3GPP TS 32.396 V16.0.0 (2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Telecommunication management;

Delta synchronization Integration Reference Point (IRP);

Solution Set (SS) definitions

(Release 16)

* *

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

GSM, UMTS, LTE, Management, CORBA, XML, SOAP

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***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.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [5](#__RefHeading___Toc335991961)

Introduction [5](#__RefHeading___Toc335991962)

1 Scope [6](#__RefHeading___Toc335991963)

2 References [6](#__RefHeading___Toc335991964)

3 Definitions and abbreviations [7](#__RefHeading___Toc335991965)

3.1 Definitions [7](#__RefHeading___Toc335991966)

3.2 Abbreviations [8](#__RefHeading___Toc335991967)

4 Solution Set definitions [8](#__RefHeading___Toc335991968)

Annex A (normative): CORBA Solution Set [9](#__RefHeading___Toc335991969)

A.1 Architectural features [9](#__RefHeading___Toc335991970)

A.1.1 Syntax for Distinguished Names [9](#__RefHeading___Toc335991971)

A.1.2 Notification Services [9](#__RefHeading___Toc335991972)

A.1.3 Push and Pull Style [9](#__RefHeading___Toc335991973)

A.1.4 Support multiple notifications in one push operation [9](#__RefHeading___Toc335991974)

A.1.5 Delta Synchronization Notification Interface [9](#__RefHeading___Toc335991975)

A.1.5.1 Method push (M) [9](#__RefHeading___Toc335991976)

A.2 Mapping [10](#__RefHeading___Toc335991977)

A.2.1 General mapping [10](#__RefHeading___Toc335991978)

A.2.2 Operation and notification mapping [10](#__RefHeading___Toc335991979)

A.2.3 Operation parameter mapping [10](#__RefHeading___Toc335991980)

A.2.4 Notification parameter mapping [13](#__RefHeading___Toc335991981)

A.3 Solution Set definitions [17](#__RefHeading___Toc335991982)

A.3.1 IDL definition structure [17](#__RefHeading___Toc335991983)

A.3.2 IDL specification (file name "DeltaSynchronizationConstDefs.idl") [18](#__RefHeading___Toc335991984)

A.3.3 IDL specification (file name "DeltaSynchronizationSystem.idl") [23](#__RefHeading___Toc335991985)

A.3.4 IDL specification (file name "DeltaSynchronizationNotifications.idl") [26](#__RefHeading___Toc335991986)

Annex B (normative): XML definitions [28](#__RefHeading___Toc335991987)

B.1 Architectural Features [28](#__RefHeading___Toc335991988)

B.1.1 Syntax for Distinguished Names [28](#__RefHeading___Toc335991989)

B.1.2 Notification Services [28](#__RefHeading___Toc335991990)

B.1.3 IOC definitions [28](#__RefHeading___Toc335991991)

B.2 Mapping [28](#__RefHeading___Toc335991992)

B.3 Solution Set definitions [28](#__RefHeading___Toc335991993)

B.3.1 XML definition structure [28](#__RefHeading___Toc335991994)

B.3.1.1 Global structure [29](#__RefHeading___Toc335991995)

B.3.1.2 XML elements fileHeader and fileFooter [29](#__RefHeading___Toc335991996)

B.3.1.2.1 XML elements fileHeaderForDeltaSynchForCM/AlarmData [29](#__RefHeading___Toc335991997)

B.3.1.2.2 XML element fileFooter [30](#__RefHeading___Toc335991998)

B.3.1.3 Delta synchronisation IRP specific XML elements [31](#__RefHeading___Toc335991999)

B.3.1.4 Delta synchronisation IRP XML File Name Conventions [31](#__RefHeading___Toc335992000)

B.3.2 Graphical Representation [32](#__RefHeading___Toc335992001)

B.3.3 XML Schema [35](#__RefHeading___Toc335992002)

B.3.3.1 XML Schema "deltaSynchGeneric.xsd" [35](#__RefHeading___Toc335992003)

B.3.3.2 XML Schema "deltaSynchForCMData.xsd" [37](#__RefHeading___Toc335992004)

B.3.3.3 XML Schema "deltaSynchForAlarms.xsd" [38](#__RefHeading___Toc335992005)

Annex C (normative): SOAP Solution Set [39](#__RefHeading___Toc335992006)

C.1 Architectural features [39](#__RefHeading___Toc335992007)

C.1.1 Syntax for Distinguished Names [39](#__RefHeading___Toc335992008)

C.1.2 Notification Services [39](#__RefHeading___Toc335992009)

C.1.3 Supported W3C specifications [39](#__RefHeading___Toc335992010)

C.1.4 Prefixes and namespaces [39](#__RefHeading___Toc335992011)

C.2 Mapping [40](#__RefHeading___Toc335992012)

C.2.1 Operation and notification mapping [40](#__RefHeading___Toc335992013)

C.2.2 Operation parameter mapping [40](#__RefHeading___Toc335992014)

C.2.3 Notification parameter mapping [41](#__RefHeading___Toc335992015)

C.3 Solution Set definitions [41](#__RefHeading___Toc335992016)

C.3.1 WSDL definition structure [41](#__RefHeading___Toc335992017)

C.3.2 Graphical Representation [41](#__RefHeading___Toc335992018)

C.3.3 WSDL specification “DeltaSynchIRPSystem.wsdl” [42](#__RefHeading___Toc335992019)

Annex D (informative): Change history [49](#__RefHeading___Toc335992020)

# Foreword

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

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

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

32.391: "Delta Synchronization Integration Reference Point (IRP); Requirements"

32.392: "Delta Synchronization Integration Reference Point (IRP): Information Service (IS)"

**32.396: "Delta Synchronization Integration Reference Point (IRP): Solution Set definitions"**

The Itf-N interface is built up by a number of 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 [2] and 3GPP TS 32.102 [3].

IRPManagers (typically Network Management Systems) and IRPAgents (typically EMs or NEs) synchronize their data concerning alarms or configuration data. In certain scenarios this synchronization is lost or not done. This IRP provides functionality to significantly reduces the amount of data which needs to be transferred in order to re-establish synchronization.

# 1 Scope

The present document specifies the Solution Set definitions for the IRP whose semantics is specified in Delta Synchronization IRP IS (3GPP TS 32.392 [5]).

This Solution Set specification is related to 3GPP TS 32.392 V14.0.X [5].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

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

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

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

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

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

[5] 3GPP TS 32.392: "Telecommunication management; Delta Synchronization Integration Reference Point (IRP): Information Service (IS)".

[6] 3GPP TS 32.391: "Configuration Management (CM); Delta SynchronizationIntegration Reference Point (IRP): Requirements".

[7] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[8] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[9] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[10] 3GPP TS 32.602: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP) management; Information Service (IS)".

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

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

[13] 3GPP TS 32.342: "Telecommunication management; File Transfer (FT) Integration Reference Point (IRP): Information Service (IS)".

[14] W3C REC-xml-20001006: "Extensible Markup Language (XML) 1.0 (Second Edition)".

[15] W3C REC-xmlschema-0-20010502: "XML Schema Part 0: Primer".

[16] W3C REC-xmlschema-1-20010502: "XML Schema Part 1: Structures".

[17] W3C REC-xmlschema-2-20010502: "XML Schema Part 2: Datatypes".

[18] W3C REC-xml-names-19990114: "Namespaces in XML"

[19] 3GPP TS 32.111-6: "Telecommunication management; Fault Management; Part 5: Alarm Integration Reference Point (IRP): Solution Set definitions ".

[20] 3GPP TS 32.616: "Telecommunication management; Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Solution Set definitions ".

[21] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

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

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

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

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

[26] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP): Requirements".

# 3 Definitions and abbreviations

## 3.1 Definitions

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

**IRP document version number string (or "IRPVersion"):** see 3GPP TS 32.311 [26].

**IRP:** See 3GPP TS 32.101 [2].

**IRPAgent:** See 3GPP TS 32.102 [3].

**IRPManager:** See 3GPP TS 32.102 [3].

**Changed alarm**: See 3GPP TS 32.391 [6].

**Changed instance**: See 3GPP TS 32.391 [6].

**Delta Synchronisation**: See 3GPP TS 32.391 [6].

**Delta Synchronisation Point**: See 3GPP TS 32.391 [6].

**Full Synchronisation**: See 3GPP TS 32.391 [6].

**XML file:** file containing an XML document

**XML document:** composed of the succession of an optional XML declaration followed by a root XML element [14].

**XML declaration:** it specifies the version of XML being used, see [14].

**XML element:** has a type, is identified by a name, may have a set of XML attribute specifications and is either composed of the succession of an XML start-tag followed by the XML content of the XML element followed by an XML end-tag, or composed simply of an XML empty-element tag; each XML element may contain other XML elements, see [14].

**empty XML element:** having an empty XML content; an empty XML element still possibly has a set of XML attribute specifications; an empty XML element is either composed of the succession of an XML start-tag directly followed by an XML end-tag, or composed simply of an XML empty-element tag, see [14].

**XML content (of an XML element):** empty if the XML element is simply composed of an XML empty-element tag; otherwise the part, possibly empty, of the XML element between its XML start-tag and its XML end-tag, see [14].

**XML start-tag:** the beginning of a non-empty XML element is marked by an XML start-tag containing the name and the set of XML attribute specifications of the XML element, see [14].

**XML end-tag:** the end of a non-empty XML element is marked by an XML end-tag containing the name of the XML element, see [14].

**XML empty-element tag:** composed simply of an empty-element tag containing the name and the set of XML attribute specifications of the XML element, see [14].

**XML attribute specification:** has a name and a value, see [14].

**DTD:** defines structure and content constraints to be respected by an XML document to be valid with regard to this DTD, see [14].

