TSGS#28(05)0286

Source: SA5 (Telecom Management)

Title: CR 32303 CM Notification IRP CORBA SS

Document for: Approval

Agenda Item: 7.5.3

Doc-1st-Level	Spec__#_	CR__#	R	Phase	Subject	Cat	Ver-Cur	Doc-2nd-Level	Workitem
SP-050286	32.303	0015	-	Rel-6	Add missing type for support of Structured Event non-filterable Name-Value pairs	F	6.3.0	S5-056364	OAM-NIM

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.303** **CR** **0015** ⌘**rev** **-** ⌘ Current version: **6.3.0** ⌘

*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:	⌘	Add missing type for support of Structured Event non-filterable Name-Value pairs
Source:	⌘	**SA5** (Nortel – Suzèle Lariven – lariven@nortel.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
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:
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:	⌘	Notification non-filterable IS parameters of Alarm IRP, Kernel CM IRP, and Bulk CM IRP are mapped in CORBA Solution Sets to Name-Value pairs of OMG Notification Service Structured Event CORBA field remainder_of_body. Field remainder_of_body is defined by OMG as being of CORBA type any. Additional definition is therefore required to specify how Name-Value pairs are to be implemented in OMG Structured Event CORBA field remainder_of_body.
Summary of change:	⌘	• Definition of type NotificationIRPNotifications::NonFilterableEventBody to which Structured Event CORBA field remainder_of_body is to be mapped. • Alignments with TS 32.150 Style Guide for CORBA SS IDL • Editorial corrections
Consequences if not approved:	⌘	The lack of definition of how Structured Event non-filterable Name-Value pairs are to be implemented would lead to interoperability issues.

Clauses affected:	⌘	4, A.1, A.2, A.3, A.4, A.5			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
		X		O&M Specifications	32.663
Other comments:	⌘	32.663 CR in S5-056367			

Change in Clause 4

4 Architectural features

The overall architectural feature of Notification IRP is specified in 3GPP TS 32.302 [5]. This clause specifies features that are specific to the CORBA Solution Set (SS).

4.1 Notification services

In the CORBA Solution Set, notifications are emitted by IRPAgent using CORBA Notification service (OMG TC Document telecom [2]) and Structured Events.

CORBA Event service (OMG CORBA services [3]) provides event routing and distribution capabilities. CORBA Notification service provides, in addition to Event service, event filtering and support for Quality of Service (QoS) as well.

A subset of CORBA Notification services shall be used to support the implementation of notification. This CORBA Notification service subset, in terms of OMG Notification service (OMG TC Document telecom [2]) defined methods, is identified in the present.

4.1.1 Support of Push and Pull Interface

The IRPAgent shall support the OMG Notification push interface model. Additionally, it may support the OMG Notification pull interface model as well.

4.1.2 Support of multiple notifications in one `push` operation

For efficiency, IRPAgent uses the following OMG Notification Service (OMG TC Document telecom [2]) defined interface to pack multiple notifications and push them to IRPManager using one method `push_structured_events`. The method takes as input a parameter of type `EventBatch` as defined in the OMG `CosNotification` module (OMG TC Document telecom [2]). This data type is a sequence of Structured Events (see clause 4). Upon invocation, this parameter will contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

The maximum number of events that will be transmitted within a single invocation of this operation is controlled by IRPAgent wide configuration parameter. The amount of time IRPAgent will accumulate individual events into the sequence before invoking this operation is controlled by IRPAgent wide configuration parameter as well.

IRPAgent may push `EventBatch` with only one Structured Event.

The OMG Notification service (OMG TC Document telecom [2]) defined IDL module is shown below.

```
module CosNotifyComm {
      …
      Interface SequencePushConsumer : NotifyPublish {
            void push_structured_events(
                  in CosNotification::EventBatch notifications)
            raises( CosEventComm::Disconnected);
                  …
         }; // SequencePushConsumer
…
}; // CosNotifyComm
```

4.1.3 Support of filterable and non-filterable notification parameters

The OMG Notification service defined IDL `CosNotification::StructuredEvent` and `CosNotification::EventBatch` data types are shown below.

```
struct StructuredEvent {
    EventHeader header;
    FilterableEventBody filterable_data;
    any remainder_of_body;
}; // StructuredEvent

typedef sequence<StructuredEvent> EventBatch;
```

Notification IS parameters are mapped:

- either to the Structured Event header, i.e. above IDL `StructuredEvent` data structure field `header`;
- or to the Structured Event body, and in this case:
  - when defined in the IS as filterable, to the Structured Event filterable body fields, i.e. above IDL `StructuredEvent` data structure field `filterable_data`;
  - when defined in the IS as non-filterable, to the Structured Event remaining body, i.e. above IDL `StructuredEvent` data structure field `remainder_of_body`.

