<p>For Attribute Value Change notification, this is defined by KernelCmNotifications::AttributeValueChange::ModifiedAttributeSet.</p> <p>The name component of ModifiedAttributeSet will be set to attribute name defined in KernelCmNRMDefs.</p>
StateManagementIRPConstDefs::AttributeNameValue	A set of up to 9 Name-value pairs See table A.3.2. All these 9 NV pairs are part of the remainder_of_body	<p>For state change notifications a series of up to 9 name-value pairs might be sent corresponding with the new and old values of each state/status attribute which has changed it's value.</p> <p>The new values of each state/status attributes that have changed are sent.</p> <p>The IRP agent may optionally send the old state/status changes. The name of the name-value pairs are defined by StateManagementIRPConstDefs::AttributeNameValue</p>
baseMOClass	One NV pair of remainder_of_body	<p>It is an attribute of the CMSynchronizationRecommended notification.</p> <p>Name of NV pair is a string,</p> <p>KernelCmNotifications::CMSynchronizationRecommended::BASE_MO_CLASS.</p> <p>Value of NV pair is a string. This string conveys the semantics of the Managed Object Class.</p>
baseMOInstance	One NV pair of remainder_of_body	<p>It is an attribute of the CMSynchronizationRecommended notification.</p> <p>Name of NV pair is the KernelCmNotifications::CMSynchronizationRecommended::BASE_MO_INSTANCE.</p> <p>Value of NV pair is a string. This is the DN string of the Managed Object Instance.</p>
scope	One NV pair of remainder_of_body	<p>It is an attribute of the CMSynchronizationRecommended notification.</p> <p>Name of NV pair is the KernelCmNotifications::CMSynchronizationRecommended::SCOPE.</p> <p>Value of NV pair is KernelCmConstDefs::ScopePara.</p>

**Table A.3.2 Name – value pairs for state change notifications**

<b>Name</b>	<b>Value</b>
OPERATIONAL_STATE	StateManagementIRPConstDefs::OperationalStateOldNewValue
USAGE_STATE	StateManagementIRPConstDefs::UsageStateOldNewValue
ADMINISTRATIVE_STATE	StateManagementIRPConstDefs::AdministrativeStateOldNewValue
ALARM_STATUS	StateManagementIRPConstDefs::AlarmStatusOldNewValue
PROCEDURAL_STATUS	StateManagementIRPConstDefs::ProceduralStatusOldNewValue
AVAILABILITY_STATUS	StateManagementIRPConstDefs::AvailabilityStatusOldNewValue
CONTROL_STATUS	StateManagementIRPConstDefs::ControlStatusOldNewValue
STANDBY_STATUS	StateManagementIRPConstDefs::StandbyStatusOldNewValue
UNKNOWN_STATUS	StateManagementIRPConstDefs::UnknownStatusOldNewValue

---

## A.4 Rules for NRM extensions

This clause discusses how the models and IDL definitions provided in the present document can be extended for a particular implementation and still remain compliant with 3GPP SA5's specifications.

### A.4.1 Extensions not allowed

The IDL specifications in the present document cannot be edited or altered. Any additional IDL specifications shall be specified in separate IDL files.

IDL interfaces (note: not MOCs) specified in the present document may not be subclassed or extended. New interfaces may be defined with vendor-specific methods.

---

## A.5 Solution Set definitions

### A.5.1 IDL definition structure

Clause A.5.2 defines the constants and types used by the Kernel CM IRP.

Clause A.5.3 defines the operations which are performed by the Kernel CM IRP agent.

Clause A.5.4 defines the notifications which are performed by the Kernel CM IRP agent.



