TSGS#26(04)0804

Source: SA5 (Telecom Management)

Title: Rel-6 CR 32.363 Entry Point (EP) IRP CORBA SS

Document for: Approval

Agenda Item: 7.5.3

Doc1stevel	Specific a	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	WorkitemsI D
SP-040804	32.363	002	--	Rel-6	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	S5-047113	OAM-NIM

3GPP TSG-SA5 (Telecom Management) **S5-047113**
Meeting #40, Sanya, CHINA, 15 - 19 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.363** **CR** **002** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields, Eliminate new definitions in EP IRP CORBA SS
Source:	⌘	**SA5** (edwin.tse@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields). Split the IDL file into 3 separate IDL files for clarity in IDL file intent. Eliminate new definitions.
Summary of change:	⌘	Place IS-defined non-filterable parameters into remaining_body of CORBA structured event. Split the IDL file into 3 IDL files as recommended in IDL Style Guide. Eliminate the new definition for type for Distinguished Name.
Consequences if not approved:	⌘	IRPAgent wastes CPU cycles on non-filterable parameters. IDL readers, familiar with the multiple-IDL-file layouts of all other Yyy IRP CORBA SS specifications, could be confused with the one-IDL-file structure used by EPIRP.

Clauses affected:	⌘	5.2, 5.3 Table 8, A.1

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘

Change in Clause 5.2

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~~EPIRPConstDefs::SupportedIRPListType supportedIRPList	M
status	~~EPIRPSystem~~EPIRPConstDefs::ResultType Exception: GetIRPOutline, InvalidIRPVersion	M

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

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

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

IS Operation parameter	SS Method parameter	Qualifier
managerIdentifier	EPIRPConstDefs::~~EPIRPSystem::~~ManagerIdentifierType managerIdentifier	M
iRPReference	string iRPReference	M
status	EPIRPConstDefs::~~EPIRPSystem::~~ResultType Exception: ReleaseIRPReference, UnknownIRPReference	M

End of change in Clause 5.2

Change in Clause “5.3 Table 8”

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	t~~T~~ype_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 ~~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 remaining_body ~~filterable_ body_fields~~	M	Name of NV pair is the IRP_DN of module EPIRPSystem:: AttributeNameValue. Value of NV pair is ~~IRP~~EPIRPConstDefs::DnType ~~of module EPIRPSystem~~.
changeMode	One NV pair of remaining_body ~~filterable_ body_fields~~	M	Name of NV pair is the CHANGE_MODE of module EPIRPSystem:: AttributeNameValue. Value of NV pair is a EPIRPConstDefs::ChangeModeType ~~of module EPIRPSystem~~.
additionalText	One NV pair of remaining_body ~~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~~		

End of change in Clause “5.3 Tablle 8”

Change in Clause A.1

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

```
// File: EPIRPConstDefs.idl

#ifndef _EPIRPCONSTDEFS_IDL_EPIRPSystem_idl
#define _EPIRPCONSTDEFS_IDL_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 EPIRPConstDefsSystem
{
  enum ResultType {OK, FAILUREailure};

  typedef string IRPIdType;
  typedef string SystemDNTypestring DNType;
  typedef sequence<DNTypestring> 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";
   };
};

#endif _EPIRPCONSTDEFS_IDL_
```

A.2 IDL specification (file name "EPIRPSystem.idl")

```
// File: EPIRPSystem.idl

#ifndef _EPIRPSYSTEM_IDL_
#define _EPIRPSYSTEM_IDL_

#include "ManagedGenericIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"
#include "EPIRPConstDefs.idl"

/* ## Module: EPIRPSystem
*/
module EPIRPSystem
{
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.
     */
     EPIRPConstDefs::ResultType get_IRP_outline(
       in ManagedGenericIRPConstDefs::VersionNumber iRPVersion,
       out EPIRPConstDefs::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.
     */
     EPIRPConstDefs::ResultType get_IRP_reference(
       in EPIRPConstDefs::ManagerIdentifierType managerIdentifier,
       in EPIRPConstDefs::SystemDNType systemDn,
       in EPIRPConstDefs::IRPIdType irpId,
       out string iRPReference
     )
     raises (GetIRPReference,
        ManagedGenericIRPSystem::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.
     */
     EPIRPConstDefs::ResultType release_IRP_reference(
       in EPIRPConstDefs::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);
   };
};

#endif _EPIRPSYSTEM_IDL_
```

A.3 IDL specification (file name " EPIRPNotifications.idl")

```
// File: EPIRPNotifications.idl

#ifndef _EPIRPNOTIFICATIONS_IDL_
#define _EPIRPNOTIFICATIONS_IDL_

#include "NotificationIRPNotifications.idl"
#include "EPIRPConstDefs.idl"

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

/* ## Module: EPIRPNotifications
*/
module EPIRPNotifications
{
   /**
   *  Constant definitions for the EPInfoChanges notification
   */
   interfacemodule notifyIRPInfoChanges: NotificationIRPNotifications::Notify
   {
     const string ET_IRPINFO_CHANGES = "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 IRPDNnType.
     */
     const string IRP_DN =
         EPIRPConstDefs::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 =
         EPIRPConstDefs::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 =
         EPIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

   };
     };

};

#endif _EPIRPNOTIFICATIONS_IDL_
```

End of change in Clause A.1
End of document