Technical Specification Group Services and System Aspects **TSGS#25(04)0569**
Meeting #25, Palm Springs, CA, USA, 13 - 16 September 2004

Source: **SA5 (Telecom Management)**

Title: **2 Rel-6 CR 32.673/663 for state change events**

Document for: **Approval**

Agenda Item: **7.5.3**

Doc-1st-Level	Doc-2nd-Level	Spec	CR	Rev	Phase	Subject	Cat	Ver-Cur	Wi
SP-040569	S5-046700	32.663	010	--	Rel-6	Add State Management Support to Kernel CM IRP CORBA SS	B	6.1.0	OAM-NIM
SP-040569	S5-046699	32.673	004	--	Rel-6	Provide constant definitions to support state change events	B	6.0.0	OAM-NIM

3GPP TSG-SA5 (Telecom Management) **S5-046700**
Meeting #38bis, Sophia Antipolis, FRANCE, 28 Jun - 2 Jul 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.663 CR 010** ⌘**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 ☒ Core Network ☒

Title:	⌘	Add State Management Support to Kernel CM IRP CORBA SS
Source:	⌘	**SA5** (islip@lucent.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	14/05/2004
Category:	⌘	**B**
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:	⌘	Add State Management Support to Kernel CM IRP CORBA SS
Summary of change:	⌘	Add IS/SS mapping and IDL definitions for State Change Notification
Consequences if not approved:	⌘	No support for State Management related notification

Clauses affected:	⌘	6.2, 6.4, 7, Annex B				
		Y	N			
Other specs affected:	⌘		X	Other core specifications	⌘	
			X	Test specifications		
			X	O&M Specifications		
Other comments:	⌘	Parent CR 32.673 CR in S5-046699				

Change in Chapter 6.2

6.2 Operation and Notification mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) defines semantics of operation and notification visible across the Kernel Configuration Management IRP. The following table in this subclause indicates mapping of these operations and notifications to their equivalents defined in this SS.

Table 6.2.1: Mapping from IS Notification/Operation to SS equivalents

IS Operation/ notification (3GPP TS 32.662 [4])	SS Method	Qualifier
getNRMIRPVersion	get_NRM_IRP_version	M
notifyObjectCreation	See Notification IRP: CORBA SS [9]	O
notifyObjectDeletion	See Notification IRP: CORBA SS [9]	O
notifyAttributeValueChange	See Notification IRP: CORBA SS [9]	O
notifyStateChange	See Notification IRP: CORBA SS [9]	O
getIRPVersion	get_kernel_CM_IRP_versions	M
getOperationProfile	get_kernal_CM_IRP operation_profile	O
getNotificationProfile	get_kernel_CM_IRP_notification_profile	O
notifyCMSynchronizationRecommended	See Notification IRP: CORBA SS [9]	O

End of Change in Chapter 6.2

Change in Chapter 6.4

6.4 Notification attribute mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) identifies and defines the semantics of attributes for `notifyObjectCreation`, `notifyObjectDeletion`, `notifyAttributeValueChange`, `notifyStateChange` and `notifyCMSynchronizationRecommended` for use for its IRP. The following table in this subclause shows the mapping of the IS notifications to SS equivalents.

Table 6.4.1: Mapping from IS notifications to SS equivalents

IS notifications in 3GPP TS 32.662 [4]	SS notifications	Qualifier
NotifyObjectCreation	push_structured_event	O
NotifyObjectDeletion	push_structured_event	O
NotifyAttributeValue~~-~~Change	push_structured_event	O
NotifyStateChange	push_structured_event	O
NotifyCMSynchronizationRecommended	push_structured_event	M

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) also qualifies the attributes. The following tables in this subclause show the mapping of these IS attributes to SS equivalents.

