Technical Specification Group Services and System Aspects
Meeting #22, Maui, USA, 15-18 December 2003 TSGS#22(03)0636

Source:	**SA5 (Telecom Management)**
Title:	**New Rel-6 TS 32.363 v.200 (Entry Point IRP: CORBA Solution Set) - for Approval**
Document for:	**Approval**
Agenda Item:	**7.5.3**

3GPP TSG-SA5 (Telecom Management) S5-037285
Meeting #36, Shanghai, China, 17-21 September 2003

Presentation of Technical Specification to TSG SA

Presentation to:	**TSG SA Meeting #22**
Document for presentation:	**TS 32.363, Version 2.0.0** **Entry Point IRP: CORBA Solution Set**
Presented for:	**Approval**

Abstract of document:

The present document is part of a TS-family covering:

TS 32.361:	"Entry Point (EP) Integration Reference Point (IRP): Requirements";
TS 32.362:	"Entry Point (EP) Integration Reference Point (IRP): Information Service (IS)";
TS 32.363:	**"Entry Point (EP) Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)";**
TS 32.364:	"Entry Point (EP) Integration Reference Point (IRP): Common Management Information Protocol (CMIP) Solution Set (SS)".

This set of specifications provides an Entry Point mechanism enabling the network manager to discover the References of IRPs in the managed systems for Release 6.

Work done against the WID contained in SP-020754 (Work Item ID: OAM-NIM).

Purpose of This Specification:

This is a Technical Specification defining the CORBA solution set for the Entry Point IRP.

Changes since last presentation to TSG-SA#21:
Completed.

Outstanding Issues:
None.

Contentious Issues:
None.

3GPP TS 32.363 V2.0.0 (2003-12)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Telecommunication management;
Entry Point (EP) Integration Reference Point (IRP):
Common Object Request Broker Architecture (CORBA)
Solution Set (SS)
(Release 6)

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 3GPP Organizational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. 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

entry point, CORBA solution set

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

Contents

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:

TS 32.361:	"Entry Point (EP) Integration Reference Point (IRP): Requirements";
TS 32.362:	"Entry Point (EP) Integration Reference Point (IRP): Information Service (IS)";
TS 32.363:	**"Entry Point (EP) Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)";**
TS 32.364:	"Entry Point (EP) Integration Reference Point (IRP): Common Management Information Protocol (CMIP) 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.

The IRPs defined in 3GPP Release 5 and earlier versions are externally published and the 3GPP specifications of Release 5 and earlier versions do not provide standard mechanism for the NM to discover the references of these top IRPs in the managed systems. The discovery is a vendor-specific matter.

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]).

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 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 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/

[1] 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:

CMIP	Common Management Information Protocol
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	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 iRPVersion	M
supportedIRPList	EPIRPSystem::SupportedIRPListType supportedIRPList	M
status	EPIRPSystem::ResultType Exception: GetIRPOutline, InvalidIRPVersion	M

Table 3: Mapping from IS `getIRPReference` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
managerIdentifier	EPIRPSystem::ManagerIdentifierType managerIdentifier	M
systemDn	EPIRPSystem::SystemDNType systemDn	M
iRPId	EPIRPSystem::IRPIdType irpId	M
iRPReference	string iRPReference	M
status	EPIRPSystem::ResultType Exception: GetIRPReference, InvalidRequestedParameters	M

Table 4: Mapping from IS `releaseIRPReference` parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
managerIdentifier	EPIRPSystem::ManagerIdentifierType managerIdentifier	M
iRPReference	string iRPReference	M
status	EPIRPSystem::ResultType 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 ManagedGenericIRPConstDefs::VersionNumberSet	M
status	Exception: GetEPIRPVersions	M

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

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

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

IS Operation parameter	SS Method parameter	Qualifier
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion	M
notificationNameProfile,notificationParameterProfile	Return value of type ManagedGenericIRPConstDefs::MethodList	M
status	Exception: GetEPIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter	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 filterable_ body_fields	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 filterable_ body_fields	M	Name of NV pair is the IRP_DN of module EPIRPSystem:: AttributeNameValue. Value of NV pair is IRPDnType of module EPIRPSystem.
changeMode	One NV pair of filterable_ body_fields	M	Name of NV pair is the CHANGE_MODE of module EPIRPSystem:: AttributeNameValue. Value of NV pair is a ChangeModeType of module EPIRPSystem.
additionalText	One NV pair of filterable_ body_fields	M	Name of NV pair is the ADDITIONAL_TEXT of module EPIRPSystem:: AttributeNameValue. Value of NV pair is a string.
There is no corresponding IS attribute.	remaining_body		

