3GPP TS 32.366 V16.0.0(2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Telecommunication management;

Entry Point (EP) 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

management, architecture, entry point,

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___Toc335990939)

Introduction [5](#__RefHeading___Toc335990940)

1Scope [6](#__RefHeading___Toc335990941)

2References [6](#__RefHeading___Toc335990942)

3Definitions and abbreviations [7](#__RefHeading___Toc335990943)

3.1Definitions [7](#__RefHeading___Toc335990944)

3.2Abbreviations [7](#__RefHeading___Toc335990945)

4Solution Set definitions [7](#__RefHeading___Toc335990946)

Annex A (normative): CORBA Solution Set [8](#__RefHeading___Toc335990947)

A.1Architectural Features [8](#__RefHeading___Toc335990948)

A.1.1Syntax for Distinguished Names and Versions [8](#__RefHeading___Toc335990949)

A.1.2 Notifications [8](#__RefHeading___Toc335990950)

A.2Mapping [8](#__RefHeading___Toc335990951)

A.2.1Operation and Notification mapping [8](#__RefHeading___Toc335990952)

A.2.2Operation parameter mapping [8](#__RefHeading___Toc335990953)

A.2.3Notification parameter mapping [10](#__RefHeading___Toc335990954)

A.3EPIRPNotification Interface [11](#__RefHeading___Toc335990955)

A.3.1Method push (M) [11](#__RefHeading___Toc335990956)

A.4Solution Set definitions [12](#__RefHeading___Toc335990957)

A.4.1IDL definition structure [12](#__RefHeading___Toc335990958)

A.4.2IDL specification “EPIRPConstDefs.idl” [13](#__RefHeading___Toc335990959)

A.4.3IDL specification “EPIRPSystem.idl” [15](#__RefHeading___Toc335990960)

A.4.4IDL specification “EPIRPNotifications.idl” [17](#__RefHeading___Toc335990961)

A.5Convention when using INS to fulfill part of EPIRP functions [18](#__RefHeading___Toc335990962)

Annex B (normative): XML definitions [20](#__RefHeading___Toc335990963)

B.1Architectural features [20](#__RefHeading___Toc335990964)

B.1.1Syntax for Distinguished Names [20](#__RefHeading___Toc335990965)

B.2Mapping [20](#__RefHeading___Toc335990966)

B.3Solution Set definitions [20](#__RefHeading___Toc335990967)

B.3.1XML definition structure [20](#__RefHeading___Toc335990968)

B.3.2Graphical Representation [20](#__RefHeading___Toc335990969)

B.3.3 XML Schema “ePIRPNotif.xsd” [21](#__RefHeading___Toc335990970)

Annex C (normative): SOAP Solution Set [22](#__RefHeading___Toc335990971)

C.1Architectural Features [22](#__RefHeading___Toc335990972)

C.1.1Syntax for Distinguished Names and versions [22](#__RefHeading___Toc335990973)

C.1.2General [22](#__RefHeading___Toc335990974)

C.2Mapping [23](#__RefHeading___Toc335990975)

C.2.1Operation and Notification mapping [23](#__RefHeading___Toc335990976)

C.2.2Operation parameter mapping [23](#__RefHeading___Toc335990977)

C.2.2.1 Operation getIRPOutline [23](#__RefHeading___Toc335990978)

C.2.2.1.1 Input parameters [23](#__RefHeading___Toc335990979)

C.2.2.1.2 Output parameters [24](#__RefHeading___Toc335990980)

C.2.2.1.3 Fault definition [24](#__RefHeading___Toc335990981)

C.2.2.2 Operation getIRPReference [24](#__RefHeading___Toc335990982)

C.2.2.2.1 Input parameters [24](#__RefHeading___Toc335990983)

C.2.2.2.2 Output parameters [24](#__RefHeading___Toc335990984)

C.2.2.2.3 Fault definition [24](#__RefHeading___Toc335990985)

C.2.2.3 Operation releaseIRPReference [25](#__RefHeading___Toc335990986)

C.2.2.3.1 Input parameters [25](#__RefHeading___Toc335990987)

C.2.2.3.2 Output parameters [25](#__RefHeading___Toc335990988)

C.2.2.3.3 Fault definition [25](#__RefHeading___Toc335990989)

C.3Solution Set definitions [25](#__RefHeading___Toc335990990)

C.3.1WSDL definition structure [25](#__RefHeading___Toc335990991)

C.3.2Graphical Representation [25](#__RefHeading___Toc335990992)

C.3.3WSDL specification “EntryPointIRPSystem.wsdl” [27](#__RefHeading___Toc335990993)

Annex D (informative): Change history [30](#__RefHeading___Toc335990994)

# 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; as identified below:

32.361: "Entry Point (EP) Integration Reference Point (IRP): Requirements"

32.362: "Entry Point (EP) Integration Reference Point (IRP): Information Service (IS)"

**32.366: "Entry Point (EP) Integration Reference Point (IRP): Solution Set (SS) definitions"**

The present document is part of a set of technical specifications defining the Telecommunication Management (TM) of 3G systems. The TM principles are described in 3GPP TS 32.101 [1]. The TM architecture is described in 3GPP TS 32.102 [2]. The other specifications define the interface (Itf-N) between the managing system (manager), which is in general the Network Manager (NM) and the managed system (agent), which is either an Element Manager (EM) or the managed NE itself. The Itf-N is composed of a number of Integration Reference Points (IRPs) defining the information in the agent that is visible for the manager, the operations that the manager may perform on this information and the notifications that are sent from the agent to the manager. EP (Entry Point) IRP is one of these IRPs with special function.