**XML schema:** more powerful than a DTD, an XML schema defines structure and content constraints to be respected by an XML document to conform with this XML schema; through the use of XML namespaces several XML schemas can be used together by a single XML document; an XML schema is itself also an XML document that shall conform with the XML schema for XML schemas, see [15], [16] and [17].

**XML namespace:** enables qualifying element and attribute names used in XML documents by associating them with namespaces identified by different XML schemas, see [15], [16] and [17].

**XML complex type:** defined in an XML schema; cannot be directly used in an XML document; can be the concrete type or the derivation base type for an XML element type or for another XML complex type; ultimately defines constraints for an XML element on its XML attribute specifications and/or its XML content, see [15], [16] and [17].

**XML element type:** declared by an XML schema; can be directly used in an XML document; as the concrete type of an XML element, directly or indirectly defines constraints on its XML attribute specifications and/or its XML content; can also be the concrete type or the derivation base type for another XML element type, see [15], [16] and [17].

## 3.2 Abbreviations

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

EM Element Manager

NM Network Management

IRP Integration Reference Point

Itf-N Interface N

NE Network Element

SS Solution Set

IS Information Service

XML eXtensible Mark-up Language

# 4 Solution Set definitions

This specification defines the following 3GPP Delta Synchronization IRP Solution Set definitions:

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

Annex A (normative):  
CORBA Solution Set

This annex contains the CORBA Solution Set for the IRP whose semantics is specified in Delta Synchronization IRP: Information Service (3GPP TS 32.392 [5]).

# A.1 Architectural features

The overall architectural feature of Delta Synchronization IRP is specified in 3GPP TS 32.392 [5].

## A.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [21].

## A.1.2 Notification Services

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

A necessary and sufficient sub set of OMG Notification Services shall be used to support DeltaSynchronizationNotifications notifications as specified in 3GPP TS 32.392 [5].

## A.1.3 Push and Pull Style

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

This CORBA SS specifies that support of Push style is Mandatory (M) and that support of Pull style is Optional (O).

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

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

## A.1.5 Delta Synchronization Notification Interface

OMG CORBA Notification push operation is used to realise the notification of Delta Synchronization IRP Notifications. All the notifications in this interface are implemented using this push\_structured\_event method.

### A.1.5.1 Method push (M)

module CosNotifyComm {

…

Interface SequencePushConsumer : NotifyPublish {

void push\_structured\_events(

in CosNotification::EventBatch notifications)

raises( CosEventComm::Disconnected);

…

}; // SequencePushConsumer

…

}; // CosNotifyComm

NOTE 1: The push\_structured\_events method takes an input parameter of type EventBatch as defined in the OMG CosNotification module (OMG Notification Service [10]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

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

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

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

# A.2 Mapping

## A.2.1 General mapping

Not applicable.

## A.2.2 Operation and notification mapping

The Delta Synchronization IS defines semantics of operations visible across the Itf-N. Table A.2.2-1 indicates mapping of these operations and notifications to their equivalents defined in this SS.

Table A.2.2-1: Mapping from IS Operation to SS equivalents

| IS Operation / Notification  (3GPP TS 32.392) | SS Method | Qualifier |
| --- | --- | --- |
| manageDeltaSynchronization | manageDeltaSynchronization | M |
| getAvailableDeltaSynchPoints | getAvailableDeltaSynchPoints | O |
| triggerDeltaSynchOfCMData | triggerDeltaSynchOfCMData | O |
| triggerDeltaSynchOfAlarms | triggerDeltaSynchOfAlarms | O |
| notifyStatusOfDeltaSynchronization | notifyStatusOfDeltaSynchronization | M |
| notifyNewDeltaSynchPoint | notifyNewDeltaSynchPoint | O |

## A.2.3 Operation parameter mapping

The Delta Synchronization IS defines semantics of parameters carried in operations across the Itf-N. The following tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table A.2.3-1: Mapping from IS manageDeltaSynchronization parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | DeltaSynchronizationConstDefs::ManagerReference | M |
| manageDeltaSynchForAlarmData | DeltaSynchronizationConstDefs::ManageDeltaSynchForXDataConditional | CM |
| manageDeltaSynchForCMData | DeltaSynchronizationConstDefs::ManageDeltaSynchForXDataConditional | CM |
| status | Exceptions:  DeltaSynchronizationConstDefs::ManageDeltaSynchronization, GenericIRPManagementSystem::ParameterNotSupported,  GenericIRPManagementSystem::InvalidParameter,  GenericIRPManagementSystem::ValueNotSupported,  GenericIRPManagementSystem::OperationNotSupported | M |

Table A.2.3-2: Mapping from IS getAvailableDeltaSynchPoints parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | DeltaSynchronizationConstDefs::ManagerReferenceOpt | O |
| synchPointsForCMDataRequested | DeltaSynchronizationConstDefs::SynchPointsRequestedConditional | CM |
| synchPointsForAlarmDataRequested | DeltaSynchronizationConstDefs::SynchPointsRequestedConditional | CM |
| synchPointListForAlarms | DeltaSynchronizationConstDefs::SynchPointListConditional | CM |
| synchPointListForCMData | DeltaSynchronizationConstDefs::SynchPointListConditional | CM |
| status | Exceptions:  DeltaSynchronizationConstDefs::DeltaSynchNotSupportedForCMData,  DeltaSynchronizationConstDefs::DeltaSynchNotSupportedForAlarmData,  DeltaSynchronizationConstDefs::DeltaSynchNotActive,  DeltaSynchronizationConstDefs::DeltaSynchForCMDataDeactivated,  DeltaSynchronizationConstDefs::DeltaSynchForAlarmDataDeactivated,  GenericIRPManagementSystem::ParameterNotSupported,  GenericIRPManagementSystem::InvalidParameter,  GenericIRPManagementSystem::ValueNotSupported,  GenericIRPManagementSystem::OperationNotSupported | M |

Table A.2.3-3: Mapping from IS triggerDeltaSynchOfCMData parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | DeltaSynchronizationConstDefs::ManagerReferenceOpt | O |
| dataRequested | DeltaSynchronizationConstDefs::CMDataRequestedOpt | O |
| baseMOInstance | KernelCmConstDefs::DNOpt | O |
| scope | KernelCmConstDefs::ScopeTypeOpt | O |
| synchPoint | DeltaSynchronizationConstDefs::SynchPoint | M |
| deltaLists | DeltaSynchronizationConstDefs::DeltaListsConditional | CM |
| newSynchPoint | DeltaSynchronizationConstDefs::SynchPointConditional | CM |
| status | Exceptions:  DeltaSynchronizationConstDefs::TriggerDeltaSynchOfCMData,  DeltaSynchronizationConstDefs::SynchronizationPointTooLongAgo,  DeltaSynchronizationConstDefs::TooManyChangesFullSynchronizationRecommended  DeltaSynchronizationConstDefs::SynchPointUnknown,  DeltaSynchronizationConstDefs::DeltaSynchNotSupportedForCMData,  DeltaSynchronizationConstDefs::DeltaSynchForCMDataDeactivated,  DeltaSynchronizationConstDefs::DeltaSynchNotActive,  GenericIRPManagementSystem::ParameterNotSupported,  GenericIRPManagementSystem::InvalidParameter,  GenericIRPManagementSystem::ValueNotSupported,  GenericIRPManagementSystem::OperationNotSupported | M |

Table A.2.3-4: Mapping from IS triggerDeltaSynchOfAlarms parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | DeltaSynchronizationConstDefs::ManagerReference | M |
| dataRequested | DeltaSynchronizationConstDefs::AlarmDataRequested | M |
| baseMOInstance | KernelCmConstDefs::DN | O |
| scope | KernelCmConstDefs::ScopeTypeOpt | O |
| synchPoint | DeltaSynchronizationConstDefs::SynchPoint | M |
| deltaLists | DeltaSynchronizationConstDefs::DeltaListsConditional | CM |
| newSynchPoint | DeltaSynchronizationConstDefs::SynchPointConditional | CM |
| status | Exceptions:  DeltaSynchronizationConstDefs::TriggerDeltaSynchOfAlarms,  DeltaSynchronizationConstDefs::SynchronizationPointTooLongAgo,  DeltaSynchronizationConstDefs::TooManyChangesFullSynchronizationRecommended  DeltaSynchronizationConstDefs::SynchPointUnknown,  DeltaSynchronizationConstDefs::DeltaSynchNotSupportedForAlarms,  DeltaSynchronizationConstDefs::DeltaSynchForAlarmsNotActive,  DeltaSynchronizationConstDefs::DeltaSynchNotActive,  GenericIRPManagementSystem::ParameterNotSupported,  GenericIRPManagementSystem::InvalidParameter,  GenericIRPManagementSystem::ValueNotSupported,  GenericIRPManagementSystem::OperationNotSupported | M |

## A.2.4 Notification parameter mapping

The delta synchronization Information Service defines semantics of parameters carried in notifications. The following table indicates the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [12]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [12], is:

Header

Fixed Header

domain\_name

type\_name

event\_name

Variable Header

Body

filterable\_body\_fields

remaining\_body

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the delta synchronization Information Service defined notification parameters.