Table 6.4.2: Mapping from IS `Notification Header` attributes to SS equivalent

IS Attribute of `Notification Header` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
managedObjectClass	KernelCmNotifDefs::NotificationCommon::MANAGED_OBJECT_CLASS	M
managedObjectInstance	KernelCmNotifDefs::NotificationCommon::MANAGED_OBJECT_INSTANCE	M
notificationId	KernelCmNotifDefs::NotificationCommon::NOTIFICATION_ID	O
eventTime	KernelCmNotifDefs::NotificationCommon::EVENT_TIME	M
systemDN	KernelCmNotifDefs::NotificationCommon::SYSTEM_DN	O
eventType	header.fixed_header.event_type.type_name	M

Table 6.4.3: Mapping from IS `notifyObjectCreation` attributes to SS equivalent OBJECT_CREATION

IS Attribute of `notifyObjectCreation` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
correlatedNotifications	KernelCmNotifDefs::MOCreation::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifDefs::MOCreation::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifDefs::MOCreation::SOURCE_INDICATOR	O
attributeList	KernelCMNotifDefs::MOCreation::MOAttributeSet (contained in remainder_of_body)	O

Table 6.4.4: Mapping from IS `notifyObjectDeletion` attributes to SS equivalent OBJECT_DELETION

IS Attribute of `notifyObjectDeletion` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
correlatedNotifications	KernelCmNotifDefs::MODeletion::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifDefs::MODeletion::ADDITIONAL_ TEXT	O
sourceIndicator	KernelCmNotifDefs::MODeletion::SOURCE_INDICATOR	O
attributeList	KernelCMNotifDefs::MODeletion::MOAttributeSet (contained in remainder_of_body)	O

Table 6.4.5: Mapping from IS `notifyAttributeValueChange` attributes to SS equivalent ATTRIBUTE_VALUE_CHANGE

IS Attribute of `notifyAttributeValue Change` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
correlatedNotifications	KernelCmNotifDefs::AttributeValueChange::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifDefs::AttributeValueChange::ADDITIONAL_TEXT	M
sourceIndicator	KernelCmNotifDefs::AttributeValueChange::SOURCE_INDICATOR	O
attributeValueChangeDefinition	KernelCMNotifDefs:: AttributeValueChange::MOAttributeSet (contained in remainder_of_body)	M

Table 6.4.6: Mapping from IS `notifyCMSynchronizationRecommended` attributes to SS equivalent REQUEST_CM_SYNCHRONIZATION

IS Attribute of notifyCMSynchronization Recommended in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
baseMOClass	KernelCmNotifDefs::CMSynchronizationRecommended::BASE_MO_CLASS	M
baseMOInstance	KernelCmNotifDefs::CMSynchronizationRecommended::BASE_MO_INSTANCE	M
scope	KernelCmNotifDefs::CMSynchronizationRecommended::SCOPE	M
additionalText	KernelCmNotifDefs::CMSynchronizationRecommended::ADDITIONAL_TEXT	O

Table 6.4.7: Mapping from IS `notifyStateChange` attributes to SS equivalent STATE_CHANGE

IS Attribute of `notifyStateChange` `Change` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
stateDefinition	State Management Attribute value pairs defined in the interface StateManagementIRPConstDefs::AttributeNameValue (See note 1)	M
correlatedNotifications	KernelCmNotifDefs::StateChange::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifDefs::StateChange::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifDefs::StateChange::SOURCE_INDICATOR	O

NOTE: The stateDefinition attribute value pair contains the state identifier in the name and the new and optional old state values in the attribute field (See TS 32.673 StateManagementIRPConstDefs IDL <*State Name*>*OldNewValue* structures)

End of Change in Chapter 6.4

Change in Chapter 7

7 Use of OMG Structured Event

In CORBA SS, OMG defined `StructuredEvent` (see OMG Notification Service [6]) is used to carry notifications. This clause identifies the OMG defined `StructuredEvent` attributes that carry the attributes of notifications defined in 3GPP TS 32.662 [4].

The composition of OMG Structured Event, as defined in OMG Notification Service [6], is:

```
Header
      Fixed Header
           domain_name
           type_name
           event_name
      Variable Header
Body
      filterable_body_fields
      remainder_of_body
```

The following table in this clause lists all OMG Structured Event attributes in its leftmost column. The second column identifies the SS attributes, if any, that shall be carried there.