It's difficult for an NM to discover all IRPs in the environment that there are several managed systems and/or if there are multiple IRPs related to each managed systems. This Entry Point is proposed to provide a convenient mechanism for NM to discover the managed systems and their related IRPs.

# 1 Scope

The present document specifies the Solution Set definitions for the IRP whose semantics are specified in the Entry Point IRP: Information Service (3GPP TS 32.362 [6]).

This Solution Set definitions specification is related to 3GPP TS 32.362 V14.0.X [6].

# 2 References

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

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

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

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

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

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

[3] 3GPP TS 32.361: "Telecommunication management; Entry Point (EP) Integration Reference Point (IRP): Requirements".

[4] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP): Information Service (IS)".

[5] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP): Requirements".

[6] 3GPP TS 32.362: "Telecommunication management; Entry Point (EP) Integration Reference Point (IRP): Information Service (IS)".

[7] 3GPP TS 32.306: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Solution Set (SS) definitions".

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

[9] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[10] ITU-T Recommendation Q.816.1 Recommendation "CORBA based TMN services: Extensions to support coarse‑grained interfaces".

[11] OMG TC Document telecom/98-12-09: "CORBAservices: Common Object Services Specification". <http://www.omg.org/technology/documents/>

[12] ISO 8859-1: "Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1".

[13] 3GPP TS 32.336: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Solution Set (SS) definitions".

[14] 3GPP TS 32.331: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Requirements".

[15] 3GPP TS 32.316: "Telecommunication management; Generic Integration Reference Point (IRP) management; Solution Set (SS) definitions".

[16] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

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

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

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

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

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions defined in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.361 [3], 3GPP TS 32.331 [14], 3GPP TS 32.150 [16] and the following apply:

**IRP document version number string (or "IRPVersion"):** See 3GPP TS 32.311 [5].

## 3.2 Abbreviations

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

CORBA Common Object Request Broker Architecture

DN Distinguished Name

EM Element Manager

EP Entry Point

EPIRP Entry Point IRP

IRP Integration Reference Point

IOC Information Object Class

IS Information Service

NE Network Element

NL Notification Log

NM Network Manager

SS Solution Set

WSDL Web Service Description Language

XML eXtensible Markup Language

# 4 Solution Set definitions

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

- 3GPP EP IRP CORBA SS (Annex A)

- 3GPP EP IRP XML definitions (Annex B)

- 3GPP EP IRP SOAP Solution Set (Annex C)

Annex A (normative):   
CORBA Solution Set

This annex contains the CORBA Solution Set for the IRP whose semantics is specified in EP IRP: Information Service (TS 32.362 [6]).

# A.1 Architectural Features

The overall architectural feature of EP IRP is specified in 3GPP TS 32.362 [6].

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

## A.1.1 Syntax for Distinguished Names and Versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [9].

The version of this IRP is represented as a string (see also clause 3 for versions).

## A.1.2 Notifications

Notifications are sent according to the Notification IRP: CORBA SS (see 3GPP TS 32.306 [7]).

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

# A.2 Mapping

## A.2.1 Operation and Notification mapping

EPIRP: IS (3GPP TS 32.362 [6]) defines semantics of operation and notification visible across the EPIRP.

Table A.2.1 indicates the mapping of these operations and notifications to their equivalents defined in this SS.

Table A.2.1: Mapping from IS Operations and Notification to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operations/ notification TS 32.362 [6] | SS Method | Qualifier |
| getIRPOutline | get\_irp\_outline | M |
| getIRPReference | get\_irp\_reference | M |
| releaseIRPReference | release\_irp\_reference | M |
| notifyIRPInfoChanges | push\_structured\_event (See clause A.3.1) | M |
| getIRPVersion (see note) | get\_ep\_irp\_versions | M |
| getOperationProfile (see note) | get\_ep\_irp\_operations\_profile | O |
| getNotificationProfile (see note) | get\_ep\_irp\_notification\_profile | O |
| NOTE: This operation is of ManagedGenericIRP IOC specified in 3GPP TS 32.312 [4]. The EPIRP IOC of 3GPP TS 32.362 [6] inherits from it. | | |

## A.2.2 Operation parameter mapping

The EPIRP: IS 3GPP TS 32.362 [6] defines semantics of parameters carried in operations across the EPIRP. The following tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| iRPVersion | ManagedGenericIRPConstDefs::VersionNumber irp\_version | M |
| supportedIRPList | EPIRPConstDefs::SupportedIRPList supported\_irp\_list | M |
| status | Return value of type EPIRPConstDefs::Result  Exception:  GetIRPOutline, InvalidIRPVersion | M |