Table A.2.4-1: Mapping for notifyStatusOfDeltaSynchronization

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name | M | It carries the IRP document version number string. See clause 3.1.  It indicates the syntax and semantics of the Structured Event as defined by the present document. |
| notificationType | type\_name | M | This is the NotifyDeltaSynchOfInstancesDeactivated of module DeltaSynchronizationNotifications. |
| There is no corresponding IS attribute. | event\_name | M | It carries no information. |
| There is no corresponding IS attribute. | Variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string.  Name of 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 [11] ). |
| notificationId | One NV pair of remaining\_body | M | 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 [11] ). |
| eventTime | One NV pair of filterable\_body\_fields | M | 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 [11]). |
| systemDN | One NV pair of filterable\_body\_fields | M | 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 (3GPP TS 32.306 [11]). |
| managerReference | One NV pair of remaining\_body | M | Name of NV pair is the MANAGER\_REFERENCE of interface notifyDeltaSynchOfInstancesDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is ManagerReference of module DeltaSynchronizationConstDefs. |
| deltaSynchStatusForCMData | One NV pair of remaining\_body | M | Name of NV pair is the DELTA\_SYNCH\_STATUS\_FOR\_CMDATA of interface notifyDeltaSynchOfInstancesDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is DeltaSynchStatus of module DeltaSynchronizationConstDefs. |
| deltaSynchStatusForAlarmData | One NV pair of remaining\_body | M | Name of NV pair is the DELTA\_SYNCH\_STATUS\_FOR\_ALARM\_DATA of interface notifyDeltaSynchOfInstancesDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is DeltaSynchStatus of module DeltaSynchronizationConstDefs. |

Table A.2.4-2: Mapping for notifyNewDeltaSynchPoint

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name | M | It carries the IRP document version number string. See subclause 3.1  It indicates the syntax and semantics of the Structured Event as defined by the present document. |
| notificationType | type\_name | M | This is the NotifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications. |
| There is no corresponding IS attribute. | event\_name | M | It carries no information. |
| There is no corresponding IS attribute. | Variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string.  Name of 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 [11]). |
| notificationId | One NV pair of remaining\_body | M | 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 [11]). |
| eventTime | One NV pair of filterable\_body\_fields | M | 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 [11]). |
| systemDN | One NV pair of filterable\_body\_fields | M | 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 (3GPP TS 32.306 [11]). |
| newSynchPoint | One NV pair of remaining\_body | M | Name of NV pair is the NEW\_DELTA\_SYNCH\_POINT of interface notifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is SynchPoint of module DeltaSynchronizationConstDefs. |
| requestedSynchPoint | One NV pair of remaining\_body | M | Name of NV pair is the REQUESTED\_SYNCH\_POINT of interface notifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is SynchPoint of module DeltaSynchronizationConstDefs. |
| deltaSynchPointType | One NV pair of remaining\_body | M | Name of NV pair is the DELTA\_SYNCH\_POINT\_TYPE of interface notifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is DeltaSynchPointType of module DeltaSynchronizationConstDefs. |
| triggeredByAgentOrManager | One NV pair of remaining\_body | M | Name of NV pair is the TRIGGERED\_BY\_AGENT\_OR\_MANAGER of interface notifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is TriggeredByAgentOrManager of module DeltaSynchronizationConstDefs. |
| agentOrManagerReference | One NV pair of remaining\_body | M | Name of NV pair is the AGENT\_OR\_MANAGER\_REFERENCE of interface notifyDeltaSynchOfAlarmsDeactivated of module DeltaSynchronizationNotifications.  Value of NV pair is AgentOrManagerReference of module DeltaSynchronizationConstDefs. |

# A.3 Solution Set definitions

## A.3.1 IDL definition structure

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

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

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

## A.3.2 IDL specification (file name "DeltaSynchronizationConstDefs.idl")

// File: DeltaSynchronizationConstDefs.idl

#ifndef \_DELTA\_SYNCHRONIZATION\_CONST\_DEFS\_IDL\_

#define \_DELTA\_SYNCHRONIZATION\_CONST\_DEFS\_IDL\_

#include <TimeBase.idl>

#include <DeltaSynchronizationConstDefs.idl>

#include <GenericIRPManagementConstDefs.idl>

#include <KernelCmConstDefs.idl>

#include <FileTransferIRPConstDefs.idl>

#include <AlarmIRPConstDefs.idl>

//FileTransferIRPConstDefs::FileLocation value;

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: DeltaSynchronizationConstDefs \*/

module DeltaSynchronizationConstDefs

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* definition of types used in several operations for Delta Synchronization: \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in several operations: \*/

typedef string ManagerReference;

typedef string AgentOrManagerReference;

/\*

ManagerReferenceConditional is a type carrying an optional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case the value is present. Otherwise the value is be absent.

\*/

union ManagerReferenceConditional switch (boolean)

{

case TRUE: ManagerReference value;

};

/\*

ManagerReferenceOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value is present. Otherwise the value is absent.

\*/

union ManagerReferenceOpt switch (boolean)

{

case TRUE: ManagerReference value;

};

enum ManageDeltaSynchForXData {ACTIVATE, DEACTIVATE};

/\*

ManageDeltaSynchForXDataConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case the value is present. Otherwise the value is absent.

\*/

union ManageDeltaSynchForXDataConditional switch (boolean)

{

case TRUE: ManageDeltaSynchForXData value;

};

typedef TimeBase::UtcT SynchPoint;

/\*

SynchPointConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case the value is present. Otherwise the value is absent.

\*/

union SynchPointConditional switch (boolean)

{

case TRUE: SynchPoint value;

};

/\*

SynchPointOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value is present. Otherwise the value is absent.

\*/

union SynchPointOpt switch (boolean)

{

case TRUE: SynchPoint value;

};

/\*

ScopeTypeOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value is present. Otherwise the value is absent.

\*/

union ScopeTypeOpt switch (boolean)

{

case TRUE: KernelCmConstDefs::ScopePara value;

};

/\*

BaseMOInstanceOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value is present. Otherwise the value is absent.

\*/

union BaseMOInstanceOpt switch (boolean)

{

case TRUE: GenericIRPManagementConstDefs::DN value;

};

enum Status {SUCCESS, FAILURE};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in operation manageDeltaSynchronization \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

enum ActivatedStatus {ACTIVATED, DEACTIVATED };

typedef ActivatedStatus ManageDeltaSynchMode;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in operation getAvailableDeltaSynchPoints \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

typedef boolean SynchPointsRequested;

/\*

SynchPointsRequestedConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. Otherwise the value may be absent.

\*/

union SynchPointsRequestedConditional switch (boolean)

{

case TRUE: SynchPointsRequested value;

};

typedef sequence <SynchPoint> SynchPointList;

/\*

SynchPointListConditional is a type carrying an optional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case the value is present. Otherwise the value is be absent.

\*/

union SynchPointListConditional switch (boolean)

{

case TRUE: SynchPointList value;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in operation triggerDeltaSynchOfCMData and \*/

/\* in operation triggerDeltaSynchOfAlarmData \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

AttributeListConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the operation’s dnsOnly=FALSE. In this case the value is present.

Otherwise the value is absent.

\*/

union AttributeListConditional switch (boolean)

{

case TRUE: GenericIRPManagementConstDefs::MOAttributeSet value;

};

struct ListedInstance

{

GenericIRPManagementConstDefs::DN moInstance; /\* DN is a string; \*/

AttributeListConditional attributeList;

};

typedef sequence <ListedInstance> ListOfInstances;

struct DeltaListsWithRealLists

{

TimeBase::UtcT startTime;

TimeBase::UtcT endTime;

ListOfInstances listOfCreatedInstances;

ListOfInstances listOfChangedInstances;

ListOfInstances listOfDeletedInstances;

};

struct AlarmDeltaListsWithRealLists

{

TimeBase::UtcT startTime;

TimeBase::UtcT endTime;

AlarmIRPConstDefs::AlarmInformationSeq listOfNewAlarms;

AlarmIRPConstDefs::AlarmInformationSeq listOfChangedAlarms;

AlarmIRPConstDefs::AlarmInformationIdSeq listOfDeletedAlarms;

};

typedef sequence <FileTransferIRPConstDefs::FileLocation> FileLocationList;

struct DeltaListsWithFileReferences

{

TimeBase::UtcT startTime;

TimeBase::UtcT endTime;

FileLocationList fileList;

/\* if several files are used, then they shall be processed by the IRPmanager in sequence, i.e. first file first, second file as second, … \*/

};

enum DeltaListContentChoice {REAL\_LISTS, FILE\_REFERENCES};

//The CmDeltaList may contain a list of ListOfInstances or a list of filenames

union CmDeltaLists switch (DeltaListContentChoice)

{

case REAL\_LISTS: DeltaListsWithRealLists deltaListRealLists;

case FILE\_REFERENCES: DeltaListsWithFileReferences deltaListFileReferences;

};

//The AlarmDeltaLists may contain a list of ListOfInstances or a list of filenames

union AlarmDeltaLists switch (DeltaListContentChoice)

{

case REAL\_LISTS: AlarmDeltaListsWithRealLists deltaListRealLists;

case FILE\_REFERENCES: DeltaListsWithFileReferences deltaListFileReferences;

};

/\*

CmDeltaListsConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case

the value is present. Otherwise the value is be absent.

\*/

union CmDeltaListsConditional switch (boolean)

{

case TRUE: CmDeltaLists value;

};

/\*

AlarmDeltaListsConditional is a type carrying a conditional parameter.

The boolean shall be TRUE, if the condition described in TS 32.392 is fulfilled. In this case

the value is present. Otherwise the value is absent.

\*/

union AlarmDeltaListsConditional switch (boolean)

{

case TRUE: AlarmDeltaLists value;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in operation triggerDeltaSynchOfAlarmData \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

enum AlarmDataRequested { ALARM\_IDS\_ONLY, COMPLETE\_ALARM\_INFORMATION };

/\*

AlarmDataRequestedOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value

is present. Otherwise the value is absent.

\*/

union AlarmDataRequestedOpt switch (boolean)

{

case TRUE: AlarmDataRequested value;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* types used in operation triggerDeltaSynchOfCMData \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

enum CMDataRequested { DNS\_ONLY, COMPLETE\_DATA\_SET };

/\*

CMDataRequestedOpt is a type carrying an optional parameter.