Attributes that are denoted as "optional" may be absent from the OMG Structured Event. As an example, if the optional `additionalText` attribute is not used for a particular notification, then the IRPAgent may exclude `additionalText` from the filterable body fields for that particular notification. Individual notifications from the same IRPAgent may include or exclude the same optional attribute.

Table 7.1: Use of OMG Structured Event

SS Attribute	OMG CORBA Structured Event attribute	Comment
There is no corresponding SS attribute	domain_name	It contains the supported SS document version (see clause 4). This version is defined by the string constant KernelCmIRPSystem::VERSION defined in this specification.
Event Type	type_name	It is an attribute of notificationHeader. It shall indicate one of the following: Object Creation, Object Deletion, Attribute Value Change, State Change and CM Synchronization Recommended. It is a string. Its value is either defined by KernelCmNotifDefs::MOCreation::EVENT_TYPE, KernelCmNotifDefs::MODeletion::EVENT_TYPE, KernelCmNotifDefs::AttributeValueChange::EVENT_TYPE, KernelCmNotifDefs::StateChange::EVENT_TYPE or KernelCmNotifDefs::CMSynchronizationRecommended::EVENT_TYPE
-	event_name	Shall be set to an empty string
There is no corresponding SS attribute	variable Header	
Managed Object Class, Managed Object Instance	One NV pair of filterable_ body_fields	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. They are attributes of notificationHeader. Name of NV pair is a string, KernelCmNotifDefs::<interface>::MANAGED_OBJECT_INSTANCE where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a string. This string conveys the semantics of both the Managed Object Class and the Managed Object Instance. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [9]).
NotificationId	One NV pair of filterable_ body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifDefs::<interface>::NOTIFICATION_ID where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [9]).

SS Attribute	OMG CORBA Structured Event attribute	Comment
EventTime	One NV pair of filterable_ body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifDefs::<interface>::EVENT_TIME where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a ManagedGenericIRPConstDefs::IRPTime defined in 3GPP TS 32.303 [9]. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [9]).
SystemDN	One NV pair of filterable_ body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifDefs::<interface>::SYSTEM_DN where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS [9].
Correlated Notifications	One NV pair of filterable_ body_fields	It is an attribute of the Object Creation, Object Deletion and Attribute Value Change notifications. Name of NV pair is a string, KernelCmNotifDefs::<interface>::CORRELATED_NOTIFICATIONS where <interface> is either MOCreation, MODeletion, StateChange or AttributeValueChange. Value of NV pair is a NotificationIRPConstDefs::CorrelatedNotificationSetType defined in 3GPP TS 32.303 [9].
Additional Text	One NV pair of filterable_ body_fields	It is an attribute of the Object Creation, Object Deletion, Attribute Value Change and CM Synchronization Recommended notifications. Name of NV pair is a string, KernelCmNotifDefs::<interface>::ADDITIONAL_TEXT where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a string.
Source Indicator	One NV pair of filterable_ body_fields	It is an attribute of the Object Creation, Object Deletion and Attribute Value Change notifications. Name of NV pair is a string, KernelCmNotifDefs::<interface>::SOURCE_INDICATOR where <interface> is either MOCreation, MODeletion, StateChange or AttributeValueChange. Value of NV pair is a string with values of either KernelCmNotifDefs::<interface>::RESOURCE_OPERATION, KernelCmNotifDefs::<interface>::MANAGEMENT_OPERATION or KernelCmNotifDefs::<interface>::UNKNOWN_OPERATION where <interface> is either MODeletion, MOCreation, StateChange or AttributeValueChange.
There is no corresponding SS attribute		Is used to transport attribute information. For Object Creation notification, this is defined by KernelCmNotifDefs::MOCreation::InitialAttributeValues. For Object Deletion notification, this is defined by KernelCmNotifDefs::MODeletion::AttributeValues. For Attribute Value Change notification, this is defined by KernelCmNotifDefs::AttributeValueChange::ModifiedAttributeSet. The name component of InitialAttributeValues, AttributeValues and ModifiedAttributeSet will be set to attribute names defined in KernelCmNRMDefs.
	A set of up to 9 Name-value pairs See table 7.1.1	For state change notifications a series of up to 9 name-value pairs might be sent corresponding with the new and old values of each state/status attribute which has changed it's value. The new values of each state/status attributes that have changed are sent. The IRP agent may optionally send the old state/status changes. The name of the name-value pairs are defined by StateManagementIRPConstDefs::
Base MO Class	One NV pair of filterable_ body_fields	It is an attribute of the CM Synchronization Recommended notifications. Name of NV pair is a string, KernelCmNotifDefs::CMSynchronizationRecommended::BASE_MO_CLASS. Value of NV pair is a string. This string conveys the semantics of the Managed Object Class.
Base MO Instance	One NV pair of filterable_ body_fields	It is an attribute of the CM Synchronization Recommendednotifications. Name of NV pair is a string, KernelCmNotifDefs::CMSynchronizationRecommended::BASE_MO_INSTANCE. Value of NV pair is a string. This is the DN string of the Managed Object Instance.
Scope	One NV pair of filterable_ body_fields	It is an attribute of the CM Synchronization Recommendednotifications. Name of NV pair is a string, KernelCmNotifDefs::CMSynchronizationRecommended::SCOPE.