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

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerIdentifier | EPIRPConstDefs::ManagerIdentifier manager\_identifier | M |
| systemDn | EPIRPConstDefs::DN system\_dn | M |
| rDN | EPIRPConstDefs::RDN r\_DN | M |
| iRPReference | string irp\_reference (stringified IOR) | M |
| status | Return value of type EPIRPConstDefs::Result  Exception:  GetIRPReference, ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.3: Mapping from IS releaseIRPReference parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managerIdentifier | EPIRPConstDefs::ManagerIdentifier manager\_identifier | M |
| iRPReference | string irp\_reference (stringified IOR) | M |
| status | Return value of type EPIRPConstDefs::Result  Exception:  ReleaseIRPReference, UnknownIRPReference | M |

Table A.2.2.4: Mapping from IS getIRPVersion parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| versionNumberSet | Return value of type ManagedGenericIRPConstDefs::VersionNumberSet | M |
| status | Exception:  GetEPIRPVersions | M |

Table A.2.2.5: Mapping from IS getOperationProfile parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| iRPVersion | ManagedGenericIRPConstDefs::VersionNumber irp\_version | M |
| operationNameProfile, operationParameterProfile | Return value of type ManagedGenericIRPConstDefs::MethodList | M |
| status | Exception:  GetEPIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.6: Mapping from IS getNotificationProfile parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| iRPVersion | ManagedGenericIRPConstDefs::VersionNumber irp\_version | M |
| notificationNameProfile, notificationParameterProfile | Return value of type ManagedGenericIRPConstDefs::MethodList | M |
| status | Exception:  GetEPIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter | M |

## A.2.3 Notification parameter mapping

The EPIRP: IS (3GPP TS 32.362 [6]) 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 [8]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [8], is:

Header

Fixed Header

domain\_name

type\_name

event\_name

Variable Header

Body

filterable\_body\_fields

remaining\_body

The following table lists all OMG Structured Event attributes in the second column. The first column identifies the EPIRP: IS (3GPP TS 32.362 [6]) defined notification parameters.

Table A.2.3: Mapping for notifyIRPInfoChanges

| 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 ET\_IRPINFO\_CHANGES of module of EPIRPSystem. |
| 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 [7]). |
| 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 [7]). |
| 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 [7]). |
| 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 [7]). |
| iRPDn | One NV pair of remaining\_body | M | Name of NV pair is the IRP\_DN of interface NotifyIRPInfoChanges of module EPIRPNotifications.  Value of NV pair is a EPIRPConstDefs::DN. |
| changeMode | One NV pair of remaining\_body | M | Name of NV pair is the CHANGE\_MODE of interface NotifyIRPInfoChanges of module EPIRPNotifications.  Value of NV pair is a EPIRPConstDefs::ChangeMode. |
| additionalText | One NV pair of remaining\_body | M | Name of NV pair is the ADDITIONAL\_TEXT of interface NotifyIRPInfoChanges of module EPIRPNotifications.  Value of NV pair is a string. |

# A.3 EPIRPNotification Interface

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

## A.3.1 Method push (M)

module CosNotifyComm {

…

Interface SequencePushConsumer : NotifyPublish {

void push\_structured\_events(

in CosNotification::EventBatch notifications)

raises( CosEventComm::Disconnected);

…

}; // SequencePushConsumer

…

}; // CosNotifyComm

1) The push\_structured\_events method takes an input parameter of type EventBatch as defined in the OMG CosNotification module (OMG Notification Service [8]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter shall contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

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

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

4) IRPAgent may push EventBatch with only one Structured Event.

# A.4 Solution Set definitions

## A.4.1 IDL definition structure

Clause A.4.2 defines the constants and types used by the EP IRP.

Clause A.4.3 defines the operations which are performed by the EP IRP agent.

Clause A.4.4 defines the notifications which are performed by the EP IRP agent.

## A.4.2 IDL specification “EPIRPConstDefs.idl”

//File: EPIRPConstDefs.idl

#ifndef \_EP\_IRP\_CONST\_DEFS\_IDL\_

#define \_EP\_IRP\_CONST\_DEFS\_IDL\_

#include <ManagedGenericIRPConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: EPIRPSystem

\*/

module EPIRPConstDefs

{

enum Result {OK, FAILURE};

/\*

The RDN carries an optional instance class name and a mandatory

instance identifier. This type (a string) may contain 0 or 1

equal sign. If an equal sign is present, the substring before

the equal sign is the class name, and the substring after the

equal sign is the instance identifier. If the equal sign is

absent, the entire string is the instance identifier.

\*/

typedef string RDN;

typedef string DN;

typedef sequence<DN> DNList;

/\*

IRPManagementScopeOpt is a type carrying an optional parameter.

If the boolean is TRUE, then the value is present.

Otherwise the value is absent.