The boolean shall be TRUE, if the operation request uses this parameter. In this case the value

is present. Otherwise the value is absent.

\*/

union CMDataRequestedOpt switch (boolean)

{

case TRUE: CMDataRequested value;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* definition of types in notifications for Delta Synchronization \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

enum DeltaSynchPointType { DELTA\_SYNCH\_POINT\_FOR\_ALARM, DELTA\_SYNCH\_POINT\_FOR\_CM\_DATA };

typedef ActivatedStatus DeltaSynchStatus;

enum TriggeredBy { IRP\_AGENT, IRP\_MANAGER };

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Definition of parameters specified in notifications for Delta Synchronization \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

interface AttributeNameValue

{

const string MANAGER\_REFERENCE = "MANAGER\_REFERENCE";

const string AGENT\_OR\_MANAGER\_REFERENCE = "AGENT\_OR\_MANAGER\_REFERENCE";

const string DELTA\_SYNCH\_STATUS\_FOR\_CMDATA = "DELTA\_SYNCH\_STATUS\_FOR\_CMDATA";

const string DELTA\_SYNCH\_STATUS\_FOR\_ALARM\_DATA = "DELTA\_SYNCH\_STATUS\_FOR\_ALARM\_DATA";

const string NEW\_DELTA\_SYNCH\_POINT = "NEW\_DELTA\_SYNCH\_POINT";

const string DELTA\_SYNCH\_POINT\_TYPE = "DELTA\_SYNCH\_POINT\_TYPE";

const string REQUESTED\_SYNCH\_POINT = "REQUESTED\_SYNCH\_POINT";

const string TRIGGERED\_BY\_AGENT\_OR\_MANAGER = "TRIGGERED\_BY\_AGENT\_OR\_MANAGER";

};

};

#endif // \_DELTA\_SYNCHRONIZATION\_CONST\_DEFS\_IDL\_

## A.3.3 IDL specification (file name "DeltaSynchronizationSystem.idl")

//File: DeltaSynchronizationSystem.idl

#ifndef \_DELTA\_SYNCHRONIZATION\_SYSTEM\_IDL\_

#define \_DELTA\_SYNCHRONIZATION\_SYSTEM\_IDL\_

#include <KernelCmConstDefs.idl>

#include <DeltaSynchronizationConstDefs.idl>

#include <GenericIRPManagementSystem.idl>

#include <AlarmIRPConstDefs.idl>

#include <AlarmIRPSystem.idl>

#include <NotificationLogIRPSystem.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: DeltaSynchronizationSystem \*/

module DeltaSynchronizationSystem

{

/\*

If the system fails to complete an operation, then it can provide a reason

to qualify the exception. The semantics carried in this reason are outside

the scope of the present document.

\*/

exception ManageDeltaSynchronization { string reason; };

exception GetAvailableDeltaSynchPoints { string reason; };

exception TriggerDeltaSynchOfCMData { string reason; };

exception TriggerDeltaSynchOfAlarms { string reason; };

exception SynchronizationPointTooLongAgo { string reason; };

exception TooManyChangesFullSynchronizationRecommended { string reason; };

exception DeltaSynchNotSupportedForCMData { string reason; };

exception DeltaSynchNotSupportedForAlarmData { string reason; };

exception DeltaSynchNotActive { string reason; };

exception DeltaSynchForCMDataDeactivated { string reason; };

exception DeltaSynchForAlarmDataDeactivated { string reason; };

exception SynchPointTooLongAgo{ string reason; };

exception SynchPointUnknown { string reason; };

exception DeltaSynchNotSupportedForAlarms { string reason; };

exception DeltaSynchForAlarmsNotActive { string reason; };

interface DeltaSynchGenericParts

{

DeltaSynchronizationConstDefs::Status manageDeltaSynchronization

/\* for the purpose of this operation see 3GPP TS 32.392 \*/

(

in DeltaSynchronizationConstDefs::ManagerReference managerReference,

in DeltaSynchronizationConstDefs::ManageDeltaSynchForXDataConditional

manageDeltaSynchForAlarmData,

in DeltaSynchronizationConstDefs::ManageDeltaSynchForXDataConditional

manageDeltaSynchForCMData

)

raises

(

ManageDeltaSynchronization,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::ValueNotSupported,

GenericIRPManagementSystem::OperationNotSupported

);

DeltaSynchronizationConstDefs::Status getAvailableDeltaSynchPoints

/\* for the purpose of this operation see 3GPP TS 32.392 \*/

(

in DeltaSynchronizationConstDefs::ManagerReferenceOpt managerReference,

in DeltaSynchronizationConstDefs::SynchPointsRequestedConditional

synchPointsForCMDataRequested,

in DeltaSynchronizationConstDefs::SynchPointsRequestedConditional

synchPointsForAlarmDataRequested,

out DeltaSynchronizationConstDefs::SynchPointListConditional synchPointListForAlarms,

out DeltaSynchronizationConstDefs::SynchPointListConditional synchPointListForCMData

)

raises

(

GetAvailableDeltaSynchPoints,

DeltaSynchNotSupportedForCMData,

DeltaSynchNotSupportedForAlarmData,

DeltaSynchNotActive,

DeltaSynchForCMDataDeactivated,

DeltaSynchForAlarmDataDeactivated,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::ValueNotSupported,

GenericIRPManagementSystem::OperationNotSupported

);

};

interface DeltaSynchOfCMData

{

DeltaSynchronizationConstDefs::Status triggerDeltaSynchOfCMData

/\* for the purpose of this operation see 3GPP TS 32.392 \*/

(

in DeltaSynchronizationConstDefs::ManagerReferenceOpt managerReference,

in DeltaSynchronizationConstDefs::CMDataRequestedOpt cmDataRequested,

in DeltaSynchronizationConstDefs::BaseMOInstanceOpt baseMOInstance,

in DeltaSynchronizationConstDefs::ScopeTypeOpt scope,

in DeltaSynchronizationConstDefs::SynchPoint synchPoint,

out DeltaSynchronizationConstDefs::CmDeltaListsConditional deltaLists,

out DeltaSynchronizationConstDefs::SynchPointConditional newSynchPoint

)

raises

(

TriggerDeltaSynchOfCMData,

SynchronizationPointTooLongAgo,

TooManyChangesFullSynchronizationRecommended,

SynchPointUnknown,

DeltaSynchNotSupportedForCMData,

DeltaSynchForCMDataDeactivated,

DeltaSynchNotActive,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::ValueNotSupported,

GenericIRPManagementSystem::OperationNotSupported

);

};

interface DeltaSynchOfAlarmData

{

DeltaSynchronizationConstDefs::Status triggerDeltaSynchOfAlarms

/\* for the purpose of this operation see 3GPP TS 32.392 \*/

(

in DeltaSynchronizationConstDefs::ManagerReferenceOpt managerReference,

in DeltaSynchronizationConstDefs::AlarmDataRequestedOpt alarmDataRequested,

in DeltaSynchronizationConstDefs::BaseMOInstanceOpt baseMOInstance,

in DeltaSynchronizationConstDefs::ScopeTypeOpt scope,

in DeltaSynchronizationConstDefs::SynchPoint synchPoint,

out DeltaSynchronizationConstDefs::AlarmDeltaListsConditional deltaLists,

out DeltaSynchronizationConstDefs::SynchPointConditional newSynchPoint

)

raises

(

TriggerDeltaSynchOfAlarms,

SynchronizationPointTooLongAgo,

TooManyChangesFullSynchronizationRecommended,

SynchPointUnknown,

DeltaSynchNotSupportedForAlarms,

DeltaSynchForAlarmsNotActive,

DeltaSynchNotActive,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::ValueNotSupported,

GenericIRPManagementSystem::OperationNotSupported

);

};

interface DeltaSynchIRPSystem : DeltaSynchGenericParts, DeltaSynchOfCMData,

DeltaSynchOfAlarmData, GenericIRPManagementSystem::GenericIRPManagement{};

};

#endif // \_DELTA\_SYNCHRONIZATION\_SYSTEM\_IDL\_

## A.3.4 IDL specification (file name "DeltaSynchronizationNotifications.idl")

//File: DeltaSynchronizationNotifications.idl

#ifndef \_DELTA\_SYNCHRONIZATION\_NOTIFICATIONS\_IDL\_

#define \_DELTA\_SYNCHRONIZATION\_NOTIFICATIONS\_IDL\_

#include <DeltaSynchronizationConstDefs.idl>

#include <NotificationIRPNotifications.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: DeltaSynchronizationNotifications

This contains the specification of notifications of Delta Synchronization.

==========================================================================

\*/

module DeltaSynchronizationNotifications

{

/\* Constant definitions for the NotifyDeltaSynchOfInstancesDeactivated notification \*/

interface notifyDeltaSynchOfInstancesDeactivated: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyStatusOfDeltaSynchronization";

/\*\*

\* This constant defines the name of the ManagerReference property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::ManagerReferenceConditional.

\*/

const string MANAGER\_REFERENCE =

DeltaSynchronizationConstDefs::AttributeNameValue::MANAGER\_REFERENCE;

/\*\*

\* This constant defines the name of the DeltaSynchStatusForCMData property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::DeltaSynchStatus.

\*/

const string DELTA\_SYNCH\_STATUS\_FOR\_CMDATA =

DeltaSynchronizationConstDefs::AttributeNameValue::DELTA\_SYNCH\_STATUS\_FOR\_CMDATA;

/\*\*

\* This constant defines the name of the DeltaSynchStatusForAlarmData property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::DeltaSynchStatus.

