
Source: SA5 (Telecom Management)
Title: 4 Rel-4/5 CRs 32303 (Configuration Management (CM);
Notification Integration Reference Point (IRP); CORBA SS)
Document for: Approval
Agenda Item: 7.5.3

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Ver-Current	Doc-2nd-Level	Workitem
SP-030137	32.303	006	-	Rel-4	Corrections to CORBA IDL specification "NotificationIRPSystem"	F	4.3.2	S5-036078	OAM-FM
SP-030137	32.303	007	-	Rel-4	Remove unused suspend_subscription and resume_subscription methods	F	4.3.2	S5-036130	OAM-FM
SP-030137	32.303	008	-	Rel-5	Remove unused suspend_subscription and resume_subscription methods	A	5.1.2	S5-036131	OAM-NIM
SP-030137	32.303	009	-	Rel-5	Corrections of CORBA IDL syntax errors	F	5.1.2	S5-036098	OAM-NIM

CHANGE REQUEST

⌘ **32.303 CR 006** ⌘ rev **-** ⌘ Current version: **4.3.2** ⌘

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:	⌘ Corrections to CORBA IDL specification "NotificationIRPSystem"		
Source:	⌘ S5		
Work item code:	⌘ OAM-FM	Date:	⌘ 28/02/2003
Category:	⌘ F	Release:	⌘ Rel-4
	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)

Reason for change:	⌘ CORBA IDL specification "NotificationIRPSystem" contains inconsistencies with mapping tables of IS parameters to CORBA parameters: <ul style="list-style-type: none"> Wrong type for CORBA parameter "time_tick" in CORBA method "get_subscription_status" Wrong type for CORBA parameter "subscription_id" in CORBA methods "change_subscription_filter", "get_subscription_status", and "detach" Missing generic CORBA exceptions "InvalidParameter" and "ParameterNotSupported" on CORBA method "detach"
Summary of change:	⌘ <ul style="list-style-type: none"> Type of CORBA parameter "time_tick" in CORBA method "get_subscription_status" changed to CORBA type "unsigned long" Type of CORBA parameter "subscription_id" in CORBA methods "change_subscription_filter", "get_subscription_status", and "detach" changed to CORBA type "NotificationIRPCConstDefs::SubscriptionId" Generic CORBA exceptions "InvalidParameter" and "ParameterNotSupported" added on CORBA method "detach"
Consequences if not approved:	⌘ It would prevent implementing Rel-4 CORBA SS of Notification IRP in a fully interoperable manner.

Clauses affected:	⌘ A.4						
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">Y</td> <td style="width: 20px;">N</td> </tr> <tr> <td style="width: 20px;"><input type="checkbox"/></td> <td style="width: 20px;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;"><input type="checkbox"/></td> <td style="width: 20px;"><input checked="" type="checkbox"/></td> </tr> </table> Test specifications	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘			
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;"><input type="checkbox"/></td> <td style="width: 20px;"><input checked="" type="checkbox"/></td> </tr> </table> O&M Specifications	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘			
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
Other comments:	⌘ The Rel-4 changes below have already been applied in Rel-5 through CR SP-020482/S5-026487.						

Change in Annex Subclause A.4

A.4 IDL specification (file name "NotificationIRPSystem.idl")

```

#ifndef NotificationIRPSystem_idl
#define NotificationIRPSystem_idl

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

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

/* ## Module: NotificationIRPSystem
This module implements capabilities of Notification IRP.
=====
*/
module NotificationIRPSystem
{
    /*
    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 GetNotificationIRPVersions { string reason; };
    exception GetNotificationIRPOperationsProfile { string reason; };
    exception GetNotificationIRPNotificationProfile { string reason; };
    exception Attach { string reason; };
    exception DetachException { string reason; };
    exception GetSubscriptionStatus { string reason; };
    exception ChangeSubscriptionFilter { string reason; };
    exception GetNotificationCategories { string reason; };
    exception SuspendSubscription { string reason; };
    exception ResumeSubscription { string reason; };
    exception GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
    exception AtLeastOneNotificationCategoryNotSupported {};

    interface NotificationIRP
    {
        /*
        Return the list of all supported Notification IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet get_notification_IRP_versions
        (
        )
        raises (GetNotificationIRPVersions);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific Notification IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
        )
        raises (GetNotificationIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
    }
}