\*/

union IRPManagementScopeOpt switch (boolean)

{

case TRUE: DNList value;

};

/\*

The IRPElement defines the structure to be returned as part of

get\_irp\_outline().

\*/

struct IRPElement

{

RDN r\_DN;

ManagedGenericIRPConstDefs::VersionNumberSet irp\_versions;

IRPManagementScopeOpt irp\_management\_scope;

};

/\*

List of all IRPElement and their associated parameters.

\*/

typedef sequence<IRPElement> IRPList;

struct SupportedIRP

{

DN system\_dn;

IRPList irp\_list;

};

typedef sequence<SupportedIRP> SupportedIRPList;

typedef string ManagerIdentifier;

enum ChangeMode {REGISTER, DEREGISTER, MODIFY};

/\*

Define the parameters specified in

the notifyEpInfoChanges notification.

\*/

interface AttributeNameValue

{

const string IRP\_DN = "IRP\_DN";

const string CHANGE\_MODE = "CHANGE\_MODE";

const string ADDITIONAL\_TEXT = "ADDITIONAL\_TEXT";

};

};

#endif // \_EP\_IRP\_CONST\_DEFS\_IDL\_

## A.4.3 IDL specification “EPIRPSystem.idl”

//File: EPIRPSystem.idl

#ifndef \_EP\_IRP\_SYSTEM\_IDL\_

#define \_EP\_IRP\_SYSTEM\_IDL\_

#include <ManagedGenericIRPConstDefs.idl>

#include <ManagedGenericIRPSystem.idl>

#include <EPIRPConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: EPIRPSystem

\*/

module EPIRPSystem

{

exception InvalidIRPVersion { string reason; };

exception UnknownIRPReference { string reason; };

/\*

System fails to complete the operation. System can provide reason

to qualify the exception. The semantics carried in reason

is outside the scope of this IRP.

\*/

exception GetIRPOutline { string reason; };

exception GetIRPReference { string reason; };

exception ReleaseIRPReference { string reason; };

exception GetEPIRPVersions { string reason; };

exception GetEPIRPOperationsProfile { string reason; };

exception GetEPIRPNotificationProfile { string reason; };

interface EPIRP

{

/\*\*

\* The IRPManager uses this operation to request the EPIRP to

\* return the outline information of the supported IRPs. The EPIRP

\* shall return the outline information of all the IRPs supported by the

\* IRPAgent that contains the EPIRP. The EPIRP may

\* additionally return the outline information of all the IRPs supported

\* by other IRPAgents.

\*/

EPIRPConstDefs::Result get\_irp\_outline(

in ManagedGenericIRPConstDefs::VersionNumber irp\_version,

out EPIRPConstDefs::SupportedIRPList supported\_irp\_list

)

raises (GetIRPOutline, InvalidIRPVersion);

/\*\*

\* The IRPManager uses this operation to request the EPIRP to return

\* the stringified IOR of the IRP identified by system\_dn and r\_DN.

\*/

EPIRPConstDefs::Result get\_irp\_reference(

in EPIRPConstDefs::ManagerIdentifier manager\_identifier,

in EPIRPConstDefs::DN system\_dn,

in EPIRPConstDefs::RDN r\_DN,

out string irp\_reference

)

raises (GetIRPReference,

ManagedGenericIRPSystem::InvalidParameter);

/\*\*

\* The IRPManager uses this operation to request the IRPAgent to

\* release a specific IRP reference. Whether the IRP reference

\* is really released or not in the IRPAgent is outside the

\* scope of this document.

\*/

EPIRPConstDefs::Result release\_irp\_reference(

in EPIRPConstDefs::ManagerIdentifier manager\_identifier,

in string irp\_reference

)

raises (ReleaseIRPReference,

UnknownIRPReference);

/\*\*

\* Return the list of all supported EPIRP versions.

\*/

ManagedGenericIRPConstDefs::VersionNumberSet get\_ep\_irp\_versions (

)

raises (GetEPIRPVersions);

/\*\*

\* Return the list of all supported operations and their supported

\* parameters for a specific EPIRP version.

\*/

ManagedGenericIRPConstDefs::MethodList get\_ep\_irp\_operations\_profile (

in ManagedGenericIRPConstDefs::VersionNumber irp\_version

)

raises (GetEPIRPOperationsProfile,

ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*\*

\* Return the list of all supported notifications and their supported

\* parameters for a specific EPIRP version.

\*/

ManagedGenericIRPConstDefs::MethodList get\_ep\_irp\_notification\_profile

(

in ManagedGenericIRPConstDefs::VersionNumber irp\_version

)

raises (GetEPIRPNotificationProfile,

ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

};

};

#endif // \_EP\_IRP\_SYSTEM\_IDL\_

## A.4.4 IDL specification “EPIRPNotifications.idl”

//File: EPIRPNotifications.idl

#ifndef \_EP\_IRP\_NOTIFICATIONS\_IDL\_

#define \_EP\_IRP\_NOTIFICATIONS\_IDL\_

#include <NotificationIRPNotifications.idl>

#include <EPIRPConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: EPIRPNotifications

\*/

module EPIRPNotifications

{

interface NotifyIRPInfoChanges: NotificationIRPNotifications::Notify

{

const string ET\_IRPINFO\_CHANGES = "notifyIrpInfoChanges";

/\*\*

\* This constant defines the name of the iRPDn property.