## A.5.2 IDL specification 'KernelCmConstDefs.idl'

```
//File: KernelCmConstDefs.idl
#ifndef _KERNEL_CM_CONST_DEFS_IDL_
#define _KERNEL_CM_CONST_DEFS_IDL_

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

module KernelCmConstDefs
{
  /**
   * Information about one attribute
   * - name defines the name of the attribute
   * - value defines the value of the attribute
   *
   */
  struct MOAttribute
  {
    string name;
    any value;
  };

  /**
   * A set of attribute names and values
   */
  typedef sequence<MOAttribute> MOAttributeSet;

  /**
   * ScopeType defines the kind of scope to use in a CM synchronization
   * request together with ScopePara.level, in the SCOPE field.
   *
   * ScopePara.level is always >= 0. If a level is bigger than the
   * depth of the tree there will be no exceptions thrown.
   * BASE_ONLY: level ignored, just return the base object.
   * BASE_NTH_LEVEL: return all subordinate objects that are on "level"
   * distance from the base object, where 0 is the base object.
   * BASE_SUBTREE: return the base object and all of its subordinates
   * down to and including the nth level.
   * BASE_ALL: level ignored, return the base object and all of it's
   * subordinates.
   */
  enum ScopeType
  {
    BASE_ONLY,
    BASE_NTH_LEVEL,
    BASE_SUBTREE,
    BASE_ALL
  };

  struct ScopePara
  {
    ScopeType type;
    unsigned long level;
  };

  /* The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
  "Name Conventions for Managed Objects".
  */
  typedef string DN;

  typedef sequence <long> NotifIdSet;

  /*
  This holds identifiers of notifications that are correlated.
  */
  struct CorrelatedNotification
  {
    DN source; // Contains DN of MO that emitted the set of notifications
              // DN string format in compliance with Name Convention for
              // Managed Object.
              // This may be a zero-length string. In this case, the MO
              // is identified by the value of the MOI attribute
              // of the Structured Event, i.e., the notification.
    NotifIdSet notif_id_set; // Set of related notification ids
  };
};
```

```
/*
Correlated Notification sets are sets of Correlated Notification
structures.
*/
typedef sequence <CorrelatedNotification> CorrelatedNotificationSet;

/*
This block identifies attributes which are included as part of the Kernel
CM IRP. These attribute values should not clash with those defined for the
attributes of notification header (see IDL of Notification IRP).
*/
interface AttributeNameValue
{
    const string SOURCE_INDICATOR = "SOURCE";
    const string ADDITIONAL_TEXT = "ADD_TEXT";
    const string CORRELATED_NOTIFICATIONS = "CORREL_NOTIFS";
    const string BASE_MO_CLASS = "BASE_MOC";
    const string BASE_MO_INSTANCE = "BASE_MOI";
    const string SCOPE = "SCOPE";
};

};

#endif // _KERNEL_CM_CONST_DEFS_IDL_
```

### A.5.3 IDL specification 'KernelCmIRPSystem.idl'

```

//File: KernelCmIRPSystem.idl
#ifndef _KERNEL_CM_IRP_SYSTEM_IDL_
#define _KERNEL_CM_IRP_SYSTEM_IDL_

#include <ManagedGenericIRPConstDefs.idl>
#include <ManagedGenericIRPSystem.idl>

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

module KernelCmIRPSystem
{
    exception GetKernelCMIRPNotificationProfileException { string reason; };
    exception GetKernelCMIRPOperationsProfileException { string reason; };
    exception GetNRMIRPVersion { string reason; };
    exception GetKernelCMIRPVersionsException { string reason; };

    /**
     * The KernelCmIrpOperations interface.
     * Supports a number of Resource Model versions.
     */
    interface KernelCmIrpOperations
    {
        /**
         * Return the list of all supported Kernel CM IRP versions.
         */
        ManagedGenericIRPConstDefs::VersionNumberSet get_kernel_cm_irp_versions (
        )
        raises (GetKernelCMIRPVersionsException);

        /**
         * Get the version(s) of the interface
         *
         * @raises GetNRMIRPVersion when the system for some reason
         * can not return the supported versions.
         * @returns all supported versions.
         */
        void get_nrm_irp_version
        (
            out ManagedGenericIRPConstDefs::VersionNumberSet version_number_list,
            out ManagedGenericIRPConstDefs::VersionNumberSet vse_version_number_list
        )
        raises (GetNRMIRPVersion);

        /**
         * Return the list of all supported operations and their supported
         * parameters for a specific KernelCM IRP version.
         */
        ManagedGenericIRPConstDefs::MethodList get_kernel_cm_irp_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber kernel_cm_irp_version
        )
        raises (GetKernelCMIRPOperationsProfileException,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

        /**
         * Return the list of all supported notifications and their supported
         * parameters for a specific KernelCM IRP version.
         */
        ManagedGenericIRPConstDefs::MethodList
        get_kernel_cm_irp_notification_profile
        (
            in ManagedGenericIRPConstDefs::VersionNumber kernel_cm_irp_version
        )
        raises (GetKernelCMIRPNotificationProfileException,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
    };
};
#endif // _KERNEL_CM_IRP_SYSTEM_IDL_