```

```

/*
Return the list of all supported notifications.
Agent should always throw a ManagedGenericIRPSystem::OperationNotSupported
exception.
Similar method, such as get_alarm_IRP_notification_profile,
is supported in other IRP versions such as Alarm IRP.
*/
ManagedGenericIRPConstDefs::MethodList
    get_notification_IRP_notification_profile (
        in ManagedGenericIRPConstDefs::VersionNumber
        notification_irp_version
    )
raises (GetNotificationIRPNotificationProfile,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Obtain the list of all supported notification categories.
*/
NotificationIRPConstDefs::NotificationCategorySet
    get_notification_categories (
        out NotificationIRPConstDefs::NotificationTypesSet
        notification_type_list
    )
raises (GetNotificationCategories,
        ManagedGenericIRPSystem::OperationNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in string filter
)
raises (Attach, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed,
        AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push_b (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in string filter,
    out CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_pull (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in string filter,
    out CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

/*
Replace the present filter constraint with the one provided.

```

```

*/
void change_subscription_filter (
    in string NotificationIRPConstDefs::SubscriptionId subscription_id,
    in string filter
)
raises (ChangeSubscriptionFilter,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Check the current state of the subscription.
*/
NotificationIRPConstDefs::NotificationCategorySet get_subscription_status
(
    in string NotificationIRPConstDefs::SubscriptionId subscription_id,
    out string filter_in_effect,
    out NotificationIRPConstDefs::SubscriptionState subscription_state,
    out unsigned long time_tick
)
raises (GetSubscriptionStatus,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
    in string manager_reference
)
raises (GetSubscriptionIds,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Suspends the event flow until a resume is issued.
*/
void suspend_subscription (
    in string subscription_id
)
raises (SuspendSubscription,
        ManagedGenericIRPSystem::OperationNotSupported);

/*
Resumes the event flow if it was suspended.
*/
void resume_subscription (
    in string subscription_id
)
raises (ResumeSubscription,
        ManagedGenericIRPSystem::OperationNotSupported);

/*
Terminates the subscription with the agent.
*/
void detach (
    in string manager_reference,
    in string NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (DetachException,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);
};
#endif

```

<p>End of Change in Annex Subclause A.4 End of Document</p>
--

CHANGE REQUEST

⌘ **32.303 CR 009** ⌘ rev **-** ⌘ Current version: **5.1.2** ⌘

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:	⌘ Corrections of CORBA IDL syntax errors		
Source:	⌘ S5		
Work item code:	⌘ OAM-NIM	Date:	⌘ 28/02/2003
Category:	⌘ F	Release:	⌘ Rel-5
	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)

Reason for change:	⌘ CORBA IDL will not compile due to incorrect syntax.		
Summary of change:	⌘ CORBA IDL syntax corrections in the raises clause of the operations 'suspend', 'resume' and 'detach' operations		
Consequences if not approved:	⌘ CORBA IDL will not compile		

Clauses affected:	⌘ Annex A4										
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;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
Other comments:	⌘ Only applies to release 5 specification.										

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.htm>. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Start of Change in Annex A4

A.4 IDL specification (file name "NotificationIRPSystem.idl")

```

#ifndef NotificationIRPSystem_idl
#define NotificationIRPSystem_idl

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

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

/* ## Module: NotificationIRPSystem
This module implements capabilities of Notification IRP.
=====
*/
module NotificationIRPSystem
{
    /*
    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 GetNotificationIRPVersions { string reason; };
    exception GetNotificationIRPOperationsProfile { string reason; };
    exception GetNotificationIRPNotificationProfile { string reason; };
    exception Attach { string reason; };
    exception DetachException { string reason; };
    exception GetSubscriptionStatus { string reason; };
    exception ChangeSubscriptionFilter { string reason; };
    exception GetNotificationCategories { string reason; };
    exception SuspendSubscription { string reason; };
    exception ResumeSubscription { string reason; };
    exception GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
    exception AtLeastOneNotificationCategoryNotSupported {};

    interface NotificationIRP
    {
        /*
        Return the list of all supported Notification IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet get_notification_irp_versions
        (
        )
        raises (GetNotificationIRPVersions);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific Notification IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_notification_irp_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
        )
        raises (GetNotificationIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
    }
}