\* The data type for the value of this property

\* is DN.

\*/

const string IRP\_DN =

EPIRPConstDefs::AttributeNameValue::IRP\_DN;

/\*\*

\* This constant defines the name of the changeMode property.

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

\*/

const string CHANGE\_MODE =

EPIRPConstDefs::AttributeNameValue::CHANGE\_MODE;

/\*\*

\* This constant defines the name of the additionalText property.

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

\*/

const string ADDITIONAL\_TEXT =

EPIRPConstDefs::AttributeNameValue::ADDITIONAL\_TEXT;

};

};

#endif // \_EP\_IRP\_NOTIFICATIONS\_IDL\_

# A.5 Convention when using INS to fulfill part of EPIRP functions

The implementation of the EPIRP and in particular, the management of CORBA object references within EPIRP, is not a subject matter for 3GPP standardization.

ITU-T SG4 Framework for CORBA-Based Telecommunications Management Network Interfaces (ITU-T Recommendation Q.816.1 [10]) uses OMG Interoperable Naming Service (INS) [11] for the management of CORBA object references. Furthermore, it specifies a convention to name and populate the CORBA object entries within the INS.

This Annex notes that, in the event that an EPIRP implementation uses INS to fulfill part of EPIRP functions, it is advantageous to populate the INS using the ITU-T defined convention.

**Convention**

The OMG INS CORBA *name component* (in short, called CORBA compound name) has the following IDL definition:

// IDL

typedef string Istring;

struct NameComponent {

Istring id;

Istring kind;

};

Istring is a placeholder for a future IDL internationalized string. The id and kind attributes must be composed of characters from the ISO 8859-1 [12] character set, excluding the null character and other non-printable characters. The strings cannot exceed 255 characters. The id attribute cannot be an empty string but the kind attribute can be an empty string.

The CORBA compound name (see Name below) is defined as a sequence of name components:

// IDL

typedef sequence<NameComponent> Name;

The 3GPP defined DN (in short, called DN) of a managed object is represented by the CORBA *compound name*. For example, a DN, quoted from 3GPP TS 32.300 [9], say “DC=se.companyZ.lmc,Network=9,SubNetwork=1,IRPAgent=1,AlarmIRP.iRPId=2”, shall be represented by a sequence of 6 *name components* where the *id* and *kind* of the first *name component* shall be “se\_companyZ\_lmc” and “DC” respectively. The CORBA *compound name*, shall be:

|  |  |  |
| --- | --- | --- |
| index | kind | id |
| 0 | “DC” | “se\_companyZ\_lmc” |
| 1 | “Network” | “9” |
| 2 | “SubNetwork” | “1” |
| 3 | “IRPAgent” | “1” |
| 4 | “AlarmIRP.iRPId” | “2” |
| 5 | “” | “Object” |

The CORBA *compound name*, in stringified name form, shall be “se\_companyZ\_lmc.DC/9.Network/1.SubNetwork/1.IRPAgent/2.AlarmIRP.iRPId/Object”.

NOTE 1: DN appears in interactions (e.g. operations, notifications) across the Itf‑N.

NOTE 2: The CORBA compound name is used internally with the IRPAgent (and its INS) and does not appear in interactions across the Itf-N.

The use of the last row of the CORBA *compound name*, i.e. kind == “” and id ==”Object”, is in accordance to the convention standardized by ITU-T Recommendation Q.816.1 [10]. According to convention standardized by ITU-T Recommendation Q.816.1 [10], the use of index 0 to 4 inclusive is to indicate the naming context of the object and the use of index 0 to 5 inclusive is to indicate the object itself.

DN DC component is composed of multiple words separated by separator, i.e. a dot. It is suggested that the applications (e.g. IRPAgent codes) that process CORBA compound name and DN should map the dot, used as separator in DN DC component, with underscore. This mapping is necessary because in the stringified CORBA *compound name*, the dot is used for the separation of *id* and *kind*. This replacement rule also implies that underscore should not be used as character of DC words.

Annex B (normative):   
XML definitions

The annex specifies the XML Definitions for the EP Integration Reference Point (IRP) as it applies to Itf-N, in accordance with EP IRP IS definitions [6], for usage with the Notification Log IRP XML Definitions [13].

# B.1 Architectural features

The overall architectural feature of EP IRP is specified in 3GPP TS 32.352 [6].

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

## B.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [9].

# B.2 Mapping

Not present in the current version of this specification.