```

## A.5.4 IDL specification 'KernelCmIRPNotifications.idl'

```
//File: KernelCmNotifications.idl
#ifndef _KERNEL_CM_NOTIFICATIONS_IDL_
#define _KERNEL_CM_NOTIFICATIONS_IDL_

#include <KernelCmConstDefs.idl>
#include <NotificationIRPNotifications.idl>

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

module KernelCmNotifications
{
  /**
   * This interface defines fields that are common for all
   * notification types.
   * All constants in the scope of this interface will be
   * visible in the interfaces that inherits this.
   * For instance constant
   * NotificationCommon::MANAGED_OBJECT_CLASS
   * can be addressed by MODeletion::MANAGED_OBJECT_CLASS
   */
  interface NotificationCommon: NotificationIRPNotifications::Notify
  {
    /**
     * This constant defines the name of the
     * source indicator property.
     */
    const string SOURCE_INDICATOR =
      KernelCmConstDefs::AttributeNameValue::SOURCE_INDICATOR;

    /**
     * Valid values for the SOURCE_INDICATOR
     * property
     */
    const string RESOURCE_OPERATION = "RESOURCE OPERATION";
    const string MANAGEMENT_OPERATION = "MANAGEMENT OPERATION";
    const string SON_OPERATION = "SON OPERATION";
    const string UNKNOWN_OPERATION = "UNKNOWN";

    /**
     * This constant defines the name of the
     * additional text property.
     * The data type for the value of this property
     * is a string.
     */
    const string ADDITIONAL_TEXT =
      KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

    /**
     * This constant defines the name of the
     * correlated notifications property.
     * The value part of the property is
     * KernelCmConstDefs::CorrelatedNotificationSet
     */
    const string CORRELATED_NOTIFICATIONS =
      KernelCmConstDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
  };

  /**
   * Constant definitions for the MO deleted notification
   */
  interface MODeletion : NotificationCommon
  {
    const string EVENT_TYPE = "x7";

    /**
     * This information mapped into the remainder_of_body
     * in the StructuredEvent
     */
    typedef KernelCmConstDefs::MOAttributeSet AttributeValues;
  };
}
/**
```

```

* Constant definitions for the MO created notification
*/
interface MOCreation : NotificationCommon
{
    const string EVENT_TYPE = "x6";

    /**
     * This information mapped into the remainder_of_body
     * in the StructuredEvent
     */
    typedef KernelCmConstDefs::MOAttributeSet InitialAttributeValues;
};

/**
 * Constant definitions for the Attribute Value Change
 * notification
 */
interface AttributeValueChange : NotificationCommon
{
    const string EVENT_TYPE = "x8";

    /**
     * Information about modified attributes for
     * one MO instance.
     * - name defines the name of the attribute
     * - new_value defines the new value of the attribute
     * - old_value defines the previous value of the attribute
     * The value is optional, which means that it may contain
     * an empty any (null inserted in the any).
     */
    struct ModifiedAttribute
    {
        string name;
        any new_value;
        any old_value;
    };

    /**
     * This information mapped into the remainder_of_body
     * in the StructuredEvent.
     */
    typedef sequence<ModifiedAttribute> ModifiedAttributeSet;
};

/**
 * Constant definitions for the CM Synchronization Recommended notification
 */
interface CMSynchronizationRecommended: NotificationIRPNotifications::Notify
{
    const string EVENT_TYPE = "x9";

    /**
     * This constant defines the name of the
     * additional text property.
     * The data type for the value of this property
     * is a string.
     */
    const string ADDITIONAL_TEXT =
        KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

    /**
     * This constant defines the name of the
     * base MO class property.
     * The value part of this property will carry
     * the base MO class name as a string.
     */
    const string BASE_MO_CLASS =
        KernelCmConstDefs::AttributeNameValue::BASE_MO_CLASS;

    /**
     * This constant defines the name of the
     * base MO instance property.
     * The value part of this property will carry
     * the base MO distinguished name as a string.
     */
    const string BASE_MO_INSTANCE =
        KernelCmConstDefs::AttributeNameValue::BASE_MO_INSTANCE;
};

```

```
/**
 * This constant defines the name of the
 * scope property.
 * The data type for the value of this property
 * is KernelCmConstDefs::ScopePara.
 */
const string SCOPE =
    KernelCmConstDefs::AttributeNameValue::SCOPE;
};

/**
 * Constant definitions for the State Change notification
 */
interface StateChange : NotificationCommon
{
    const string EVENT_TYPE = "xA";
};

};

#endif // _KERNEL_CM_NOTIFICATIONS_IDL_
```

## Annex B (normative): XML definitions

The annex specifies the XML Definitions for the Kernel CM Integration Reference Point (IRP) as it applies to Itf-N, in accordance with Kernel CM IRP IS definitions [4] as well as Notification Log IRP XML Definitions [14].

### B.1 Architectural features

The overall architectural feature of Kernel CM IRP is specified in 3GPP TS 32.662 [4].

This clause specifies features that are specific to the XML Schema definitions.

#### B.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [6].

### B.2 Mapping

Not present in the current version of this specification.