SS Attribute	OMG CORBA Structured Event attribute	Comment
		Value of NV pair is KernelCmNotifDefs::ScopePara.

Table 7.1.1 Name ñ value pairs for state change notifications

Name	Value
OPERATIONAL_STATE	StateManagementIRPConstDefs::OperationalStateOldNewValue
USAGE_STATE	StateManagementIRPConstDefs::UsageStateOldNewValue
ADMINISTRATIVE_STATE	StateManagementIRPConstDefs::AdministrativeStateOldNewValue
ALARM_STATUS	StateManagementIRPConstDefs::AlarmStatusOldNewValue
PROCEDURAL_STATUS	StateManagementIRPConstDefs::ProceduralStatusOldNewValue
AVAILABILITY_STATUS	StateManagementIRPConstDefs::AvailabilityStatusOldNewValue
CONTROL_STATUS	StateManagementIRPConstDefs::ControlStatusOldNewValue
STANDBY_STATUS	StateManagementIRPConstDefs::StandbyStatusOldNewValue
UNKNOWN_STATUS	StateManagementIRPConstDefs::UnknownStatusOldNewValue

End of Change in Chapter 7

Change in Annex B

Annex B (normative): CORBA IDL, Notification Definitions

```
#ifndef KernelCmNotifDefs_idl
#define KernelCmNotifDefs_idl

#include <TimeBase.idl>          // CORBA Time Service
#include <NotificationIRPConstDefs.idl>
#include <StateManagementIRPConstDefs.idl>

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

module KernelCmNotifDefs
{


      /**
       * Definition of ITU-T defined semantics.
       * These constants are used in the type_name
       * (header.fixed_header.event_type.type_name)
       * field to denote the notification type
       * Note all values are unique among themselves.  Other IRP documents
       * cannot use the same values.
       */
      const string ET_OBJECT_CREATION = "x6";

      const string ET_OBJECT_DELETION = "x7";

     const string ET_ATTRIBUTE_VALUE_CHANGE = "x8";
     const string ET_CM_SYNCHRONIZATION_RECOMMENDED = "x9";
     const string ET_STATE_CHANGE = "xA";



      /**
       * Information about one attribute
       * - name defines the name of the attribute
       * - value defines the value of the attribute
       *
       */
      struct MOAttribute
```

```
   {
      string name;
      any value;
   };


   /**
    * A set of attribute names and values
    */
   typedef sequence<MOAttribute> MOAttributeSet;


 /**
  * ScopeType defines the kind of scope to use in a CM synchronization
  * request together with ScopePara.level, in the SCOPE field.
  *
  * ScopePara.level is always >= 0. If a level is bigger than the
  * depth of the tree there will be no exceptions thrown.
  * BASE_ONLY: level ignored, just return the base object.
  * BASE_NTH_LEVEL: return all subordinate objects that are on "level"
  * distance from the base object, where 0 is the base object.
  * BASE_SUBTREE: return the base object and all of its subordinates
  * down to and including the nth level.
  * BASE_ALL: level ignored, return the base object and all of it's
  * subordinates.
  */
 enum ScopeType
 {
    BASE_ONLY,
    BASE_NTH_LEVEL,
    BASE_SUBTREE,
    BASE_ALL
 };

 struct ScopePara
 {
    ScopeType type;
    unsigned long level;
 };
typedef sequence <long> NotifIdSetType;

/*
This holds identifiers of notifications that are correlated.
*/
struct CorrelatedNotification
{
   DN source;  // Contains DN of MO that emitted the set of notifications
               // DN string format in compliance with Name Convention for
               // Managed Object.
               // This may be a zero-length string.  In this case, the MO
               // is identified by the value of the MOI attribute
               // of the Structured Event, i.e., the notification.
   NotifIdSetType notif_id_set;  // Set of related notification ids
};

/*
Correlated Notification sets are sets of Correlated Notification
structures.
*/
typedef sequence <CorrelatedNotification> CorrelatedNotificationSetType;

   /**
    *  This interface defines fields that are common for all
    *  notification types.
    *  All constants in the scope of this interface will be
    *  visible in the interfaces that inherits this.
    *  For instance constant
    *  NotificationCommon::MANAGED_OBJECT_CLASS
    *  can be addressed by MODeletion::MANAGED_OBJECT_CLASS
    */
   /*
   This block identifies attributes which are included as part of the Kernel