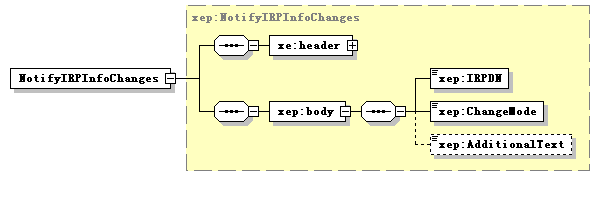
# B.3 Solution Set definitions

## B.3.1 XML definition structure

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

Clause B.3.3 provides XML definitions of EP IRP notifications as defined in 3GPP TS 32.362 [6].

## B.3.2 Graphical Representation



NOTE: The use of XML schema key word “sequence” to support IS-defined set (not sequence) is for the purpose of XML processor efficiency. This shall not imply the use of “sequence” in other technology.

### B.3.3 XML Schema “ePIRPNotif.xsd”

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

<!--

3GPP TS 32.366 EPIRP Notification XML Schema

ePIRPNotif.xsd

-->

<schema xmlns:xep="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#ePIRPNotif" xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification" xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm" xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#ePIRPNotif" elementFormDefault="qualified" attributeFormDefault="unqualified">

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

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

<simpleType name="ChangeMode">

<restriction base="string">

<enumeration value="Register"/>

<enumeration value="Deregister"/>

<enumeration value="Modify"/>

</restriction>

</simpleType>

<complexType name="NotifyIRPInfoChanges">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

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

<element name="ChangeMode" type="xep:ChangeMode"/>

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

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<element name="NotifyIRPInfoChanges" type="xep:NotifyIRPInfoChanges"/>

</schema>

Annex C (normative):  
SOAP Solution Set

The overall architectural feature of the Entry Point IRP is specified in 3GPP TS 32.362 [6]. This clause specifies features that are specific to the SOAP solution set.

# C.1 Architectural Features

## C.1.1 Syntax for Distinguished Names and versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [9].

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

## C.1.2 General

The SOAP 1.1 specification [17] and WSDL 1.1 specification [19] are supported.

The SOAP 1.2 specification [20] 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 [18]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

The Entry Point IRP SOAP SS uses the Notification IRP SOAP SS of 3GPP TS 32.306 [7]. The IRPAgent shall support the push interface model, which means that the IRPAgent sends Entry Point notifications to the IRPManager as soon as new events occur. The IRPManager does not need to check ("pull") for events.

This specification uses a number of namespace prefixes throughout that are listed in Table C.1.2

Table C.1.2: Prefixes and Namespaces used in this specification

|  |  |
| --- | --- |
| **PREFIX** | **NAMESPACE** |
| (no prefix) | http://schemas.xmlsoap.org/wsdl/ |
| soap | http://schemas.xmlsoap.org/wsdl/soap/ |
| entryPointIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPSystem |
| entryPointIRPData | http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPData |
| 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 |
| genericIRPData | http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPData |
| ntfIRPNtfSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem |

# C.2 Mapping

## C.2.1 Operation and Notification mapping

The Entry Point IRP IS (3GPP TS 32.362 [6]) defines the operations and their semantics.

Table C.2.1 maps the operations defined in the Entry Point IRP IS to their equivalent types, messages, port type operation, and binding operation in this Solution Set (SS).

Table C.2.1 also maps the notifications of the Entry Point IRP IS, as well as inherited operations.

Table C.2.1 also qualifies if an operation is Mandatory (M) or Optional (O).

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

|  |  |  |  |
| --- | --- | --- | --- |
| IS Operation in 3GPP TS 32.362 [6] | SS: Operation for WSDL port type and WSDL binding | SS: Port of EntryPointIRPService | Qualifier |
| getIRPOutline | getIRPOutline (note 1) | EntryPointIRPPort | M |
| getIRPReference | getIRPReference (note 1) | EntryPointIRPPort | M |
| releaseIRPReference | releaseIRPReference (note 1) | EntryPointIRPPort | M |
| notifyIRPInfoChanges | notify (note 2) | NotificationIRPNtfPort | M |
| getIRPVersion (note 3) | See TS 32.316 [15] | GenericIRPPort | M |
| getOperationProfile (note 3) | See TS 32.316 [15] | GenericIRPPort | O |
| getNotificationProfile (note 3) | See TS 32.316 [15] | GenericIRPPort | O |
| NOTE 1: The operation is under the port type entryPointIRPSystem:EntryPointIRPPortType and under the binding entryPointIRPSystem:EntryPointIRPBinding. | | | |
| NOTE 2: 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 [7]. This binding is linked to a port of the EntryPointIRPService as indicated in the table above. | | | |
| NOTE 3: The IS operation is inherited from the ManagedGenericIRP IOC specified in 3GPP TS 32.312 [4].  This inheritance is by the EPIRP IOC of 3GPP TS 32.362 [6] inheriting from the ManagedGenericIRP IOC. The corresponding binding is linked to a port of the EntryPointIRPService as indicated in the table above. | | | |

## C.2.2 Operation parameter mapping

The Entry Point IRP IS (3GPP TS 32.362 [6]) 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.