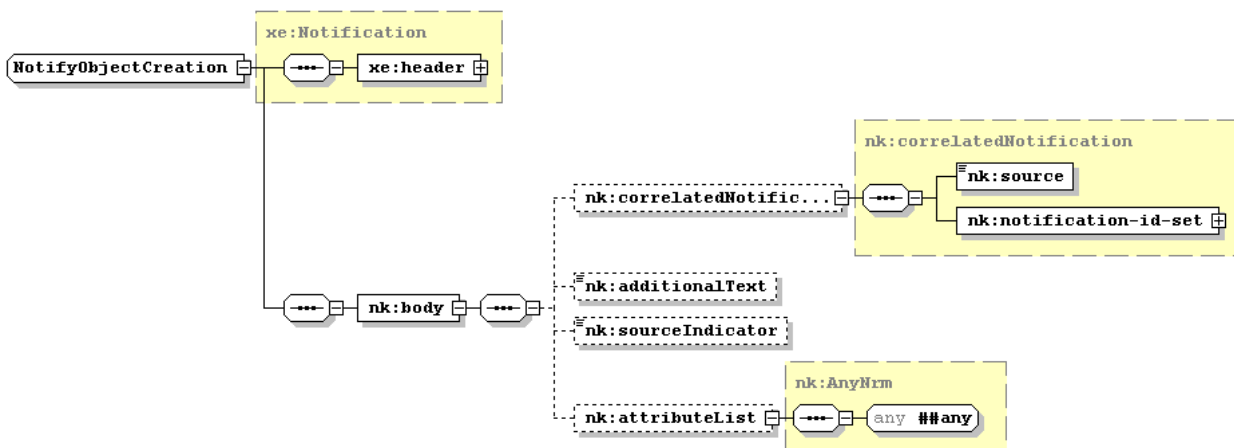
### B.3 Solution Set definitions

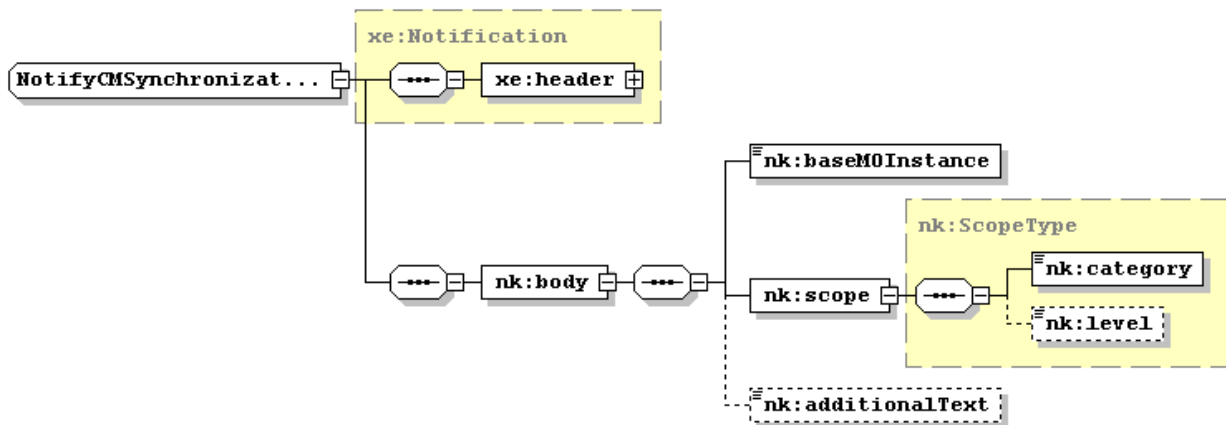
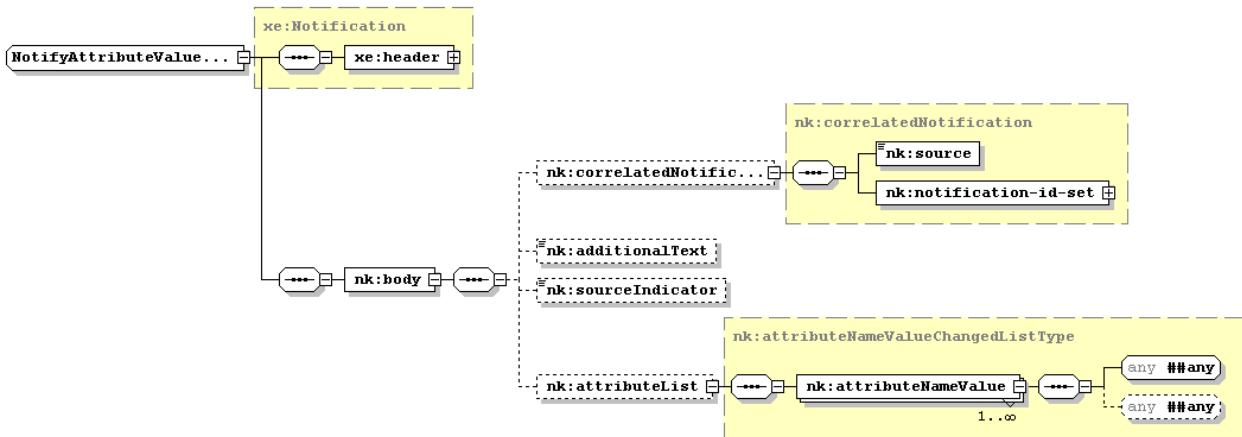
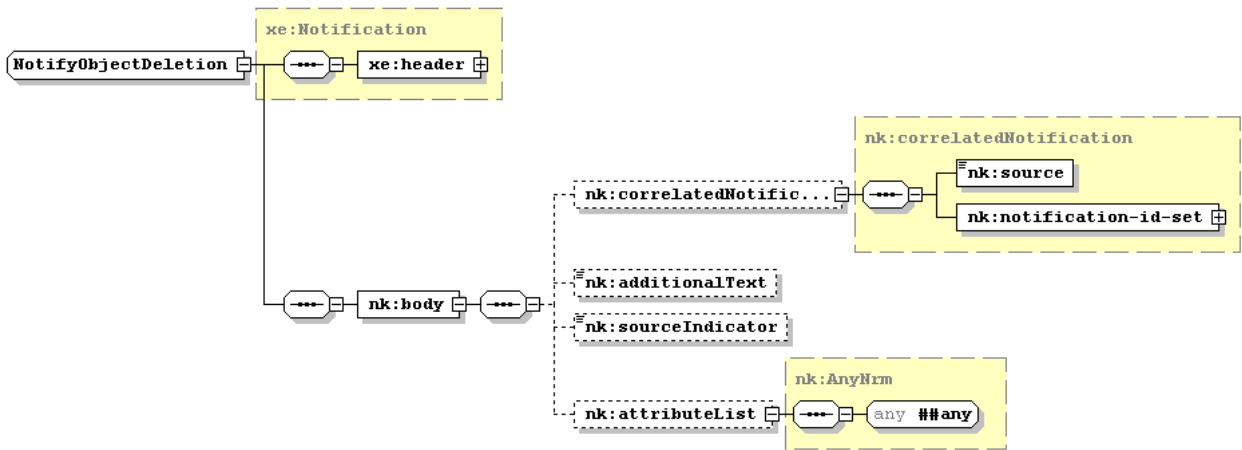
#### B.3.1 XML definition structure