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 " EPIRPSystem.idl")

```
#ifndef EPIRPSystem_idl
#define EPIRPSystem_idl

#include "NotificationIRPConstDefs.idl"
#include "ManagedGenericIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

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

/* ## Module: EPIRPSystem
This module implements capabilities of EPIRP.
================================================================
*/
module EPIRPSystem
{
  enum ResultType {OK, Failure};

  typedef string IRPIdType;
  typedef string SystemDNType;
  typedef sequence<string> DNListType;

  /*
   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: DNListType value;
  };

  /*
  The IRPElement defines the structure to be returned as part of
  getIRPOutline().
  */
  struct IRPElement
  {
    IRPIdType irpId;
    ManagedGenericIRPConstDefs::VersionNumber irpVersion;
    IRPManagementScopeOpt irpManagementScope;
  };

  /*
   List of all IRPElement and their associated parameters.
  */
  typedef sequence<IRPElement> IRPListType;

  struct SupportedIRPListTypeElement
  {
    SystemDNType systemDN;
    IRPListType irpList;
  };

  typedef sequence<SupportedIRPListTypeElement> SupportedIRPListType;

  typedef string ManagerIdentifierType;

  typedef string IRPDnType;
  enum ChangeModeType {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";
};


const string ET_IRPINFO_CHANGES = "notifyIrpInfoChanges";

exception InvalidIRPVersion { string reason; };
exception InvalidRequestedParameters { 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.
   */
   ResultType    get_IRP_outline(
     in ManagedGenericIRPConstDefs::VersionNumber iRPVersion,
     out SupportedIRPListType supportedIRPList
   )
   raises (GetIRPOutline,InvalidIRPVersion);

   /**
   * The IRPManager uses this operation to request the EPIRP
   * to return a reference for a specific version of a specific IRP.
   */
   ResultType    get_IRP_reference(
     in ManagerIdentifierType managerIdentifier,
     in SystemDNType systemDn,
     in IRPIdType irpId,
     out string iRPReference
   )
   raises (GetIRPReference,
      InvalidRequestedParameters);

   /**
   * 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.
   */
   ResultType    release_IRP_reference(
     in ManagerIdentifierType managerIdentifier,
     in string iRPReference
   )
   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 iRPVersion
      )
      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 iRPVersion
      )
      raises (GetEPIRPNotificationProfile,
              ManagedGenericIRPSystem::OperationNotSupported,
              ManagedGenericIRPSystem::InvalidParameter);
   };


   /**
   *  Constant definitions for the EPInfoChanges notification
   */
   module notifyIRPInfoChanges
   {
     const string MANAGED_OBJECT_CLASS =
       NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;
     const string MANAGED_OBJECT_INSTANCE =
       NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;
     const string NOTIFICATION_ID =
       NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;
     const string EVENT_TIME =
       NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;
     const string SYSTEM_DN =
       NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;
     const string EVENT_TYPE = ET_IRPINFO_CHANGES;

     /**
     * This constant defines the name of the iRPDn property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property
     * is IRPDnType.
     */
     const string IRP_DN = AttributeNameValue::IRP_DN;

     /**
     * This constant defines the name of the changeMode property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property is ChangeModeType.
     */
     const string CHANGE_MODE = AttributeNameValue::CHANGE_MODE;

     /**
     * This constant defines the name of the additionalText property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property is string.
     */
     const string ADDITIONAL_TEXT = AttributeNameValue::ADDITIONAL_TEXT;
     };

};

#endif
```

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=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"	"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/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**	**Old**	**New**
Sep 2003	S_21	SP-030425	--	--	Submitted to TSG SA#21 for Information	1.0.0	
Dec 2003	S_22	SP-030636	--	--	Submitted to TSG SA#22 for Approval	2.0.0	