### C.2.2.1 Operation getIRPOutline

#### C.2.2.1.1 Input parameters

Table C.2.2.1.1: Mapping from IS getIRPOutline input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| iRPVersion | iRPVersion | M |

#### C.2.2.1.2 Output parameters

Table C.2.2.1.2: Mapping from IS getIRPOutline output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| supportedIRPList | supportedIRPList | M |
| status | status | M |

#### C.2.2.1.3 Fault definition

Table C.2.2.1.3: Mapping from IS getIRPOutline exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| invalidIRPVersion | InvalidIRPVersion | M |
| operationFailed | OperationFailed | M |

### C.2.2.2 Operation getIRPReference

#### C.2.2.2.1 Input parameters

TableC.2.2.2.1: Mapping from IS getIRPReference input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| managerIdentifier | managerIdentifier | M |
| systemDN | systemDN | M |
| rDN | rDN | M |

#### C.2.2.2.2 Output parameters

TableC.2.2.2.2: Mapping from IS getIRPReference output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| iRPReference | iRPReference | M |
| status | status | M |

#### C.2.2.2.3 Fault definition

TableC.2.2.2.3: Mapping from IS getIRPReference exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| invalidRequestedParameters | InvalidRequestedParameters | M |
| operationFailed | OperationFailed | M |

### C.2.2.3 Operation releaseIRPReference

#### C.2.2.3.1 Input parameters

TableC.2.2.3.1: Mapping from IS releaseIRPReference input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| managerIdentifier | managerIdentifier | M |
| iRPReference | iRPReference | M |

#### C.2.2.3.2 Output parameters

Table C.2.2.3.2: Mapping from IS releaseIRPReference output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| status | status | M |

#### C.2.2.3.3 Fault definition

Table C.2.2.3.3: Mapping from IS releaseIRPReference exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| unknownIRPReference | UnknownIRPReference | M |
| operationFailed | OperationFailed | M |

# C.3 Solution Set definitions

## C.3.1 WSDL definition structure

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

Clause C.3.3 defines the services which are supported the EP 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.

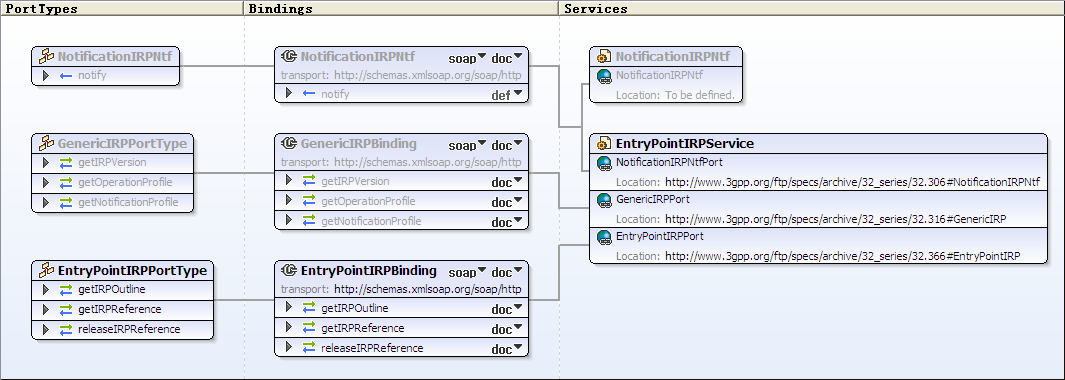


Figure C.3.2: Entry Point IRP SOAP Solution Set WSDL structure

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

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

<!--

3GPP TS 32.366 Entry Point IRP SOAP Solution Set

-->

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:entryPointIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPSystem" xmlns:entryPointIRPData="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPData" 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:genericIRPData="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPData" xmlns:ntfIRPNtfSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPSystem">

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

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

<types>

<schema targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRPData" xmlns="http://www.w3.org/2001/XMLSchema">

<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/32.316#GenericIRPData"/>

<!-- The following types are defined for the Entry Point IRP operations -->

<simpleType name="OperationStatusTwo">

<restriction base="string">

<enumeration value="OperationSucceeded"/>

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

<complexType name="IRPManagementScope">

<sequence>

<element name="subTreeDN" type="xn:dn" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="IRPElement">

<sequence>

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

<element name="iRPVersionSet" type="genericIRPData:VersionNumberSetType"/>

<element name="iRPManagementScope" type="entryPointIRPData:IRPManagementScope" minOccurs="0"/>

</sequence>

</complexType>

<complexType name="IRPList">

<sequence>

<element name="iRPElement" type="entryPointIRPData:IRPElement" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="SupportedIRP">

<sequence>

<element ref="xn:systemDN"/>

<element name="iRPList" type="entryPointIRPData:IRPList"/>

</sequence>

</complexType>

<complexType name="SupportedIRPList">

<sequence>

<element name="supportedIRP" type="entryPointIRPData:SupportedIRP" maxOccurs="unbounded"/>