The OMG Notification service defined IDL `CosNotification::FilterableEventBody` data type and its~~other~~ supporting types are shown below.

```
struct Property {
    PropertyName name;
    PropertyValue value;
};

typedef sequence<Property> PropertySeq;

typedef PropertySeq FilterableEventBody;
```

In order to ensure uniform implementation for notification IS parameters mapped to Structured Event Name-Value pairs whether defined in the IS as filterable or as non-filterable, IDL `StructuredEvent` data structure field `remainder_of_body` of type `any` shall be mapped to the IDL data structure `NotificationIRPNotifications::NonFilterableEventBody` defined in annex clause A.5:

```
struct NonFilterableEventBody {
    CosNotification::PropertySeq name_value_pairs;
    any remainder_of_non_filterable_body;
};
```

End of Change in Clause 4

Change in Annex Clause A.1

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

```
//File: ManagedGenericIRPConstDefs.idl

#ifndef _MANAGED_GENERIC_IRP_CONST_DEFS_IDL_
#define _MANAGED_GENERIC_IRP_CONST_DEFS_IDL_

#include "<TimeBase.idl">

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

```
/* ## Module: ManagedGenericIRPConstDefs
This module contains definitions commonly used among all IRPs such as Alarm IRP.
================================================================
*/
module ManagedGenericIRPConstDefs
{
   /*
   Definition imported from CosTime.
   The time refers to time in Greenwich Time Zone.
   It also consists of a time displacement factor in the form of minutes of
   displacement from the Greenwich Meridian.
   */
   typedef TimeBase::UtcT IRPTime;

   enum Signal {OK, FAILURE, PARTIALFAILURE};

   /*
   The VersionNumber is a string that identifies the IRP specification name
   and its version number. –See definition "IRP document version number
   string" or "IRPVersion".

   The VersionNumberSet is a sequence of such VersionNumber. –It is returned
   by get_XXX_IRP_versions(). –The sequence order has no significance.
   */
   typedef string VersionNumber;
   typedef sequence <VersionNumber> VersionNumberSet;

   typedef string MethodName;
   typedef string ParameterName;
   typedef sequence <ParameterName> ParameterList;

   /*
   The Method defines the structure to be returned as part of
   get_supported_operations_profile(). –The name shall be the actual method
   name (ex. "attach_push", "change_subscription_filter", etc.)
   The parameter_list contains a list of strings. –Each string shall be
   the actual parameter name (ex. "manager_reference", "filter", etc.)
   */
   struct Method
   {
      MethodName name;
      ParameterList parameter_list;
   };

   /*
   List of all methods and their associated parameters.
   */
   typedef sequence <Method> MethodList;

   /*
   StringTypeOpt is a type carrying an optional parameter.
   If the boolean is TRUE, then the value is present.
   Otherwise the value is absent.
   */
   union StringTypeOpt switch (boolean)
   {
      case TRUE: string value;
   };

   /*
   ShortTypeOpt is a type carrying an optional parameter.
   If the boolean is TRUE, then the value is present.
   Otherwise the value is absent.
   */
   union ShortTypeOpt switch (boolean)
   {
      case TRUE: short value;
   };

   /*
   UnsignedShortTypeOpt is a type carrying an optional parameter.
   If the boolean is TRUE, then the value is present.
   Otherwise the value is absent.
   */
   union UnsignedShortTypeOpt switch (boolean)
   {
      case TRUE: unsigned short value;
```

```
    };

    /*
    LongTypeOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present.
    Otherwise the value is absent.
    */
    union LongTypeOpt switch (boolean)
    {
       case TRUE: long value;
    };

    /*
    UnsignedLongTypeOpt is a type carrying an optional parameter.
    If the boolean is TRUE, then the value is present.
    Otherwise the value is absent.
    */
    union UnsignedLongTypeOpt switch (boolean)
    {
       case TRUE: unsigned long value;
    };

};

#endif --// _ManagedGenericIRPConstDefs_idl_ _MANAGED_GENERIC_IRP_CONST_DEFS_IDL_
```

End of Change in Annex Clause A.1

Change in Annex Clause A.2

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

```
//File: ManagedGenericIRPSystem.idl

#ifndef _ManagedGenericIRPSystem_idl_ _MANAGED_GENERIC_IRP_SYSTEM_IDL_
#define _ManagedGenericIRPSystem_idl_ _MANAGED_GENERIC_IRP_SYSTEM_IDL_

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

/* ## Module: ManagedGenericIRPSystem
This module contains definitions commonly used among all IRPs such as Alarm IRP.
================================================================
*/
module ManagedGenericIRPSystem
{
    /*
    Exception thrown when an unsupported optional parameter
    is passed with information.
    The parameter shall be the actual unsupported parameter name.
    */
    exception ParameterNotSupported { string parameter; };

    /*
    Exception thrown when an invalid parameter value is passed.
    The parameter shall be the actual parameter name.
    */
    exception InvalidParameter { string parameter; };

    /*
    Exception thrown when a valid but unsupported parameter value is passed.
    The parameter shall be the actual parameter name.
    */
    exception ValueNotSupported { string parameter; };

    /*
    Exception thrown when an unsupported optional method is called.
    */
    exception OperationNotSupported {};
```