\*/

const string DELTA\_SYNCH\_STATUS\_FOR\_ALARM\_CMDATA =

DeltaSynchronizationConstDefs::AttributeNameValue::DELTA\_SYNCH\_STATUS\_FOR\_ALARM\_DATA;

};

/\* Constant definitions for the notifyNewDeltaSynchPoint notification \*/

interface notifyNewDeltaSynchPoint: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyNewDeltaSynchPoint";

/\*\*

\* This constant defines the name of the AgentOrManagerReference property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::AgentOrManagerReference.

\*/

const string AGENT\_OR\_MANAGER\_REFERENCE =

DeltaSynchronizationConstDefs::AttributeNameValue::AGENT\_OR\_MANAGER\_REFERENCE;

/\*\*

\* This constant defines the name of the NewDeltaSynchPoint property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::SynchPoint.

\*/

const string NEW\_DELTA\_SYNCH\_POINT =

DeltaSynchronizationConstDefs::AttributeNameValue::NEW\_DELTA\_SYNCH\_POINT;

/\*\*

\* This constant defines the name of the RequestedSynchPoint property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::SynchPoint.

\*/

const string REQUESTED\_SYNCH\_POINT =

DeltaSynchronizationConstDefs::AttributeNameValue::REQUESTED\_SYNCH\_POINT;

/\*\*

\* This constant defines the name of the DeltaSynchPointType property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs:: DeltaSynchPointType.

\*/

const string DELTA\_SYNCH\_POINT\_TYPE =

DeltaSynchronizationConstDefs::AttributeNameValue::DELTA\_SYNCH\_POINT\_TYPE;

/\*\*

\* This constant defines the name of the TriggeredByAgentOrManager property,

\* which is transported in the filterable\_body fields.

\* The data type for the value of this property is

\* DeltaSynchronizationConstDefs::TriggeredBy.

\*/

const string TRIGGERED\_BY\_AGENT\_OR\_MANAGER =

DeltaSynchronizationConstDefs::AttributeNameValue::TRIGGERED\_BY\_AGENT\_OR\_MANAGER;

};

};

#endif // \_DELTA\_SYNCHRONIZATION\_NOTIFICATIONS\_IDL\_

Annex B (normative):  
XML definitions

This annex contains the XML Definitions for the Delta SynchronisationIRP for the IRP whose semantics is specified in Delta SynchronisationIRP: Information Service (3GPP TS 32.392 [5]).

This XML Definitions specification defines the XML syntax of the Delta Synchronisation IRP XML Data File.

# B.1 Architectural Features

The overall architectural feature of Delta Synchronisation IRP is specified in 3G TS 32.392 [5]. This clause specifies features that are specific to the XML definitions.

## B.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [21].

## B.1.2 Notification Services

This annex defines the XML syntax of Delta Synchronisation IRP notifications that is to be used for the Delta Synchronisation IRP SOAP Solution Set and in conjunction with Notification Log IRP XML Definitions for Notification Log IRP XML Data File and the NL IRP XML Notification Format.

## B.1.3 IOC definitions

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

# B.2 Mapping

Not present in the current version of this specification.

# B.3 Solution Set definitions

## B.3.1 XML definition structure

The present document defines the delta synchronizatiron-specific XML schema and element types in deltaSynchForCMData.xsd / deltaSynchForAlarms.xsd which shall be used for the files of which the locations can be supplied as an option in output of operation triggerDeltaSynchOfCMData or triggerDeltaSynchOfAlarms defined in 3GPP TS 32.392 [5].

The present schema re-uses the structure and content of 3GPP TS 32.616 [20] for delta synchronization for CM data and those of 3GPP TS 32.111-6 [19] for delta synchronization for alarm data. The differences or additions are listed here:

For delta synchronization for CM data:  
The modifier XML attribute specification shall be used as follows:

- Create for all created NRM instances

- Update for all modified NRM instances

- Delete for all deleted NRM instances

For delta synchronization for alarm data:

- Created alarms shall be indicated by content XML elements of type="xa:NotifyNewAlarm”

- Changed alarms shall be indicated by content XML elements of type="xa:NotifyChangedAlarm” or of type="xa:NotifyClearedAlarm” or of type="xa:NotifyComments”

Deleted alarms shall be indicated by content XML elements of type="xa:NotifyAckStateChanged

### B.3.1.1 Global structure

See 3GPP TS 32.616 [20]

The following XML namespaces are potentially used in delta synchronisation XML files:

- delta synchronisation for CM data files are associated with XML schema deltaSynchForCMData.xsd;

- delta synchronisation for alarm data files are associated with XML schema deltaSynchForAlarmData.xsd;

### B.3.1.2 XML elements fileHeader and fileFooter

#### B.3.1.2.1 XML elements fileHeaderForDeltaSynchForCM/AlarmData

See 3GPP TS 32.616 [20]

The file header defined there is extended as follows (new parts marked by underlining):

For delta synchronization of CM data:

<element name="fileHeaderDeltaSynchForCMData">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0" maxOccurs="1">

<complexType>

<sequence>

<attribute name="managerReference" type="string" use="optional"/>

<attribute name="dataRequested" type="dsa:AlarmDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

</sequence>

</complexType>

</element>

</complexType>

</element>

<element name="fileHeaderDeltaSynchForCMData">

<complexType>

<sequence>

<element name="fileHeader">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

</complexType>

</element>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0" maxOccurs="1">

<complexType>

<attribute name="dataRequested" type="dsc:CMDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="managerReference" type="string" use="optional"/>

</complexType>

</element>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

</sequence>

</complexType>

</element>

For delta synchronization of alarm data:

<element name="fileHeaderDeltaSynchForCMData">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0" maxOccurs="1">

<complexType>

<sequence>

<attribute name="managerReference" type="string" use="optional"/>

<attribute name="dataRequested" type="dsa:AlarmDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

</sequence>

</complexType>

</element>

</complexType>

</element>

<element name="fileHeaderDeltaSynchForAlarms">

<complexType>

<sequence>

<element name="fileHeader">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

</complexType>

</element>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0" maxOccurs="1">

<complexType>

<attribute name="dataRequested" type="dsa:AlarmDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="managerReference" type="string" use="optional"/>

</complexType>

</element>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

</sequence>

</complexType>

</element>

#### B.3.1.2.2 XML element fileFooter

See 3GPP TS 32.616 [20]

### B.3.1.3 Delta synchronisation IRP specific XML elements

For delta synchronization of CM data: Currently none, except those in clause Annex B.3.1.2.

For delta synchronization of alarm data:

<element name="deltaAlarm" type=”dsa:DeltaAlarm maxOccurs="unbounded"/>

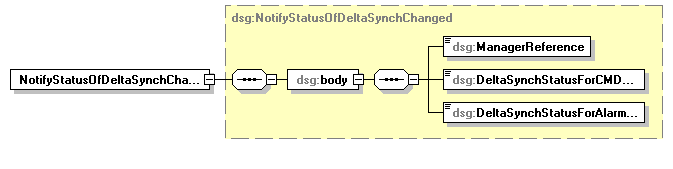
of type

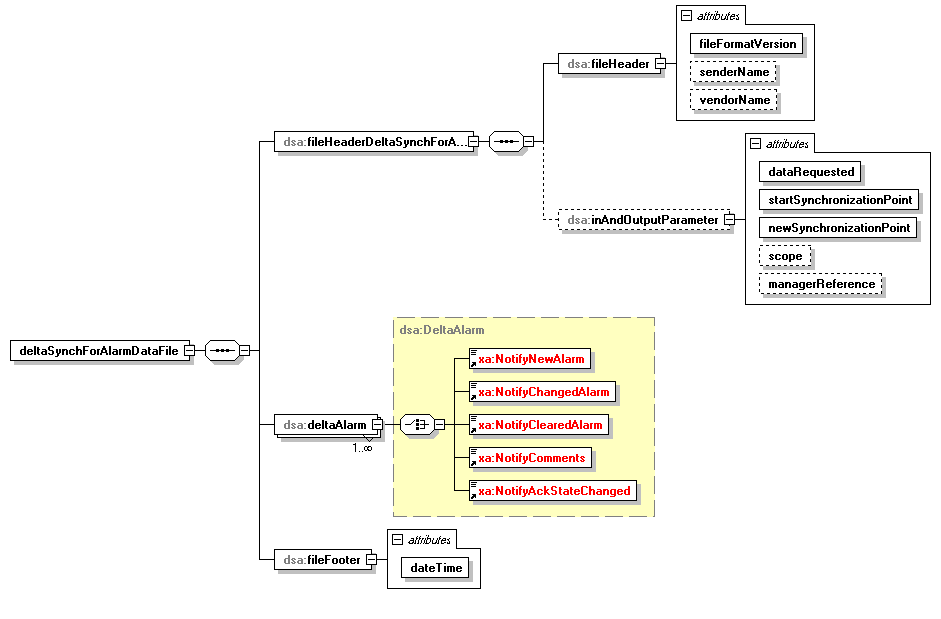
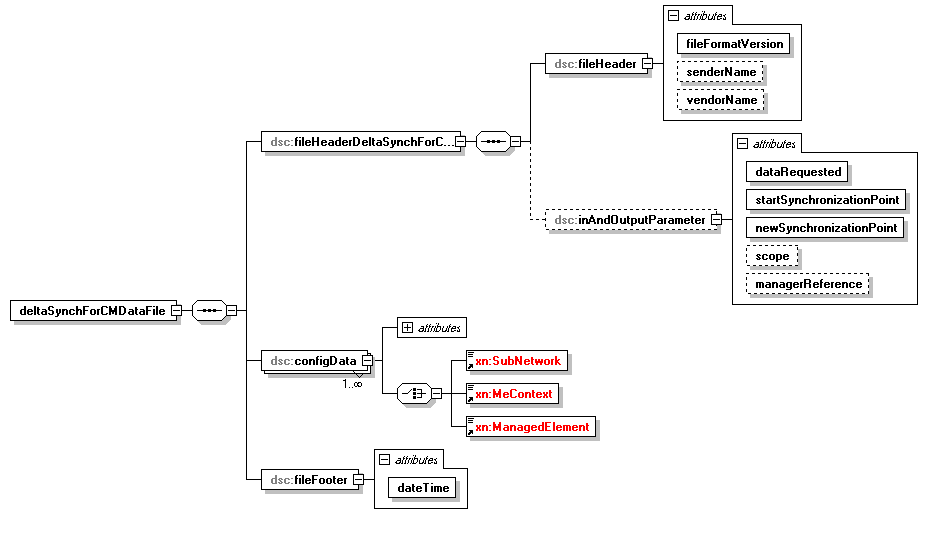
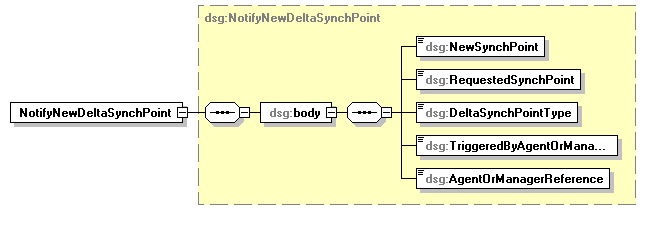
<complexType name=”DeltaAlarm”>  
 <choice>  
 <element ref="xa:NotifyNewAlarm"/>  
 <element ref="xa:NotifyChangedAlarm"/>  
 <element ref="xa:NotifyClearedAlarm"/>  
 <element ref="xa:NotifyComments"/>  
 <element ref="xa:NotifyAckStateChanged"/>  
 </choice>  
</complexType>