Clause B.3.2 provides a graphical representation of the XML elements.

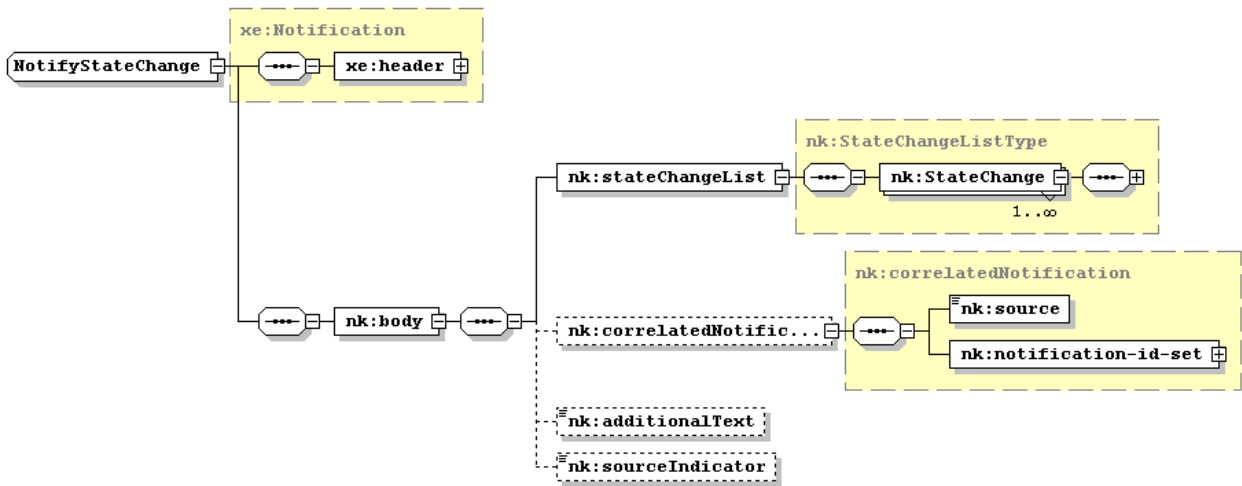
Clause B.3.3 provides XML definitions of Kernel CM IRP notifications as defined in 3GPP TS 32.662 [4].