```

```

/*
Return the list of all supported notifications.
Agent should always throw a ManagedGenericIRPSystem::OperationNotSupported
exception.
Similar method, such as get_alarm_IRP_notification_profile,
is supported in other IRP versions such as Alarm IRP.
*/
ManagedGenericIRPConstDefs::MethodList
    get_notification_IRP_notification_profile (
        in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
    )
raises (GetNotificationIRPNotificationProfile,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Obtain the list of all supported notification categories.
*/
NotificationIRPConstDefs::NotificationCategorySet
    get_notification_categories (
        out NotificationIRPConstDefs::NotificationTypesSet
            notification_type_list
    )
raises (GetNotificationCategories,
        ManagedGenericIRPSystem::OperationNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter
)
raises (Attach, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed,
        AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push_b (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    out CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_pull (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    out CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

/*
Replace the present filter constraint with the one provided.
*/
void change_subscription_filter (
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    in string filter
)
raises (ChangeSubscriptionFilter,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Check the current state of the subscription.

```



```

*/
NotificationIRPConstDefs::NotificationCategorySet get_subscription_status
(
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    out ManagedGenericIRPConstDefs::StringTypeOpt filter_in_effect,
    out NotificationIRPConstDefs::SubscriptionState subscription_state,
    out unsigned long time_tick
)
raises (GetSubscriptionStatus,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
    in string manager_reference
)
raises (GetSubscriptionIds,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Suspends the event flow until a resume is issued.
*/
void suspend_subscription (
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (SuspendSubscription,
        ManagedGenericIRPSystem::OperationNotSupported,++)
        ManagedGenericIRPSystem::InvalidParameter);

/*
Resumes the event flow if it was suspended.
*/
void resume_subscription (
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (ResumeSubscription,
        ManagedGenericIRPSystem::OperationNotSupported,++)
        ManagedGenericIRPSystem::InvalidParameter);

/*
Terminates the subscription with the agent.
*/
void detach (
    in string manager_reference,
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (DetachException,++)
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);
};
#endif

```

<p>End of Change in Annex A4</p> <p>End of Document</p>

CR-Form-v7

CHANGE REQUEST

32.303 CR 008 # rev **-** # Current version: **5.1.2**

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:	# Remove unused suspend_subscription and resume_subscription methods		
Source:	# S5		
Work item code:	# OAM-NIM	Date:	# 28/02/2003
Category:	# A	Release:	# Rel-5
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
	F (correction)		2 (GSM Phase 2)
	A (corresponds to a correction in an earlier release)		R96 (Release 1996)
	B (addition of feature),		R97 (Release 1997)
	C (functional modification of feature)		R98 (Release 1998)
	D (editorial modification)		R99 (Release 1999)
	Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Rel-4 (Release 4)
			Rel-5 (Release 5)
			Rel-6 (Release 6)

Reason for change:	# 32.303 reads as follows for the suspendSubscription operation: If subscription is established using attach_push, there is no SS equivalent. In other words, IRPManager cannot suspend subscription. If subscription is established using attach_push_b, the SS equivalent shall be suspend_connection. This method is defined by OMG Notification Service (OMG TC Document telecom [2]). The IDL specification of this method is not included in Annex A. If IRPAgent supports the optional attach_push_b method, it shall support this method as mandatory. If subscription is established using attach_pull, there is no SS equivalent. Yet, a method called suspend_subscription method has been created. Similarly, a resume_subscription method has been created. Since these cannot be used, they should be removed.
Summary of change:	# Removed the suspend_subscription and resume_subscription methods and the SuspendSubscription and ResumeSubscription exceptions. No in-line text needs to change, since the rest of the document does not discuss these methods.
Consequences if not approved:	# Confusion. What should be done with methods that no one is allowed to use?

Clauses affected:	# Annex A.4				
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;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Y	N				
<input type="checkbox"/>	<input checked="" type="checkbox"/>				
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Test specifications	<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>					
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> O&M Specifications	<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>					

Other comments: ☼ This is the Rel-5 Mirror CR to 32303CR007 S5-036130.