   CM IRP. These attribute values should not clash with those defined for the
   attributes of notification header (see IDL of Notification IRP).
   */
   interface AttributeNameValue
   {
      const string SOURCE_INDICATOR = "SOURCE";
```

```
         const string ADDITIONAL_TEXT = "ADD_TEXT";
         const string CORRELATED_NOTIFICATIONS = "CORREL_NOTIFS";
        const string BASE_MO_CLASS = "BASE_MOC";
        const string BASE_MO_INSTANCE = "BASE_MOI";
        const string SCOPE = "SCOPE";
      };

      interface NotificationCommon
      {

         /**
          * This constant defines a field in the filterable
          * information in a StructuredEvent.
          * This string is mapped to the name part of a
          * Property in the event and the value part will
          * carry the MO class name represented
          * as a string.
          */
         const string MANAGED_OBJECT_CLASS =
           NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;



         /**
          * This constant defines a field in the filterable
          * information in a StructuredEvent.
          * This string is mapped to the name part of a
          * Property in the event and the value part will
          * carry the MO distinguished name represented
          * as a string.
          */
         const string MANAGED_OBJECT_INSTANCE =
NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;



         /**
          *  This constant defines the name of the notification
          *  ID property, which is transported in the
          *  filterable_body_fields
          */
         const string NOTIFICATION_ID =
           NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;



         /**
          *  This constant defines the name of the
          *  event time property, which is transported in the
          *  filterable_body_fields.
          *  The data type for the value of this property
          *  is defined by datatype CommonIRPConstDefs::IRPTime
          */
         const string EVENT_TIME =
           NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;




         /**
          *  This constant defines the name of the
          *  system name property, which is transported in the
          *  filterable_body_fields
          */
         const string SYSTEM_DN =
           NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;



         /**
          *  This constant defines the name of the
          *  source indicator  property, which is transported in the
          *  filterable_body_fields
          */
         const string SOURCE_INDICATOR =
           KernelCmNotifDefs::AttributeNameValue::SOURCE_INDICATOR;

         /**
```

```
    * Valid values for the SOURCE_INDICATOR
    * property
    */
   const string RESOURCE_OPERATION =  "RESOURCE OPERATION";
   const string MANAGEMENT_OPERATION = "MANAGEMENT OPERATION";
   const string UNKNOWN_OPERATION = "UNKNOWN";


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




   /**
    *  This constant defines the name of the
    *  correlated notifications property,
    *  which is transported in the
    *  filterable_body_fields
    *  The value part of the property is
    *    KernelCmNotifDefs::CorrelatedNotificationSetType
    */
   const string CORRELATED_NOTIFICATIONS =
     KernelCmNotifDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;


};


/**
 *  Constant definitions for the MO deleted notification
 */
interface MODeletion : NotificationCommon
{

   const string EVENT_TYPE = ET_OBJECT_DELETION;


   /**
    * This information mapped into the remainder_of_body
    * in the StructuredEvent
    */
   typedef MOAttributeSet AttributeValues;

};



/**
 *  Constant definitions for the MO created notification
 */
interface MOCreation : NotificationCommon
{
   const string EVENT_TYPE = ET_OBJECT_CREATION;


   /**
    * This information mapped into the remainder_of_body
    * in the StructuredEvent
    */
   typedef MOAttributeSet InitialAttributeValues;

};



/**
 *  Constant definitions for the Attribute Value Change
 *  notification
 */
```

```
  interface AttributeValueChange : NotificationCommon
  {