#### B.3.2 Graphical Representation









### B.3.3 XML Schema 'kernelNtf.xsd'

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:nk="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#kernelNtf"
xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"
xmlns:sm="http://www.3gpp.org/ftp/specs/archive/32_series/32.676#stateManagementIRP"
xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32_series/32.626#genericNrm"
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#kernelNtf"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.306#notification"/>
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.676#stateManagementIRP"/>
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.626#genericNrm"/>
  <complexType name="correlatedNotification">
    <sequence>
      <element name="source" type="xn:dn"/>
      <element name="notification-id-set">
        <complexType>
          <sequence>
            <element name="notification-id" type="xe:NotificationId" minOccurs="0"
maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
  <simpleType name="SourceIndicatorType">
    <restriction base="string">
      <enumeration value="Resource_operation"/>
      <enumeration value="Management_operation"/>
      <enumeration value="SON_operation"/>
      <enumeration value="Unknown"/>
    </restriction>
  </simpleType>
  <complexType name="attributeNameValueChangedListType">
    <sequence>
      <element name="attributeNameValue" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <any namespace="##any" processContents="lax" id="newAttribute"/>
            <any namespace="##any" processContents="lax" id="oldAttribute" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
  <complexType name="ScopeType">
    <sequence>
      <element name="category" type="nk:ScopeCategoryType"/>
      <element name="level" type="unsignedLong" minOccurs="0"/>
    </sequence>
  </complexType>
  <simpleType name="ScopeCategoryType">
    <restriction base="string">
      <enumeration value="BASE_OBJECT_ONLY"/>
      <enumeration value="NTH_LEVEL_SUBORDINATES"/>
      <enumeration value="BASE_NTH_LEVEL"/>
      <enumeration value="BASE_ALL"/>
    </restriction>
  </simpleType>
  <complexType name="StateChangeListType">
    <sequence>
      <element name="StateChange" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <any
namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.676#stateManagementIRP"
processContents="strict" id="newState"/>
            <any
namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.676#stateManagementIRP"
processContents="strict" id="oldState" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>

```

```

<complexType name="AnyNrm">
  <sequence>
    <any namespace="##any" processContents="lax"/>
  </sequence>
</complexType>
<!-- notifyObjectCreation notification definition -->
<complexType name="NotifyObjectCreation">
  <complexContent>
    <extension base="xe:Notification">
      <sequence>
        <element name="body">
          <complexType>
            <sequence>
              <element name="correlatedNotification" type="nk:correlatedNotification"
minOccurs="0"/>
              <element name="additionalText" type="string" minOccurs="0"/>
              <element name="sourceIndicator" type="nk:SourceIndicatorType"
minOccurs="0"/>
              <element name="attributeList" type="nk:AnyNrm" minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- notifyObjectDeletion notification definition -->
<complexType name="NotifyObjectDeletion">
  <complexContent>
    <extension base="xe:Notification">
      <sequence>
        <element name="body">
          <complexType>
            <sequence>
              <element name="correlatedNotification" type="nk:correlatedNotification"
minOccurs="0"/>
              <element name="additionalText" type="string" minOccurs="0"/>
              <element name="sourceIndicator" type="nk:SourceIndicatorType"
minOccurs="0"/>
              <element name="attributeList" type="nk:AnyNrm" minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- notifyAttributeValueChange notification definition -->
<complexType name="NotifyAttributeValueChange">
  <complexContent>
    <extension base="xe:Notification">
      <sequence>
        <element name="body">
          <complexType>
            <sequence>
              <element name="correlatedNotification" type="nk:correlatedNotification"
minOccurs="0"/>
              <element name="additionalText" type="string" minOccurs="0"/>
              <element name="sourceIndicator" type="nk:SourceIndicatorType"
minOccurs="0"/>
              <element name="attributeList" type="nk:attributeNameValueChangedListType"
minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- notifyCMSynchronizationRecommended notification definition -->
<complexType name="NotifyCMSynchronizationRecommended">
  <complexContent>
    <extension base="xe:Notification">
      <sequence>
        <element name="body">
          <complexType>
            <sequence>
              <!--element name="baseMOClass" type="string"/-->

```