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.htm>. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ☼ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/>. For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

A.4 IDL specification (file name "NotificationIRPSystem.idl")

```

#ifndef NotificationIRPSystem_idl
#define NotificationIRPSystem_idl

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

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

/* ## Module: NotificationIRPSystem
This module implements capabilities of Notification IRP.
=====
*/
module NotificationIRPSystem
{
    /*
    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 GetNotificationIRPVersions { string reason; };
    exception GetNotificationIRPOperationsProfile { string reason; };
    exception GetNotificationIRPNotificationProfile { string reason; };
    exception Attach { string reason; };
    exception DetachException { string reason; };
    exception GetSubscriptionStatus { string reason; };
    exception ChangeSubscriptionFilter { string reason; };
    exception GetNotificationCategories { string reason; };
exception SuspendSubscription { string reason; };
exception ResumeSubscription { string reason; };
    exception GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
    exception AtLeastOneNotificationCategoryNotSupported {};

    interface NotificationIRP
    {
        /*
        Return the list of all supported Notification IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet get_notification_IRP_versions
        (
        )
        raises (GetNotificationIRPVersions);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific Notification IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
        )
        raises (GetNotificationIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

        /*
        Return the list of all supported notifications.
        Agent should always throw a ManagedGenericIRPSystem::OperationNotSupported
        exception.
        Similar method, such as get_alarm_IRP_notification_profile,
        is supported in other IRP versions such as Alarm IRP.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_notification_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
        )
        raises (GetNotificationIRPNotificationProfile,

```

```

        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Obtain the list of all supported notification categories.
*/
NotificationIRPConstDefs::NotificationCategorySet
    get_notification_categories (
        out NotificationIRPConstDefs::NotificationTypesSet
        notification_type_list
    )
    raises (GetNotificationCategories,
        ManagedGenericIRPSystem::OperationNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter
)
    raises (Attach, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed,
        AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push_b (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    out CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference
)
    raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_pull (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
    notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    out CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference
)
    raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

/*
Replace the present filter constraint with the one provided.
*/
void change_subscription_filter (
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    in string filter
)
    raises (ChangeSubscriptionFilter,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Check the current state of the subscription.
*/
NotificationIRPConstDefs::NotificationCategorySet get_subscription_status
(
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    out ManagedGenericIRPConstDefs::StringTypeOpt filter_in_effect,
    out NotificationIRPConstDefs::SubscriptionState subscription_state,
    out unsigned long time_tick
)
    raises (GetSubscriptionStatus,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
    in string manager_reference
)
    raises (GetSubscriptionIds,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

```

```
----- /*  
----- Suspends the event flow until a resume is issued.  
----- */  
----- void suspend_subscription (  
-----     in NotificationIRPConstDefs::SubscriptionId subscription_id  
----- )  
----- raises (SuspendSubscription,  
-----     ManagedGenericIRPSystem::OperationNotSupported)+  
-----     ManagedGenericIRPSystem::InvalidParameter);  
  
----- /*  
----- Resumes the event flow if it was suspended.  
----- */  
----- void resume_subscription (  
-----     in NotificationIRPConstDefs::SubscriptionId subscription_id  
----- )  
----- raises (ResumeSubscription,  
-----     ManagedGenericIRPSystem::OperationNotSupported)+  
-----     ManagedGenericIRPSystem::InvalidParameter);  
  
----- /*  
    Terminates the subscription with the agent.  
    */  
    void detach (  
        in string manager_reference,  
        in NotificationIRPConstDefs::SubscriptionId subscription_id  
    )  
    raises (DetachException);  
        ManagedGenericIRPSystem::ParameterNotSupported,  
        ManagedGenericIRPSystem::InvalidParameter);  
};  
};  
#endif
```

CHANGE REQUEST

32.303 CR 007 # rev **-** # Current version: **4.3.2**

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:	# Remove unused suspend_subscription and resume_subscription methods		
Source:	# S5		
Work item code:	# OAM-FM	Date:	# 28/02/2003
Category:	# F	Release:	# Rel-4
	Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
	F (correction)		2 (GSM Phase 2)
	A (corresponds to a correction in an earlier release)		R96 (Release 1996)
	B (addition of feature),		R97 (Release 1997)
	C (functional modification of feature)		R98 (Release 1998)
	D (editorial modification)		R99 (Release 1999)
	Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Rel-4 (Release 4)
			Rel-5 (Release 5)
			Rel-6 (Release 6)

Reason for change:	# 32.303 reads as follows for the suspendSubscription operation: If subscription is established using attach_push, there is no SS equivalent. In other words, IRPManager cannot suspend subscription. If subscription is established using attach_push_b, the SS equivalent shall be suspend_connection. This method is defined by OMG Notification Service (OMG TC Document telecom [2]). The IDL specification of this method is not included in Annex A. If IRPAgent supports the optional attach_push_b method, it shall support this method as mandatory. If subscription is established using attach_pull, there is no SS equivalent. Yet, a method called suspend_subscription method has been created. Similarly, a resume_subscription method has been created. Since these cannot be used, they should be removed.
Summary of change:	# Removed the suspend_subscription and resume_subscription methods and the SuspendSubscription and ResumeSubscription exceptions. No in-line text needs to change, since the rest of the document does not discuss these methods.
Consequences if not approved:	# Confusion. What should be done with methods that no one is allowed to use?