     const string EVENT_TYPE = ET_ATTRIBUTE_VALUE_CHANGE;

     /**
      * Information about modidified attributes for
      * one MO instance.
      * - name defines the name of the attribute
      * - newValue defines the new value of the attribute
      * - oldValue defines the previous value of the attribute
      *    The value is optional, which means that it may contain
      *    an empty any (null inserted in the any).
      *
      */
     struct ModifiedAttribute
     {
        string name;
        any newValue;
        any oldValue;
     };


     /**
      * This information mapped into the remainder_of_body
      * in the StructuredEvent.
      */
     typedef sequence<ModifiedAttribute> ModifiedAttributeSet;


   };

  /**
   *  Constant definitions for the CM Synchronization Recommended notification
   */
  interface CMSynchronizationRecommended
  {
     const string EVENT_TYPE = ET_CM_SYNCHRONIZATION_RECOMMENDED;


    /**
      * This constant defines a field in the filterable
      * information in a StructuredEvent.
      * This string is mapped to the name part of a
      * Property in the event and the value part will
      * carry the MO class name represented
      * as a string.
      */
     const string MANAGED_OBJECT_CLASS =
       NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;



     /**
      * This constant defines a field in the filterable
      * information in a StructuredEvent.
      * This string is mapped to the name part of a
      * Property in the event and the value part will
      * carry the MO distinguished name represented
      * as a string.
      */
     const string MANAGED_OBJECT_INSTANCE =
NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;



     /**
      *  This constant defines the name of the notification
      *  ID property, which is transported in the
      *  filterable_body_fields
      */
     const string NOTIFICATION_ID =
       NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;



     /**
      *  This constant defines the name of the
```

```
          *  event time property, which is transported in the
          *  filterable_body_fields.
          *  The data type for the value of this property
          *  is defined by datatype CommonIRPConstDefs::IRPTime
          */
         const string EVENT_TIME =
           NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;

         /**
          *  This constant defines the name of the
          *  system name property, which is transported in the
          *  filterable_body_fields
          */
         const string SYSTEM_DN =
           NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;


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

          *  This constant defines the name of the
          *  base MO class property,
          *  which is transported in the filterable_body
          *  fields.
          *  The value part of this property will carry
          *  the base MO class name as a string.
          */
         const string BASE_MO_CLASS =
           KernelCmNotifDefs::AttributeNameValue::BASE_MO_CLASS;

         /**
          *  This constant defines the name of the
          *  base MO instance property,
          *  which is transported in the filterable_body
          *  fields.
          *  The value part of this property will carry
          *  the base MO distinguished name as a string.
          */
         const string BASE_MO_INSTANCE =
           KernelCmNotifDefs::AttributeNameValue::BASE_MO_INSTANCE;


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

      };

      /**
       *  Constant definitions for the State Change notification
       */
      interface StateChange : NotificationCommon
      {

         const string EVENT_TYPE = ET_STATE_CHANGE;

         const string SOURCE_INDICATOR          =
            KernelCmNotifDefs::AttributeNameValue::SOURCE_INDICATOR;
         const string ADDITIONAL_TEXT           =
            KernelCmNotifDefs::AttributeNameValue::ADDITIONAL_TEXT;
         const string CORRELATED_NOTIFICATIONS =
            KernelCmNotifDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
      };
```

```
};

#endif
```