</sequence>

</complexType>

<!-- getIRPOutline Request-->

<element name="getIRPOutline">

<complexType>

<sequence>

<element name="iRPVersion" type="genericIRPData:VersionNumberType"/>

</sequence>

</complexType>

</element>

<!-- getIRPOutline Response -->

<element name="getIRPOutlineResponse">

<complexType>

<sequence>

<element name="supportedIRPList" type="entryPointIRPData:SupportedIRPList"/>

<element name="status" type="entryPointIRPData:OperationStatusTwo"/>

</sequence>

</complexType>

</element>

<!-- getIRPOutline Fault -->

<element name="getIRPOutlineFault">

<simpleType>

<restriction base="string">

<enumeration value="InvalidIRPVersion"/>

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- getIRPReference Request -->

<element name="getIRPReference">

<complexType>

<sequence>

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

<element ref="xn:systemDN"/>

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

</sequence>

</complexType>

</element>

<!-- getIRPReference Response -->

<element name="getIRPReferenceResponse">

<complexType>

<sequence>

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

<element name="status" type="entryPointIRPData:OperationStatusTwo"/>

</sequence>

</complexType>

</element>

<!-- getIRPReference Fault -->

<element name="getIRPReferenceFault">

<simpleType>

<restriction base="string">

<enumeration value="InvalidRequestedParameters"/>

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- releaseIRPReference Request -->

<element name="releaseIRPReference ">

<complexType>

<sequence>

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

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

</sequence>

</complexType>

</element>

<!-- releaseIRPReference Response -->

<element name="releaseIRPReferenceResponse">

<complexType>

<sequence>

<element name="status" type="entryPointIRPData:OperationStatusTwo"/>

</sequence>

</complexType>

</element>

<!-- releaseIRPReference Fault -->

<element name="releaseIRPReferenceFault">

<simpleType>

<restriction base="string">

<enumeration value="UnknownIRPReference"/>

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

</schema>

</types>

<message name="getIRPOutline">

<part name="parameter" element="entryPointIRPData:getIRPOutline"/>

</message>

<message name="getIRPOutlineResponse">

<part name="parameter" element="entryPointIRPData:getIRPOutlineResponse"/>

</message>

<message name="getIRPOutlineFault">

<part name="parameter" element="entryPointIRPData:getIRPOutlineFault"/>

</message>

<message name="getIRPReference">

<part name="parameter" element="entryPointIRPData:getIRPReference"/>

</message>

<message name="getIRPReferenceResponse">

<part name="parameter" element="entryPointIRPData:getIRPReferenceResponse"/>

</message>

<message name="getIRPReferenceFault">

<part name="parameter" element="entryPointIRPData:getIRPReferenceFault"/>

</message>

<message name="releaseIRPReference">

<part name="parameter" element="entryPointIRPData:releaseIRPReference"/>

</message>

<message name="releaseIRPReferenceResponse">

<part name="parameter" element="entryPointIRPData:releaseIRPReferenceResponse"/>

</message>

<message name="releaseIRPReferenceFault">

<part name="parameter" element="entryPointIRPData:releaseIRPReferenceFault"/>

</message>

<portType name="EntryPointIRPPortType">

<operation name="getIRPOutline">

<input message="entryPointIRPSystem:getIRPOutline"/>

<output message="entryPointIRPSystem:getIRPOutlineResponse"/>

<fault name="getIRPOutlineFault" message="entryPointIRPSystem:getIRPOutlineFault"/>

</operation>

<operation name="getIRPReference">

<input message="entryPointIRPSystem:getIRPReference"/>

<output message="entryPointIRPSystem:getIRPReferenceResponse"/>

<fault name="getIRPReferenceFault" message="entryPointIRPSystem:getIRPReferenceFault"/>

</operation>

<operation name="releaseIRPReference">

<input message="entryPointIRPSystem:releaseIRPReference"/>

<output message="entryPointIRPSystem:releaseIRPReferenceResponse"/>

<fault name="releaseIRPReferenceFault" message="entryPointIRPSystem:releaseIRPReferenceFault"/>

</operation>

</portType>

<binding name="EntryPointIRPBinding" type="entryPointIRPSystem:EntryPointIRPPortType">

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

<operation name="getIRPOutline">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#getIRPOutline" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getIRPOutlineFault">

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

</fault>

</operation>

<operation name="getIRPReference">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#getIRPReference" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getIRPReferenceFault">

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

</fault>

</operation>

<operation name="releaseIRPReference">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#releaseIRPReference" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="releaseIRPReferenceFault">

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

</fault>

</operation>

</binding>

<service name="EntryPointIRPService">

<port name="EntryPointIRPPort" binding="entryPointIRPSystem:EntryPointIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.366#EntryPointIRP"/>

</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-100510 | -- | -- |  | Presentation to SA for Information and Approval | 1.0.0 |
| 2010-10 | -- | -- | -- | -- |  | Publication | 10.0.0 |
| 2012-09 | -- | -- | -- | -- |  | Automatic upgrade from previous Release version 10.0.0 | 11.0.0 |
| 2014-09 | SA#65 | SP-140559 | 0001 | - |  | 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 | 0002 | - | 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 | 0003 | - | 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 |