### B.3.1.4 Delta synchronisation IRP XML File Name Conventions

For NL IRP XML File Name Conventions the generic file name definitions as specified by the FT IRP apply (see [13])

## B.3.2 Graphical Representation





## B.3.3 XML Schema

### B.3.3.1 XML Schema "deltaSynchGeneric.xsd"

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.396 Delta Synchronisation IRP

Delta synchronization specific data file XML schemas

deltaSynchGeneric.xsd

-->

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dsg="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchGeneric" xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchGeneric" elementFormDefault="qualified">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification"/>

<!-- XML types specific for delta synchronization generic part -->

<simpleType name="DeltaSynchStatus">

<restriction base="string">

<enumeration value="Activated"/>

<enumeration value="Deactivated"/>

</restriction>

</simpleType>

<complexType name="NotifyStatusOfDeltaSynchChanged">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element name="ManagerReference" type="string"/>

<element name="DeltaSynchStatusForCMData" type="dsg:DeltaSynchStatus"/>

<element name="DeltaSynchStatusForAlarmData" type="dsg:DeltaSynchStatus"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<element name="NotifyStatusOfDeltaSynchChanged" type="dsg:NotifyStatusOfDeltaSynchChanged"/>

<simpleType name="DeltaSynchPointType">

<restriction base="string">

<enumeration value="DeltaSynchPointForAlarm"/>

<enumeration value="DeltaSynchPointForCMData"/>

</restriction>

</simpleType>

<simpleType name="TriggeredByAgentOrManager">

<restriction base="string">

<enumeration value="IRPAgent"/>

<enumeration value="IRPManager"/>

</restriction>

</simpleType>

<complexType name="NotifyNewDeltaSynchPoint">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element name="NewSynchPoint" type="dateTime"/>

<element name="RequestedSynchPoint" type="dateTime"/>

<element name="DeltaSynchPointType" type="dsg:DeltaSynchPointType"/>

<element name="TriggeredByAgentOrManager" type="dsg:TriggeredByAgentOrManager"/>

<element name="AgentOrManagerReference" type="string"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<element name="NotifyNewDeltaSynchPoint" type="dsg:NotifyNewDeltaSynchPoint"/>

</schema>

### B.3.3.2 XML Schema "deltaSynchForCMData.xsd"

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.396 Delta Synchronisation IRP

Delta synchronization specific data file XML schemas for CM data XML schema

deltaSynchForCMData.xsd

-->

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dsc="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchForCMData" xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm" xmlns:nk="http://www.3gpp.org/ftp/specs/archive/32\_series/32.666#kernelNtf" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchForCMData" elementFormDefault="qualified">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.666#kernelNtf"/>

<!-- XML types specific for delta synchronization of CM data -->

<simpleType name="CMDataRequested">

<restriction base="string">

<enumeration value="DNsOnly"/>

<enumeration value="CompleteDataSet"/>

</restriction>

</simpleType>

<!--Delta synchronization for CM data file root XML element -->

<element name="deltaSynchForCMDataFile">

<complexType>

<sequence>

<element name="fileHeaderDeltaSynchForCMData">

<complexType>

<sequence>

<element name="fileHeader">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

</complexType>

</element>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0">

<complexType>

<attribute name="dataRequested" type="dsc:CMDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="managerReference" type="string" use="optional"/>

</complexType>

</element>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

</sequence>

</complexType>

</element>

<element name="configData" maxOccurs="unbounded">

<complexType>

<choice>

<element ref="xn:SubNetwork"/>

<element ref="xn:MeContext"/>

<element ref="xn:ManagedElement"/>

</choice>

<attribute name="dnPrefix" type="string" use="optional"/>

</complexType>

</element>

<element name="fileFooter">

<complexType>

<attribute name="dateTime" type="dateTime" use="required"/>

</complexType>

</element>

</sequence>

</complexType>

</element>

</schema>

### B.3.3.3 XML Schema "deltaSynchForAlarms.xsd"

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.396 Delta Synchronisation IRP

Delta synchronization specific data file XML schemas for alarm data XML schema

deltaSynchForAlarmData.xsd

-->

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:dsa="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchForAlarms" xmlns:nk="http://www.3gpp.org/ftp/specs/archive/32\_series/32.666#kernelNtf" xmlns:xa="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPNotif" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#deltaSynchForAlarms" elementFormDefault="qualified">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPNotif"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.666#kernelNtf"/>

<!-- XML types specific for delta synchronization of alarm data -->

<simpleType name="AlarmDataRequested">

<restriction base="string">

<enumeration value="AlarmIdsOnly"/>

<enumeration value="CompleteAlarmInformation"/>

</restriction>

</simpleType>

<complexType name="DeltaAlarm">

<choice>

<element ref="xa:NotifyNewAlarm"/>

<element ref="xa:NotifyChangedAlarm"/>

<element ref="xa:NotifyClearedAlarm"/>

<element ref="xa:NotifyComments"/>

<element ref="xa:NotifyAckStateChanged"/>

</choice>

</complexType>

<!--Delta synchronization for alarm data file root XML element -->

<element name="deltaSynchForAlarmDataFile">

<complexType>

<sequence>

<element name="fileHeaderDeltaSynchForAlarms">

<complexType>

<sequence>

<element name="fileHeader">

<complexType>

<attribute name="fileFormatVersion" type="string" use="required"/>

<attribute name="senderName" type="string" use="optional"/>

<attribute name="vendorName" type="string" use="optional"/>

</complexType>

</element>

<!-- The content of the following element is a copy of the original

request/response parameters -->

<element name="inAndOutputParameter" minOccurs="0">

<complexType>

<attribute name="dataRequested" type="dsa:AlarmDataRequested" use="required"/>

<attribute name="startSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="newSynchronizationPoint" type="dateTime" use="required"/>

<attribute name="managerReference" type="string" use="optional"/>

</complexType>

</element>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

</sequence>

</complexType>

</element>

<element name="deltaAlarm" type="dsa:DeltaAlarm" maxOccurs="unbounded"/>

<element name="fileFooter">

<complexType>

<attribute name="dateTime" type="dateTime" use="required"/>

</complexType>

</element>

</sequence>

</complexType>

</element>

</schema>

Annex C (normative):  
SOAP Solution Set

This annex specifies the SOAP Solution Set for the IRP whose semantics are specified in Delta Synchronisation IRP: Information Service (3GPP TS 32.392 [5]).

# C.1 Architectural features

The overall architectural feature of Delta Synchronisation IRP is specified in 3GPP TS 32.392 [5]. This clause specifies features that are specific to the SOAP solution set..

## C.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300[21].

## C.1.2 Notification Services

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

## C.1.3 Supported W3C specifications

The SOAP 1.1 specification [22] and WSDL 1.1 specification [24] are supported.

The SOAP 1.2 specification [25] is supported optionally.

This specification uses "document" style in WSDL file.

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

The filter language used in the SS is the XPath Language (see W3C XPath 1.0 specification [23]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

Relevant definitions are imported from the Delta Synchronisation IRP XML definitions of Annex B

## C.1.4 Prefixes and namespaces

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

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

|  |  |
| --- | --- |
| **PREFIX** | **NAMESPACE** |
| (no prefix) | http://schemas.xmlsoap.org/wsdl/ |
| soap | http://schemas.xmlsoap.org/wsdl/soap/ |
| deltaSynchIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPSystem |
| deltaSynchIRPData | http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPData |
| xn | http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm |
| genericIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem |
| ntfIRPNtfSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification/NotificationIRPNtfSystem |

# C.2 Mapping

## C.2.1 Operation and notification mapping

The Delta Synchronisation IRP IS (3GPP TS 32.392 [5]) defines semantics of operation and notification visible across the Itf-N. Table C.2.1 indicates mapping of these operations and notifications to their equivalents defined in this SS.