Annex C (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Sep 2002	S_17	SP-020466	--	--	Submitted to TSG SA #17 for Approval	1.0.0	5.0.0
Mar 2003	S_19	SP-030143	001	--	CORBA IDL Compiler Errors	5.0.0	5.1.0
Mar 2003	S_19	SP-030145	002	--	Add IDL definition of notifyCMSynchronizationRecommended notification for KernelCM IRP	5.1.0	6.0.0
Jun 2004	S_24	SP-040261	004	--	Add Missing CorrelatedNotificationSetType definition	6.0.0	6.1.0

3GPP TSG-SA5 (Telecom Management) **S5-046699**
Meeting #38bis, Sophia Antipolis, FRANCE, 28 Jun - 2 Jul 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.673 CR 004** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

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: ⌘ Provide constant definitions to support state change events

Source: ⌘ **SA5** (islip@lucent.com)

Work item code: ⌘ OAM-NIM **Date:** ⌘ 02/07/2004

Category: ⌘ **B** ***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: ⌘ Provide constant definitions to support state change events

Summary of change: ⌘ Add constant definitions in annex A1.
These constants are for use as the "name" part of the name-value pairs used in when populating the filterable body of structured event.
The constants are defined to support both the New state value, and the old state value.
Note that old state values according to the ITU-T X.721 and X.731 recommendations are optional.

Consequences if not approved: ⌘ In ability to transfer state change notifications in release 6.

Clauses affected: ⌘ Annex A.1

Other specs affected: ⌘

Y	N		
	X	Other core specifications	⌘
	X	Test specifications	
X		O&M Specifications	Rel-6 32.663

Other comments: ⌘ Parent CR to 32.663 CR in S5-046700

How to create CRs using this form:

Annex A (normative): IDL specifications

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

```
#ifndef StateManagementIRPConstDefs_idl
#define StateManagementIRPConstDefs_idl

#include "CosNotification.idl"
#include "ManagedGenericIRPConstDefs.idl"
#include "StateManagementIRPOptConstDefs.idl"

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

/* ## Module: StateManagementIRPConstDefs
This module contains commonly used definitions for State Management IRP
=========================================================================
*/
module StateManagementIRPConstDefs
{
/*
  Constant definitions for state management notifications uses when populating the
  Cos::Structured event.
  The "name" party of the structured event carries the following constant definitions
  appropriate to the state being notified.
  Refer to TS 32.663 regarding how to populate the structured event
*/
   interface AttributeNameValue {
      const string OPERATIONAL_STATE    = "operationalState";
      const string USAGE_STATE          = "usageState";
      const string ADMINISTRATIVE_STATE = "administrativeState";
      const string ALARM_STATUS         = "alarmStatus";
      const string PROCEDURAL_STATUS    = "proceduralStatus";
      const string AVAILABILITY_STATUS  = "availabilityStatus";
      const string CONTROL_STATUS       = "controlStatus";
      const string STANDBY_STATUS       = "standbyStatus";
      const string UNKNOWN_STATUS       = "unknownStatus";
   };


/*
   The following structures provide the new state value,
   and the optional old state value
   The structures are passed in the value part of the cos structured event
*/
   Struct OperationalStateOldNewValue{
      OperationalState new;
      StateManagementIRPOptConstDefs::OperationalStateTypeOpt old;
   };


   Struct UsageStateOldNewValue{
      UsageState new;
      StateManagementIRPOptConstDefs:: UsageStateTypeOpt old;
   };


   Struct AdministrativeStateOldNewValue{
      AdministrativeState new;
      StateManagementIRPOptConstDefs:: AdministrativeStateTypeOpt old;
   };


   Struct AlarmStatusOldNewValue{
      AlarmStatusValues new;
      StateManagementIRPOptConstDefs:: AdministrativeStateTypeOpt old;
   };
   Struct ProceduralStatusOldNewValue{
      ProceduralStatusValues new;
      StateManagementIRPOptConstDefs:: ProceduralStatusTypeOpt old;
   };

   Struct AvailabilityStatusOldNewValue{
      AvailabilityStatusValues new;
      StateManagementIRPOptConstDefs:: AvailabilityStatusTypeOpt old;
   };

   Struct ControlStatusOldNewValue{
      ControlStatusValues new;
      StateManagementIRPOptConstDefs:: ControlStatusTypeOpt old;
   };

   Struct StandbyStatusOldNewValue{
      StandbyStatus new;
      StateManagementIRPOptConstDefs:: StandbyStatusTypeOpt old;
   };

   Struct UnknownStatusOldNewValue{
      UnknownStatus new;
      StateManagementIRPOptConstDefs:: UnknownStatusTypeOpt old;
   };

   /*
   Definition of Operational State based on X.721 [3], if mandatory.
   */
   enum OperationalState
   {
      Disabled, Enabled
   };

   /*
   Definition of Usage State based on X.721 [3], if mandatory.
   */
   enum UsageState
   {
      Idle, Active, Busy
   };

   /*
   Definition of Administrative State based on X.721 [3], if mandatory.
   */
   enum AdministrativeState
   {
      Locked, Unlocked, ShuttingDown
   };

   /*
   Definition of Alarm Status based on X.721 [3], if mandatory.
   */
   enum AlarmStatusValues
   {
      UnderRepair, Critical, Major, Minor, AlarmOutstanding
   };
   typedef sequence <AlarmStatusValues,5> AlarmStatus;

   /*
   Definition of Procedural Status based on X.721 [3], if mandatory.
   */
   enum ProceduralStatusValues
   {
      InitializationRequired, NotInitialized, Initializing, Reporting,
      Terminating
   };
   typedef sequence <ProceduralStatusValues,5> ProceduralStatus;

   /*
   Definition of Availability Status based on X.721 [3], if mandatory.
   */
   enum AvailabilityStatusValues
   {
      InTest, Failed, PowerOff, OffLine, OffDuty, Dependency, Degraded,
      NotInstalled, LogFull
   };
   typedef sequence <AvailabilityStatusValues,9> AvailabilityStatus;

   /*
   Definition of Control Status based on X.721 [3], if mandatory.
   */
   enum ControlStatusValues
   {
      SubjectToTest, PartOfServicesLocked, ReservedForTest, Suspended
   };
   typedef sequence <ControlStatusValues,4> ControlStatus;

   /*
   Definition of Standby Status based on X.721 [3], if mandatory.
   */
   enum StandbyStatus
   {
      HotStandby, ColdStandby, ProvidingService
   };

   /*
   Definition of Unknown Status based on X.721 [3], if mandatory
   (if switch is TRUE then value equal to TRUE implies "unknown status").
   */
   union UnknownStatus switch(boolean)
   {
      case TRUE: boolean value;
   };

};
#endif
```

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Sep 2002	S_17	SP-020470	--	--	Submitted to TSG SA #17 for Approval	1.0.0	5.0.0
Mar 2003	S_19	SP-030143	001	--	CORBA IDL Compiler Errors, Invalid CORBA IDL Include Reference	5.0.0	5.1.0
Mar 2004	S_23	SP-040105	--	--	Automatic upgrade to Rel-6 (no CR)	5.1.0	6.0.0