```
};

#endif --// _ManagedGenericIRPSystem_idl_ _MANAGED_GENERIC_IRP_SYSTEM_IDL_
```

End of Change in Annex Clause A.2

Change in Annex Clause A.3

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

```
//File: NotificationIRPConstDefs.idl

#ifndef _NOTIFICATION_IRP_CONST_DEFS_IDL_
#define _NOTIFICATION_IRP_CONST_DEFS_IDL_

#include "<ManagedGenericIRPConstDefs.idl">

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


/* ## Module: NotificationIRPConstDefs
This module contains definitions specific for Notification IRP.
================================================================
*/
module NotificationIRPConstDefs
{

   /*
   Define the parameters (in the notification header) specified in
   the Notification IRP: IS.
   */
   interface AttributeNameValue
   {
      const string NOTIFICATION_ID = "a";
      const string EVENT_TIME = "b";
      const string SYSTEM_DN = "c";
      const string MANAGED_OBJECT_CLASS = "d";
      const string MANAGED_OBJECT_INSTANCE = "e";
   };

   /*
   It defines the notification categories.
   A notification category is identified by the IRP name and its version number.
   */
   typedef ManagedGenericIRPConstDefs::VersionNumberSet NotificationCategorySet;

   /*
   It defines the notification types of a particular notification category.
   */
   typedef sequence <string> NotificationTypePerNotificationCategory;

   /*
   This sequence identifies all notification types of all notification
   categories identified by NotificationCategorySet. --The number of elements
   in this sequence shall be identical to that of NotificationCategorySet.
   */
   typedef sequence <NotificationTypePerNotificationCategory>
       NotificationTypesSet;

   /*
   It defines a sequence of SubscriptionIds.
   */
   typedef string SubscriptionId;
   typedef sequence <SubscriptionId> SubscriptionIdSet;
```

```
    /*
    This indicates if the subscription is Active (not suspended), Suspended,
    or Invalid.
    */
    enum SubscriptionState {ACTIVE, SUSPENDED, INVALID};

};

#endif --// _NotificationIRPConstDefs_idl_ _NOTIFICATION_IRP_CONST_DEFS_IDL_
```

End of Change in Annex Clause A.3

Change in Annex Clause A.4

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

```
//File: NotificationIRPSystem.idl

#ifndef _NOTIFICATION_IRP_SYSTEM_IDL_
#define _NOTIFICATION_IRP_SYSTEM_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 GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
    exception AtLeastOneNotificationCategoryNotSupported {};

    interface NotificationIRP
    {
        /*
        Return the list of all supported Notification IRP versions
        Each IRPVersion is defined by the rule in TS 32.311 clause titled
        "IRP document version number string"
        */
        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);

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

};

#endif -// -NotificationIRPSystem_idl- _NOTIFICATION_IRP_SYSTEM_IDL_
```

End of Change in Annex Clause A.4

Change in Annex Clause A.5

A.5 IDL specification (file name "NotificationIRPNotifications.idl")

```
//File: NotificationIRPNotifications.idl

#ifndef _NOTIFICATION_IRP_NOTIFICATIONS_IDL_
#define _NOTIFICATION_IRP_NOTIFICATIONS_IDL_

#include <CosNotification.idl>
#include <NotificationIRPConstDefs.idl>

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

module NotificationIRPNotifications
{

    interface Notify
    {
        -/**
        Notification IRP IS defines 6 attributes for the notification header.
```

```
        They are: objectClass, objectInstance, notificationId, eventTime,
        systemDN and notificationType.

        The first 2 attributes are mapped into 1 name-value pair. –The name of
        the mapped IDL construct is MANAGED_OBJECT_INSTANCE. –The const
        string of this mapped IDL construct is defined here.

        The notificationId, eventTime and systemDN are respectively mapped
        into 3 name-value pairs. –The const string(s) of these 3 mapped IDL
        constructs are defined here.

        The notificationType is not mapped into any name-value pair
        but is mapped into the type_name position-dependent
        field of the CORBA structured-event. –There is no need for a const string
        definition for it.
        ––*/

        –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;

    };

    /**
    Type to which OMG CosNotification::StructuredEvent remainder_of_body any is to be mapped
    */

    struct NonFilterableEventBody {
        CosNotification::PropertySeq name_value_pairs;
        any remainder_of_non_filterable_body;
    };

};

#endif –// _NOTIFICATION_IRP_NOTIFICATIONS_IDL_
```

End of Change in Annex Clause A.5
End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2004	S_23	SP-040105	--	--	Automatic upgrade to Rel-6 (no CR)	5.2.0	6.0.0
Sep 2004	S_25	SP-040562	012	--	Update 32.303 using IDL Style Guide	6.0.0	6.1.0
Dec 2004	S_26	SP-040793	013	--	Remove filter requirement in IDL comments in the Notification IRP CORBA SS	6.1.0	6.2.0
Mar 2005	S_27	SP-050035	014	--	Generic System Context, update of reference to IS specification	6.2.0	6.3.0