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

| IS Operation / Notification (3GPP TS 32.392) | SS Method | SS Port | Qualifier |
| --- | --- | --- | --- |
| manageDeltaSynchronization | manageDeltaSynchronization | DeltaSynchGenericPartsPort | M |
| getAvailableDeltaSynchPoints | getAvailableDeltaSynchPoints | DeltaSynchGenericPartsPort | O |
| triggerDeltaSynchOfCMData | triggerDeltaSynchOfCMData | DeltaSynchOfCMDataPort | O |
| triggerDeltaSynchOfAlarms | triggerDeltaSynchOfAlarms | DeltaSynchOfAlarmDataPort | O |
| notifyStatusOfDeltaSynchronization | notify (note 1) | NotificationIRPNtfPort | M |
| notifyNewDeltaSynchPoint | notify (note 1) | NotificationIRPNtfPort | O |
| NOTE 1: The IS equivalent maps to an XML definition specified in Annex B, and this being an input parameter to the operation notify under the port type ntfIRPNtfSystem:NotificationIRPNtf and under the binding ntfIRPNtfSystem:NotificationIRPNtf of 3GPP TS 32.306 [11]. | | | |

## C.2.2 Operation parameter mapping

The Delta Synchronisation IRP IS (3GPP TS 32.392 [5]) defines semantics of parameters carried in the operations. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | managerReference | M |
| manageDeltaSynchForAlarmData | manageDeltaSynchForAlarmData | CM |
| manageDeltaSynchForCMData | manageDeltaSynchForCMData | CM |
| status | status | M |

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | managerReference | O |
| synchPointsForCMDataRequested | synchPointsForCMDataRequested | CM |
| synchPointsForAlarmDataRequested | synchPointsForAlarmDataRequested | CM |
| synchPointListForAlarms | synchPointListForAlarms | CM |
| synchPointListForCMData | synchPointListForCMData | CM |
| status | status | M |

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | managerReference | O |
| dataRequested | dataRequested | O |
| baseMOInstance | baseMOInstance | O |
| scope | scope | O |
| synchPoint | synchPoint | M |
| deltaLists | deltaLists | CM |
| newSynchPoint | newSynchPoint | CM |
| status | status | M |

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerReference | managerReference | M |
| dataRequested | dataRequested | M |
| baseMOInstance | baseMOInstance | O |
| scope | scope | O |
| synchPoint | synchPoint | M |
| deltaLists | deltaLists | CM |
| newSynchPoint | newSynchPoint | CM |
| status | status | M |

## C.2.3 Notification parameter mapping

The Delta Synchronisation IRP Notifications are defined in Annex B.

# C.3 Solution Set definitions

## C.3.1 WSDL definition structure

Clause C.3.2 provides a graphical representation of the Partial Suspension of Itf-N IRP service.

Clause C.3.3 defines the services which are supported the Partial Suspension of Itf-N IRP agent.

## C.3.2 Graphical Representation

The WSDL structure is depicted in Figure C.3.2 below, depicting port type, binding and service. The port type contains port type operations, which again contains input, output and fault messages. The binding contains binding operations, which have the same name as the port type operations. The binding connects to a port inside the service.



Figure C.3.2: Delta Synchronisation IRP SOAP Solution Set WSDL structure

## C.3.3 WSDL specification “DeltaSynchIRPSystem.wsdl”

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.396 Delta Synchronization IRP SOAP Solution Set

-->

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:deltaSynchIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPSystem"

xmlns:deltaSynchIRPData="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPData"

xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm"

xmlns:genericIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem"

xmlns:ntfIRPNtfSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem"

targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPSystem">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem"/>

<types>

<schema targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchIRPData" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:nk="http://www.3gpp.org/ftp/specs/archive/32\_series/32.666#kernelNtf" xmlns:xai="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs">

<!-- listOfInstances Type -->

<complexType name="listOfInstances">

<choice>

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="MOInstance" type="xn:dn" minOccurs="0"/>

<element name="attributeList" minOccurs="0">

<complexType>

<sequence maxOccurs="unbounded">

<element name="attributeName" type="string"/>

<element name="attributeValue" type="string" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="fileReference" type="xn:dn"/>

</sequence>

</choice>

</complexType>

<!-- listOfAlarms Type -->

<complexType name="listOfAlarms">

<choice>

<sequence>

<element name="alarmIdList">

<complexType>

<sequence minOccurs="0" maxOccurs="unbounded">

<element ref="xai:alarmId"/>

</sequence>

</complexType>

</element>

</sequence>

<sequence>

<element name="alarmInformationList">

<complexType>

<sequence minOccurs="0" maxOccurs="unbounded">

<choice>

<element name="nonSecurityAlarm" type="xai:NonSecurityAlarm"/>

<element name="securityAlarm" type="xai:SecurityAlarm"/>

</choice>

</sequence>

</complexType>

</element>

</sequence>

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="fileReference" type="xn:dn"/>

</sequence>

</choice>

</complexType>

<!-- manageDeltaSynchronization Request -->

<element name="manageDeltaSynchronizationRequest">

<complexType>

<sequence>

<element name="managerReference" type="string"/>

<element name="manageDeltaSynchForAlarmData" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="Activate"/>

<enumeration value="Deactivate"/>

</restriction>

</simpleType>

</element>

<element name="manageDeltaSynchForCMData" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="Activate"/>

<enumeration value="Deactivate"/>

</restriction>

</simpleType>

</element>

</sequence>

</complexType>

</element>

<!-- manageDeltaSynchronization Response -->

<element name="manageDeltaSynchronizationResponse">

<complexType>

<sequence>

<element name="status">

<simpleType>

<restriction base="string">

<enumeration value="Success"/>

<enumeration value="Failure"/>

</restriction>

</simpleType>

</element>

<element name="failureReason" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="DeltaSynchNotSupportedForCMData"/>

<enumeration value="DeltaSynchNotSupportedForAlarmData"/>

<enumeration value="operation\_failed"/>

<enumeration value="operation\_failed\_invalid\_input\_parameter"/>

<enumeration value="operation\_failed\_internal\_problem"/>

</restriction>

</simpleType>

</element>

</sequence>

</complexType>

</element>

<!-- manageDeltaSynchronization Fault -->

<element name="manageDeltaSynchronizationFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- getAvailableDeltaSynchPoints Request -->

<element name="getAvailableDeltaSynchPointsRequest">

<complexType>

<sequence>

<element name="managerReference" type="string" minOccurs="0"/>

<element name="synchPointsForCMDataRequested" minOccurs="0"/>

<element name="synchPointsForAlarmDataRequested" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- getAvailableDeltaSynchPoints Response -->

<element name="getAvailableDeltaSynchPointsResponse">

<complexType>

<sequence>

<element name="synchPointListForAlarms" minOccurs="0">

<complexType>

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="synchPoint" type="dateTime"/>

</sequence>

</complexType>

</element>

<element name="synchPointListForCMData" minOccurs="0">

<complexType>

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="synchPoint" type="dateTime"/>

</sequence>

</complexType>

</element>

<element name="status">

<simpleType>

<restriction base="string">

<enumeration value="Success"/>

<enumeration value="Failure"/>

</restriction>

</simpleType>

</element>

<element name="failureReason" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="DeltaSynchNotSupportedForCMData"/>

<enumeration value="DeltaSynchNotSupportedForAlarmData"/>

<enumeration value="DeltaSynchNotActive"/>

<enumeration value="DeltaSynchForCMDataDeactivated"/>

<enumeration value="DeltaSynchForAlarmDataDeactivated"/>

<enumeration value="operation\_failed"/>

<enumeration value="operation\_failed\_invalid\_input\_parameter"/>

<enumeration value="operation\_failed\_internal\_problem"/>

</restriction>

</simpleType>

</element>

</sequence>

</complexType>

</element>

<!-- getAvailableDeltaSynchPoints Fault -->

<element name="getAvailableDeltaSynchPointsFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- triggerDeltaSynchOfCMData Request -->

<element name="triggerDeltaSynchOfCMDataRequest">

<complexType>

<sequence>

<element name="managerReference" type="string" minOccurs="0"/>

<element name="cmDataRequested">

<simpleType>

<restriction base="string">

<enumeration value="DNsOnly"/>

<enumeration value="CompleteDataSet"/>

</restriction>

</simpleType>

</element>

<element name="baseMOInstance" type="xn:dn" minOccurs="0"/>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

<element name="synchPoint" type="dateTime"/>

</sequence>

</complexType>

</element>

<!-- triggerDeltaSynchOfCMData Response -->

<element name="triggerDeltaSynchOfCMDataResponse">

<complexType>

<sequence>

<element name="deltaLists" minOccurs="0">

<complexType>

<sequence>

<element name="startTime" type="dateTime"/>

<element name="endTime" type="dateTime"/>

<element name="listOfCreatedInstances" type="deltaSynchIRPData:listOfInstances"/>

<element name="listOfChangedInstances" type="deltaSynchIRPData:listOfInstances"/>

<element name="listOfDeletedInstances" type="deltaSynchIRPData:listOfInstances"/>

</sequence>

</complexType>

</element>

<element name="newSynchPoint" type="dateTime" minOccurs="0"/>

<element name="status">

<simpleType>

<restriction base="string">

<enumeration value="Success"/>

<enumeration value="Failure"/>

</restriction>

</simpleType>

</element>

<element name="failureReason" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="SynchrPointTooLongAgo"/>

<enumeration value="TooManyChangesFullSynchronizationRecommended"/>

<enumeration value="SynchPointUnknown"/>

<enumeration value="DeltaSynchNotSupportedForCMData"/>

<enumeration value="DeltaSynchForCMDataDeactivated"/>

<enumeration value="operation\_failed"/>

<enumeration value="operation\_failed\_invalid\_input\_parameter"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_managerReference"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_baseMOInstance"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_scope"/>

