

Source: SA5 (Telecom Management)
Title: CR 32363 Entry Point (EP) IRP
Document for: Approval
Agenda Item: 7.5.3

Doc-1st-Level	Spec	CR	R	Phase	Subject	Ca	VerCr	Doc-2nd-Level	Workitem
SP-050039	32.363	003	--	Rel-6	IDL incompliant to the style guide	F	6.2.0	S5-056068	OAM-NIM
SP-050039	32.363	004	--	Rel-6	Align EPIRPSystem IDL structure with TS 32.150	F	6.2.0	S5-056045	OAM-NIM

CHANGE REQUEST

⌘ **32.363 CR 004** ⌘ rev - ⌘ Current version: **6.2.0** ⌘

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 Core Network

Title:	⌘ Align EPIRPSystem IDL structure with TS 32.150		
Source:	⌘ SA5 (Nortel – Suzèle Lariven – lariven@nortelnetworks.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 28/01/2005
Category:	⌘ F	Release:	⌘ Rel-6
	Use <u>one</u> 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 <u>one</u> of the following releases: Ph2 (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) Rel-7 (Release 7)

Reason for change:	⌘ Align EPIRPSystem IDL structure with rules for CORBA Solution Sets described in 32.150.
Summary of change:	⌘ Add pragma prefix after #include lines, and immediately before the module statement ⌘ Add missing comment indication
Consequences if not approved:	⌘ IDL not aligned with rules for CORBA Solution Sets described in 32.150 and can't be compiled

Clauses affected:	⌘ A.2								
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="width: 20px; text-align: center;">⌘</td> <td style="width: 20px; text-align: center;">⌘</td> </tr> <tr> <td style="width: 20px; text-align: center;">⌘</td> <td style="width: 20px; text-align: center;">⌘</td> </tr> <tr> <td style="width: 20px; text-align: center;">⌘</td> <td style="width: 20px; text-align: center;">⌘</td> </tr> </table> Other core specifications ⌘ Test specifications ⌘ O&M Specifications ⌘	Y	N	⌘	⌘	⌘	⌘	⌘	⌘
Y	N								
⌘	⌘								
⌘	⌘								
⌘	⌘								
Other comments:	⌘								

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"

// 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::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
        * the stringified IOR of the IRP identified by systemDn and irpId.
        */
        EPIRPConstDefs::ResultType get_IRP_reference(
            in EPIRPConstDefs::ManagerIdentifierType managerIdentifier,
            in EPIRPConstDefs::DNType systemDn,
            in EPIRPConstDefs::IRPIdType irpId,
            out string irpReference
        )
        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::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__

```

<p>End of Change in Annex A.1 End of Document</p>
--

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	6.0.0
Jun 2004	S_24	SP-040248	001	--	Clarification of return value of getIRPReference and Correction of Distinguished Name (DN) and IDL errors	6.0.0	6.1.0
Dec 2004	S_26	SP-040877	002	1	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields, Eliminate new definitions in EP IRP CORBA SS	6.1.0	6.2.0

CHANGE REQUEST

⌘ **32.363 CR 003** ⌘ rev **-** ⌘ Current version: **6.2.0** ⌘

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 Core Network

Title:	⌘ IDL incompliant to the style guide		
Source:	⌘ SA5 (huangsq@zte.com.cn)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 28/01/2005
Category:	⌘ F	Release:	⌘ Rel-6
	Use <u>one</u> 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 <u>one</u> 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) Rel-7 (Release 7)

Reason for change:	⌘ The IDL does not reflect the format recommended by the style guide (TS 32.150).
Summary of change:	⌘ Add double slash between " #endif " and the macro.
Consequences if not approved:	⌘ The IDL won't conform to the styleguide, and will compile errors when using java compilers (e.g., idlj.exe).

Clauses affected:	⌘ Annex A										
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> </table>	Y	N		X		X		X	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
	X										
	X										
	X										
Other comments:	⌘										

Annex A (normative): IDL specifications

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

```
// File: EPIRPCConstDefs.idl

#ifndef _EPIRPCCONSTDEFS_IDL_
#define _EPIRPCCONSTDEFS_IDL_

#include "ManagedGenericIRPCConstDefs.idl"

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

/* ## Module: EPIRPCSystem
*/
module EPIRPCConstDefs
{
    enum ResultType {OK, FAILURE};

    typedef string IRPIdType;
    typedef string DNType;
    typedef sequence<DNType> 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;
        ManagedGenericIRPCConstDefs::VersionNumberSet irpVersions;
        IRPManagementScopeOpt irpManagementScope;
    };

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

    struct SupportedIRPListTypeElement
    {
        DNType systemDN;
        IRPListType irpList;
    };

    typedef sequence<SupportedIRPListTypeElement> SupportedIRPListType;

    typedef string ManagerIdentifierType;

    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
{
    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::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
        * the stringified IOR of the IRP identified by systemDn and irpId.
        */
        EPIRPConstDefs::ResultType get_IRP_reference(
            in EPIRPConstDefs::ManagerIdentifierType managerIdentifier,
            in EPIRPConstDefs::DNType systemDn,
            in EPIRPConstDefs::IRPIdType irpId,
            out string irpReference
        )
        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::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
{

    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 DNType.
         */
        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 ChangeModeType.
         */
        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 // _EPIRPNOTIFICATIONS_IDL_
```

End of change in Annex A

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	6.0.0
Jun 2004	S_24	SP-040248	001	--	Clarification of return value of getIRPReference and Correction of Distinguished Name (DN) and IDL errors	6.0.0	6.1.0
Dec 2004	S_26	SP-040877	002	1	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields, Eliminate new definitions in EP IRP CORBA SS	6.1.0	6.2.0