```

        <element name="baseMOInstance" type="string"/>
        <element name="scope" type="nk:ScopeType"/>
        <element name="additionalText" type="string" minOccurs="0"/>
    </sequence>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<!-- notifyStateChange notification definition -->
<complexType name="NotifyStateChange">
    <complexContent>
        <extension base="xe:Notification">
            <sequence>
                <element name="body">
                    <complexType>
                        <sequence>
                            <element name="stateChangeList" type="nk:StateChangeListType"/>
                            <element name="correlatedNotification" type="nk:correlatedNotification"
minOccurs="0"/>
                            <element name="additionalText" type="string" minOccurs="0"/>
                            <element name="sourceIndicator" type="nk:SourceIndicatorType"
minOccurs="0"/>
                        </sequence>
                    </complexType>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="NotifyObjectCreation" type="nk:NotifyObjectCreation"/>
<element name="NotifyObjectDeletion" type="nk:NotifyObjectDeletion"/>
<element name="NotifyAttributeValueChange" type="nk:NotifyAttributeValueChange"/>
<element name="NotifyCMSynchronizationRecommended" type="nk:NotifyCMSynchronizationRecommended"/>
<element name="NotifyStateChange" type="nk:NotifyStateChange"/>
</schema>

```

## Annex C (normative): SOAP Solution Set

The overall architectural feature of Kernel Configuration Management IRP is specified in 3GPP TS 32.662 [4].

This clause specifies features that are specific to the SOAP SS.

The SOAP 1.1 specification [18] and WSDL 1.1 specification [21] are supported.

The SOAP 1.2 specification [22] is supported optionally.

This specification uses "document" style in WSDL file.

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

This specification uses a number of namespace prefixes throughout that are listed in Table C.1.

**Table C.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
kernelCMIRPSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPSystem
kernelCMIRPData	http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPData
ntfIRPSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.306#NotificationIRPNtfSystem
genericIRPSystem	http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRPSystem

## C.1 Architectural Features

### C.1.1 Syntax for Distinguished Names and versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [5].

The version of this IRP is represented as a string (see also clause C.1.3).

### C.1.2 Notifications

Notifications are sent according to the Notification IRP: SOAP SS (see 3GPP TS 32.306 [9]).

The contents of the Kernel CM IRP notifications are defined in KernelCM IRP XML definition in Annex B.

### C.1.3 IRP document version number string

The IRP document version number (sometimes called "IRPVersion" or "SS version number") string is used to identify this specification. The string is derived using a rule described in 3GPP TS 32.311 [13]. The value of this string is defined by a constant in C.3.3.

This string (or sequence of strings, if more than one version is supported) is returned in `getIRPVersion` method and is carried in the first field of the notification header of all notifications related to this IRP.

## C.2 Mapping

### C.2.1 General mappings

The IS parameter name `managedObjectInstance` is mapped into DN.

Attributes modelling associations as defined in the NRM (here also called "reference attributes") are in this SS mapped to attributes. The names of the reference attributes in the NRM are mapped to the corresponding attribute names in the MOC. When the cardinality for an association is 0..1 or 1..1 the datatype for the reference attribute is defined as an `MOReference`. The value of an MO reference contains the distinguished name of the associated MO. When the cardinality for an association allows more than one referred MO, the reference attribute will be of type `MOReferenceSet`, which contains a sequence of MO references.

If a reference attribute is changed, an `AttributeValueChange` notification is emitted.

### C.2.2 Operation and Notification mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) defines semantics of operation and notification visible across the Kernel Configuration Management IRP. The following table in this subclause indicates mapping of these operations and notifications to their equivalents defined in this SS.

**Table C.2.2: Mapping from IS Notification/Operation to SS equivalents**

IS Operation/ notification (3GPP TS 32.662 [4])	SS Operation	Qualifier
<code>getNRMIRPVersion</code>	<code>getNRMIRPVersion</code>	M
<code>notifyObjectCreation</code>	<code>ntfIRPSystem:notify</code> (note 2)	O
<code>notifyObjectDeletion</code>	<code>ntfIRPSystem:notify</code> (note 2)	O
<code>notifyAttributeValueChange</code>	<code>ntfIRPSystem:notify</code> (note 2)	O
<code>notifyCMSynchronizationRecommended</code>	<code>ntfIRPSystem:notify</code> (note 2)	O
<code>notifyStateChange</code>	<code>ntfIRPSystem:notify</code> (note 2)	O
<code>getIRPVersion</code> (see note 1)	<code>getIRPVersion</code>	M
<code>getOperationProfile</code> (see note 1)	<code>getOperationProfile</code>	O
<code>getNotificationProfile</code> (see note 1)	<code>getNotificationProfile</code>	O
NOTE 1: This operation is of IOC <code>ManagedGenericIRP</code> specified in [18]. The IOC <code>KernelCmIRP</code> of [4] inherits from it.		
NOTE 2: The IS equivalent is mapped into parameters of <code>ntfIRPSystem::notify</code> .		

## C.2.3 Operation parameter mapping

The Kernel CM IRP: IS (3GPP TS 32.662 [4]) defines semantics of parameters carried in operations across the Kernel Configuration Management IRP. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

### C.2.3.1 Operation getNRMIRPVersion

#### C.2.3.1.1 Input parameters

None.

Here is the XML schema fragment of the getNRMIRPVersion request:

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

#### C.2.3.1.2 Output parameters

**Table C.2.3.1.2: Mapping from IS getNRMIRPVersion output parameters to SS equivalents**

IS Operation parameter	SS Operation parameter	Qualifier
versionNumberList	kernelCMIRPData:VersionNumberSetType versionNumberSet	M
vSEVersionNumberList	kernelCMIRPData:VersionNumberSetType vSEVersionNumberList	M
status	kernelCMIRPData:getNRMIRPVersionFault	M

Here is the XML schema fragment of the getNRMIRPVersion response:

```
<!-- getNRMIRPVersion Response -->
<element name="getNRMIRPVersionResponse">
  <complexType>
    <sequence>
      <element name="versionNumberList" type="kernelCMIRPData:VersionNumberSetType"/>
      <element name="vSEVersionNumberList" type="
kernelCMIRPData:VersionNumberSetType"/>
    </sequence>
  </complexType>
</element>
```

#### C.2.2.1.3 Fault definition