Clauses affected:	# Annex A.4				
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;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Y	N				
<input type="checkbox"/>	<input checked="" type="checkbox"/>				
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> Test specifications	<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>					
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table> O&M Specifications	<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>					

Other comments: ☞ The Rel-5 Mirror CR is submitted in 32303CR008 S5-036131.

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.htm>.

Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ☞ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/>. For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

A.4 IDL specification (file name "NotificationIRPSystem.idl")

```

#ifndef NotificationIRPSystem_idl
#define NotificationIRPSystem_idl

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

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

/* ## Module: NotificationIRPSystem
This module implements capabilities of Notification IRP.
=====
*/
module NotificationIRPSystem
{
    /*
    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 GetNotificationIRPVersions { string reason; };
    exception GetNotificationIRPOperationsProfile { string reason; };
    exception GetNotificationIRPNotificationProfile { string reason; };
    exception Attach { string reason; };
    exception DetachException { string reason; };
    exception GetSubscriptionStatus { string reason; };
    exception ChangeSubscriptionFilter { string reason; };
    exception GetNotificationCategories { string reason; };
    exception SuspendSubscription { string reason; };
    exception ResumeSubscription { string reason; };
    exception GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
    exception AtLeastOneNotificationCategoryNotSupported {};

    interface NotificationIRP
    {
        /*
        Return the list of all supported Notification IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet get_notification_IRP_versions
        (
        )
        raises (GetNotificationIRPVersions);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific Notification IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
        )
        raises (GetNotificationIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,

```

```

        ManagedGenericIRPSystem::InvalidParameter);

/*
Return the list of all supported notifications.
Agent should always throw a ManagedGenericIRPSystem::OperationNotSupported
exception.
Similar method, such as get_alarm_IRP_notification_profile,
is supported in other IRP versions such as Alarm IRP.
*/
ManagedGenericIRPConstDefs::MethodList
    get_notification_IRP_notification_profile (
        in ManagedGenericIRPConstDefs::VersionNumber
            notification_irp_version
    )
raises (GetNotificationIRPNotificationProfile,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Obtain the list of all supported notification categories.
*/
NotificationIRPConstDefs::NotificationCategorySet
    get_notification_categories (
        out NotificationIRPConstDefs::NotificationTypesSet
            notification_type_list
    )
raises (GetNotificationCategories,
        ManagedGenericIRPSystem::OperationNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in string filter
)
raises (Attach, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed,
        AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_push_b (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in string filter,
    out CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_pull (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in string filter,
    out CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

```

```

/*
Replace the present filter constraint with the one provided.
*/
void change_subscription_filter (
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    in string filter
)
raises (ChangeSubscriptionFilter,
    ManagedGenericIRPSystem::OperationNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);

/*
Check the current state of the subscription.
*/
NotificationIRPConstDefs::NotificationCategorySet get_subscription_status
(
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    out string filter_in_effect,
    out NotificationIRPConstDefs::SubscriptionState subscription_state,
    out unsigned long time_tick
)
raises (GetSubscriptionStatus,
    ManagedGenericIRPSystem::OperationNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);

NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
    in string manager_reference
)
raises (GetSubscriptionIds,
    ManagedGenericIRPSystem::OperationNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);


/*
Suspend the event flow until a resume is issued.
*/
void suspend_subscription (
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (SuspendSubscription,
    ManagedGenericIRPSystem::OperationNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);

/*
Resumes the event flow if it was suspended.
*/
void resume_subscription (
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (ResumeSubscription,
    ManagedGenericIRPSystem::OperationNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);


/*
Terminates the subscription with the agent.
*/
void detach (
    in string manager_reference,
    in NotificationIRPConstDefs::SubscriptionId subscription_id
)
raises (DetachException,
    ManagedGenericIRPSystem::ParameterNotSupported,
    ManagedGenericIRPSystem::InvalidParameter);
}
};
#endif

```

