

ETSI TS 132 363 V8.2.0 (2009-10)

Technical Specification

**Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Entry Point (EP) Integration Reference Point (IRP);
Common Object Request Broker Architecture (CORBA)
Solution Set (SS)
(3GPP TS 32.363 version 8.2.0 Release 8)**



Reference

RTS/TSGS-0532363v820

Keywords

GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

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

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

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

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

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

© European Telecommunications Standards Institute 2009.
All rights reserved.

DECTTM, **PLUGTESTS**TM, **UMTS**TM, **TIPHON**TM, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

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

LTETM is a Trade Mark of ETSI currently being registered

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

GSM[®] and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

Intellectual Property Rights	2
Foreword.....	2
Foreword.....	4
Introduction	4
1 Scope	5
2 References	5
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	6
4 Architectural features	6
4.1 Notifications	6
4.2 Syntax for Distinguished Names and versions	6
5 Mapping	7
5.1 Operation and Notification mapping	7
5.2 Operation parameter mapping	7
5.3 Notification parameter mapping.....	8
6 EPIRPNotification Interface.....	10
6.1 Method push (M).....	10
Annex A (normative): IDL specifications	11
A.1 IDL specification (file name "EPIRPCConstDefs.idl")	11
A.2 IDL specification (file name "EPIRPSystem.idl")	13
A.3 IDL specification (file name " EPIRPNotifications.idl")	15
Annex B (informative): Convention when using INS to fulfill part of EPIRP functions.....	16
Annex C (informative): Change history	18
History	19

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)"
- 32.363: "Entry Point (EP) Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)"**
- 32.365: "Entry Point (EP) Integration Reference Point (IRP): eXtensible Markup Language (XML) definitions"
- 32.367: "Entry Point (EP) Integration Reference Point (IRP): SOAP Solution Set (SS)"

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

This Solution Set specification is related to 3GPP TS 32.362 V8.0.X.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".
- [2] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [3] 3GPP TS 32.361: "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 32.303: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".
- [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: "CORBA services: 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".

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] 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
NM	Network Manager
SS	Solution Set

4 Architectural features

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

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

4.1 Notifications

Notifications are sent according to the Notification IRP: CORBA SS (see 3GPP TS 32.303 [7]).

The contents of the EPIRP notifications are defined in the present document.

4.2 Syntax for Distinguished Names and versions

The format 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).

5 Mapping

5.1 Operation and Notification mapping

EPIRP: IS (3GPP TS 32.362 [6]) defines semantics of operation and notification visible across the EPIRP. Table 1 indicates the mapping of these operations and notifications to their equivalents defined in this SS.

Table 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 6.1)	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.		

5.2 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 2: 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 3: 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 4: 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 5: Mapping from IS `getIRPVersion` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
versionNumberSet	Return value of type <code>ManagedGenericIRPConstDefs::VersionNumberSet</code>	M
status	Exception: <code>GetEPIRPVersions</code>	M

Table 6: Mapping from IS `getOperationProfile` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
iRPVersion	<code>ManagedGenericIRPConstDefs::VersionNumber irp_version</code>	M
operationNameProfile, operationParameterProfile	Return value of type <code>ManagedGenericIRPConstDefs::MethodList</code>	M
status	Exception: <code>GetEPIRPOperationsProfile</code> , <code>ManagedGenericIRPSystem::OperationNotSupported</code> , <code>ManagedGenericIRPSystem::InvalidParameter</code>	M

Table 7: Mapping from IS `getNotificationProfile` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
iRPVersion	<code>ManagedGenericIRPConstDefs::VersionNumber irp_version</code>	M
notificationNameProfile, notificationParameterProfile	Return value of type <code>ManagedGenericIRPConstDefs::MethodList</code>	M
status	Exception: <code>GetEPIRPNotificationProfile</code> , <code>ManagedGenericIRPSystem::OperationNotSupported</code> , <code>ManagedGenericIRPSystem::InvalidParameter</code>	M

5.3 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 8: 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 32.303 [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 32.303 [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 32.303 [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 32.303 [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.

6 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.

6.1 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.

Annex A (normative): IDL specifications

A.1 IDL specification (file name "EPIRPCConstDefs.idl")

```
//File: EPIRPCConstDefs.idl
#ifndef _EP_IRP_CONST_DEFS_IDL_
#define _EP_IRP_CONST_DEFS_IDL_

#include <ManagedGenericIRPCConstDefs.idl>

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

/* ## Module: EPIRPCSystem
*/
module EPIRPCConstDefs
{
    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;
        ManagedGenericIRPCConstDefs::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.2 IDL specification (file name "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.3 IDL specification (file name " 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_
```


Annex B (informative): 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 C (informative): Change history

Change history								
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Cat	Old	New
Sep 2003	SA_21	SP-030425	--	--	Submitted to TSG SA#21 for Information		1.0.0	
Dec 2003	SA_22	SP-030636	--	--	Submitted to TSG SA#22 for Approval		2.0.0	6.0.0
Jun 2004	SA_24	SP-040248	0001	--	Clarification of return value of getIRPReference and Correction of Distinguished Name (DN) and IDL errors	F	6.0.0	6.1.0
Dec 2004	SA_26	SP-040877	0002	1	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields, Eliminate new definitions in EP IRP CORBA SS	F	6.1.0	6.2.0
Mar 2005	SA_27	SP-050039	0003	--	IDL incompliant to the style guide	F	6.2.0	6.3.0
Mar 2005	SA_27	SP-050039	0004	--	Align EPIRPSYSTEM IDL structure with TS 32.150	F	6.2.0	6.3.0
Sep 2005	SA_29	SP-050461	0005		Align the CORBA SS IDL with TS 32.150 Style Guide	F	6.3.0	6.4.0
Jun 2007	SA_36	--	--	--	Automatic upgrade to Rel-7 (no CR) at freeze of Rel-7. Deleted reference to CMIP SS, discontinued from R7 onwards.	--	6.4.0	7.0.0
Sep 2007	SA_37	SP-070614	0006	--	Use IRP class name to method parameter - Align with 32.362 IS	C	7.0.0	8.0.0
Mar 2009	SA_43	SP-090207	0007	--	Include reference to SOAP Solution Set specification	D	8.0.0	8.1.0
Sep 2009	SA-45	SP-090534	0008	--	Correction of rDN attribute in IDL for CORBA compliance.	F	8.1.0	8.2.0

History

Document history		
V8.0.0	February 2009	Publication
V8.1.0	April 2009	Publication
V8.2.0	October 2009	Publication