<enumeration value="operation\_failed\_internal\_problem"/>

</restriction>

</simpleType>

</element>

</sequence>

</complexType>

</element>

<!-- triggerDeltaSynchOfCMData Fault -->

<element name="triggerDeltaSynchOfCMDataFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- triggerDeltaSynchOfAlarms Request -->

<element name="triggerDeltaSynchOfAlarmsRequest">

<complexType>

<sequence>

<element name="managerReference" type="string" minOccurs="0"/>

<element name="alarmDataRequested">

<simpleType>

<restriction base="string">

<enumeration value="AlarmIdsOnly"/>

<enumeration value="CompleteAlarmInformation"/>

</restriction>

</simpleType>

</element>

<element name="baseMOInstance" type="xn:dn" minOccurs="0"/>

<element name="scope" type="nk:ScopeType" minOccurs="0"/>

<element name="synchPoint" type="dateTime"/>

</sequence>

</complexType>

</element>

<!-- triggerDeltaSynchOfAlarms Response -->

<element name="triggerDeltaSynchOfAlarmsResponse">

<complexType>

<sequence>

<element name="deltaLists" minOccurs="0">

<complexType>

<sequence>

<element name="startTime" type="dateTime"/>

<element name="endTime" type="dateTime"/>

<element name="listOfNewAlarms" type="deltaSynchIRPData:listOfAlarms"/>

<element name="listOfChangedAlarms" type="deltaSynchIRPData:listOfAlarms"/>

<element name="listOfDeletedAlarms" type="deltaSynchIRPData:listOfAlarms"/>

</sequence>

</complexType>

</element>

<element name="newSynchPoint" type="dateTime"/>

<element name="status">

<simpleType>

<restriction base="string">

<enumeration value="Success"/>

<enumeration value="Failure"/>

</restriction>

</simpleType>

</element>

<element name="failureReason" minOccurs="0">

<simpleType>

<restriction base="string">

<enumeration value="SynchrPointTooLongAgo"/>

<enumeration value="TooManyChangesFullSynchronizationRecommended"/>

<enumeration value="SynchPointUnknown"/>

<enumeration value="DeltaSynchNotSupportedForAlarmData"/>

<enumeration value="DeltaSynchForAlarmsNotActive"/>

<enumeration value="operation\_failed"/>

<enumeration value="operation\_failed\_invalid\_input\_parameter"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_managerReference"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_baseMOInstance"/>

<enumeration value="operation\_failed\_unsupported\_optional\_input\_parameter\_scope"/>

<enumeration value="operation\_failed\_internal\_problem"/>

</restriction>

</simpleType>

</element>

</sequence>

</complexType>

</element>

<!-- triggerDeltaSynchOfAlarms Fault -->

<element name="triggerDeltaSynchOfAlarmsFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

</schema>

</types>

<message name="manageDeltaSynchronizationRequest">

<part name="parameter" element="deltaSynchIRPData:manageDeltaSynchronizationRequest"/>

</message>

<message name="manageDeltaSynchronizationResponse">

<part name="parameter" element="deltaSynchIRPData:manageDeltaSynchronizationResponse"/>

</message>

<message name="manageDeltaSynchronizationFault">

<part name="parameter" element="deltaSynchIRPData:manageDeltaSynchronizationFault"/>

</message>

<message name="getAvailableDeltaSynchPointsRequest">

<part name="parameter" element="deltaSynchIRPData:getAvailableDeltaSynchPointsRequest"/>

</message>

<message name="getAvailableDeltaSynchPointsResponse">

<part name="parameter" element="deltaSynchIRPData:getAvailableDeltaSynchPointsResponse"/>

</message>

<message name="getAvailableDeltaSynchPointsFault">

<part name="parameter" element="deltaSynchIRPData:getAvailableDeltaSynchPointsFault"/>

</message>

<message name="triggerDeltaSynchOfCMDataRequest">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfCMDataRequest"/>

</message>

<message name="triggerDeltaSynchOfCMDataResponse">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfCMDataResponse"/>

</message>

<message name="triggerDeltaSynchOfCMDataFault">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfCMDataFault"/>

</message>

<message name="triggerDeltaSynchOfAlarmsRequest">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfAlarmsRequest"/>

</message>

<message name="triggerDeltaSynchOfAlarmsResponse">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfAlarmsResponse"/>

</message>

<message name="triggerDeltaSynchOfAlarmsFault">

<part name="parameter" element="deltaSynchIRPData:triggerDeltaSynchOfAlarmsFault"/>

</message>

<portType name="DeltaSynchGenericParts">

<operation name="manageDeltaSynchronization">

<input message="deltaSynchIRPSystem:manageDeltaSynchronizationRequest"/>

<output message="deltaSynchIRPSystem:manageDeltaSynchronizationResponse"/>

<fault name="manageDeltaSynchronizationFault" message="deltaSynchIRPSystem:manageDeltaSynchronizationFault"/>

</operation>

<operation name="getAvailableDeltaSynchPoints">

<input message="deltaSynchIRPSystem:getAvailableDeltaSynchPointsRequest"/>

<output message="deltaSynchIRPSystem:getAvailableDeltaSynchPointsResponse"/>

<fault name="getAvailableDeltaSynchPointsFault" message="deltaSynchIRPSystem:getAvailableDeltaSynchPointsFault"/>

</operation>

</portType>

<portType name="DeltaSynchOfCMData">

<operation name="triggerDeltaSynchOfCMData">

<input message="deltaSynchIRPSystem:triggerDeltaSynchOfCMDataRequest"/>

<output message="deltaSynchIRPSystem:triggerDeltaSynchOfCMDataResponse"/>

<fault name="triggerDeltaSynchOfCMDataFault" message="deltaSynchIRPSystem:triggerDeltaSynchOfCMDataFault"/>

</operation>

</portType>

<portType name="DeltaSynchOfAlarmData">

<operation name="triggerDeltaSynchOfAlarms">

<input message="deltaSynchIRPSystem:triggerDeltaSynchOfAlarmsRequest"/>

<output message="deltaSynchIRPSystem:triggerDeltaSynchOfAlarmsResponse"/>

<fault name="triggerDeltaSynchOfAlarmsFault" message="deltaSynchIRPSystem:triggerDeltaSynchOfAlarmsFault"/>

</operation>

</portType>

<binding name="DeltaSynchGenericParts" type="deltaSynchIRPSystem:DeltaSynchGenericParts">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="manageDeltaSynchronization">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#manageDeltaSynchronization" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="manageDeltaSynchronizationFault">

<soap:fault name="manageDeltaSynchronizationFault" use="literal"/>

</fault>

</operation>

<operation name="getAvailableDeltaSynchPoints">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#getAvailableDeltaSynchPoints" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getAvailableDeltaSynchPointsFault">

<soap:fault name="getAvailableDeltaSynchPointsFault" use="literal"/>

</fault>

</operation>

</binding>

<binding name="DeltaSynchOfCMData" type="deltaSynchIRPSystem:DeltaSynchOfCMData">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="triggerDeltaSynchOfCMData">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#triggerDeltaSynchOfCMData" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="triggerDeltaSynchOfCMDataFault">

<soap:fault name="triggerDeltaSynchOfCMDataFault" use="literal"/>

</fault>

</operation>

</binding>

<binding name="DeltaSynchOfAlarmData" type="deltaSynchIRPSystem:DeltaSynchOfAlarmData">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="triggerDeltaSynchOfAlarms">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#triggerDeltaSynchOfAlarms" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="triggerDeltaSynchOfAlarmsFault">

<soap:fault name="triggerDeltaSynchOfAlarmsFault" use="literal"/>

</fault>

</operation>

</binding>

<service name="DeltaSynchIRPService">

<port name="DeltaSynchGenericPartsPort" binding="deltaSynchIRPSystem:DeltaSynchGenericParts">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchGenericParts"/>

</port>

<port name="DeltaSynchOfCMDataPort" binding="deltaSynchIRPSystem:DeltaSynchOfCMData">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchOfCMData"/>

</port>

<port name="DeltaSynchOfAlarmDataPort" binding="deltaSynchIRPSystem:DeltaSynchOfAlarmData">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.396#DeltaSynchOfAlarmData"/>

</port>

<port name="GenericIRPPort" binding="genericIRPSystem:GenericIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRP"/>

</port>

<port name="NotificationIRPNtfPort" binding="ntfIRPNtfSystem:NotificationIRPNtf">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtf"/>

</port>

</service>

</definitions>

Annex D (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2010-09 | SA#49 | SP-100507 | -- | -- |  | Presentation to SA for Information and Approval | 1.0.0 |
| 2010-10 | -- | -- | -- | -- |  | Publication | 10.0.0 |
| 2010-12 | SA#50 | SP-100831 | 001 | 1 |  | Correcting IDL syntax errors | 10.1.0 |
| 2011-12 | SA#54 | SP-110717 | 002 | 2 |  | Correct syntax errors in XML schemas and WSDL | 10.2.0 |
| 2012-09 | SA#57 | - | - | - |  | Automatic upgrade from previous Release version 10.2.0 | 11.0.0 |
| 2014-09 | SA#65 | SP-140559 | 003 | - |  | Update the link from Solution Set to Information Service due to the end of Release 12 | 12.0.0 |
| 2016-01 | - | - | - | - |  | Update to Rel-13 version (MCC) | 13.0.0 |
| 2016-06 | SA#72 | SP-160407 | 0004 | - | F | Update the link from IRP Solution Set to IRP Information Service | 13.1.0 |
| 2017-03 | SA#75 | - | - | - |  | Promotion to Release 14 without technical change | 14.0.0 |
| 2017-06 | SA#76 | SP-170502 | 0005 | - | F | Update the link from IRP Solution Set to IRP Information Service | 14.1.0 |
| 2018-06 | - | - | - | - | - | Update to Rel-15 version (MCC) | **15.0.0** |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | **16.0.0** |