```
<!-- getNRMIRPVersion Fault -->
<element name="getNRMIRPVersionFault">
  <complexType>
    <choice>
      <element name="getNRMIRPVersionFault" type="string"/>
    </choice>
  </complexType>
</element>
```

## C.3 Solution Set definitions

### C.3.1 WSDL definition structure

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

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

### C.3.2 Graphical Representation

The WSDL structure is like Figure C.3.2:

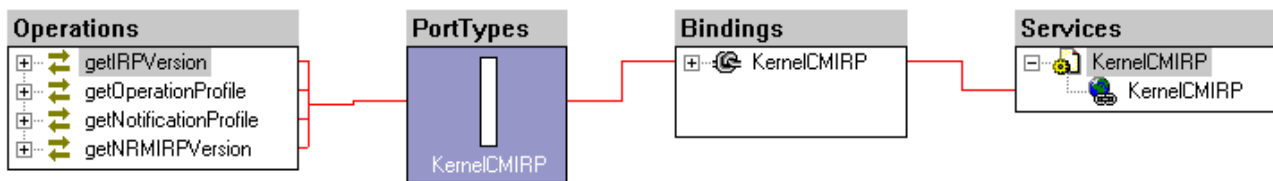


Figure C.3.2: kernelCM IRP SOAP Solution Set WSDL structure



### C.3.3 WSDL specification 'KernelCMIRPSystem.wsdl'

```

<?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/32.316#GenericIRPSystem"
xmlns:kernelCMIRPSystem="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPSystem"
xmlns:kernelCMIRPData="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPData"
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPSystem">
  <import namespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.316#GenericIRPSystem" />
  <types>
    <schema
targetNamespace="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#KernelCMIRPData"
xmlns="http://www.w3.org/2001/XMLSchema">
      <!-- getNRMIRPVersion Request -->
      <element name="getNRMIRPVersion">
      </element>
      <!-- getNRMIRPVersion Response -->
      <element name="getNRMIRPVersionResponse">
        <complexType>
          <sequence>
            <element name="versionNumberList"
type="kernelCMIRPData:VersionNumberSetType"/>
            <element name="vSEVersionNumberList"
type="kernelCMIRPData:VersionNumberSetType"/>
          </sequence>
        </complexType>
      </element>
      <!-- getNRMIRPVersion Fault -->
      <element name="getNRMIRPVersionFault">
        <complexType>
          <choice>
            <element name="getNRMIRPVersionFault" type="string"/>
          </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="kernelCMIRPData: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="kernelCMIRPData:ParameterSetType"/>
        </sequence>
      </complexType>
      <complexType name="OperationSetType">
        <sequence>
          <element name="operation" type="kernelCMIRPData:OperationType"
maxOccurs="unbounded"/>
        </sequence>
      </complexType>
      <complexType name="NotificationType">
        <sequence>
          <element name="notificationName" type="string"/>
          <element name="parameterSet" type="kernelCMIRPData:ParameterSetType"/>
        </sequence>
      </complexType>
      <complexType name="NotificationSetType">
        <sequence>
          <element name="notification" type="kernelCMIRPData:NotificationType"
maxOccurs="unbounded"/>

```

```

    </sequence>
  </complexType>

</schema>
</types>
<message name="getNRMIRPVersionRequest">
  <part name="parameter" element="kernelCMIRPData:getNRMIRPVersion"/>
</message>
<message name="getNRMIRPVersionResponse">
  <part name="parameter" element="kernelCMIRPData:getNRMIRPVersionResponse"/>
</message>
<message name="getNRMIRPVersionFault">
  <part name="parameter" element="kernelCMIRPData:getNRMIRPVersionFault"/>
</message>

<portType name="KernelCMIRP">
  <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="getNRMIRPVersion">
    <input message="kernelCMIRPSystem:getNRMIRPVersionRequest"/>
    <output message="kernelCMIRPSystem:getNRMIRPVersionResponse"/>
    <fault name="getNRMIRPVersionFault" message="kernelCMIRPSystem:getNRMIRPVersionFault"/>
  </operation>
</portType>
<binding name="KernelCMIRP" type="kernelCMIRPSystem:KernelCMIRP">
  <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/32.666#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/32.666#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/32.666#getNotificationProfile"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

```

```
<fault name="getNotificationProfileFault">
  <soap:fault name="getNotificationProfileFault" use="literal"/>
</fault>
</operation>
<operation name="getNRMIRPVersion">
  <soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32_series/32.666#
getNRMIRPVersion"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
  <fault name="getNRMIRPVersionFault">
    <soap:fault name="getNRMIRPVersionFault" use="literal"/>
  </fault>
</operation>
</binding>
<service name="KernelCMIRP">
  <port name="KernelCMIRP" binding="kernelCMIRPSystem:KernelCMIRP">
    <soap:address location="To be defined."/>
  </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-100519	--	--	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
2014-09	SA#65	SP-140559	001	-	Update the link from Solution Set to Information Service due to the end of Release 12	11.0.0	12.0.0
2016-01	-	-	-	-	Update to Rel-13 version (MCC)	12.0.0	<b>13.0.0</b>

---

# History

<b>Document history</b>		
V13.0.0	February 2016	Publication