TSGS#26(04)0812

Source: SA5 (Telecom Management)

Title: 2 Rel-6 CR 32.661/3 Kernel CM; Requirements / CORBA SS

Document for: Approval

Agenda Item: 7.5.3

Doc1stevel	Specific a	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	WorkitemsI D
SP-040812	32.661	004	--	Rel-6	Add Signalling Transport Network (STN) NRM IRP in KernelCM IRP Requirements	B	6.1.0	S5-049052	OAM-NIM
SP-040812	32.663	011	--	Rel-6	Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields) - Align with IS in 32.662	F	6.2.0	S5-047117	OAM-NIM

CR-Form-v7

CHANGE REQUEST

⌘ 32.661 CR 004 ⌘rev - ⌘ Current version: 6.1.0 ⌘

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

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

Title:	⌘	Add Signalling Transport Network (STN) NRM IRP in KernelCM IRP Requirements
Source:	⌘	**SA5** (llrui@bupt.edu.cn; liyewen@chinamobile.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
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:	⌘	Since Signalling Transport Network (STN) NRM IRP is introduced in R6, the scope of NRM IRP which can use KernelCMIRP needs to be extended.
Summary of change:	⌘	KernelCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP.
Consequences if not approved:	⌘	

Clauses affected:	⌘	2, 4.1

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

Change in Clause 2

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document.*

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".

[4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[5] 3GPP TS 32.632: "Telecommunication management; Configuration Management (CM); Core Network Resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[6] 3GPP TS 32.642: "Telecommunication management; Configuration Management (CM): UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[7] 3GPP TS 32.652: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[8] 3GPP TS 32.662: "Telecommunication management; Configuration Management (CM); Kernel CM Information Service (IS)".

[9] 3GPP TS 32.742: "Telecommunication management; Configuration Management (CM); Signalling Transport Network (STN) Interface Network Resource Model (NRM) Integration Reference Point (IRP): Information Service (IS)".

End of Change in Clause 2

Change in Clause 4.1

4.1 General Requirements

This requirements specification defines requirements for the IS for this IRP. As such, capabilities specified here as being required in the IS are not necessarily required in the product implementation. That which is required in the product implementation will be specified in the IS itself.

The following general and high-level requirements apply for the present IRP:

A. IRP-related requirements in 3GPP TS 32.101 [1].

B. IRP-related requirements in 3GPP TS 32.102 [2].

C. IRP-related requirements in 3GPP TS 32.600 [3].

In addition to the above, the following more specific requirements apply:

1. The IS defined by this IRP shall enable an NM to operate on (access) any of the NRMs defined in [4], [5], [6], ~~and~~ [7] and [8].

2. The IS defined by this IRP shall as far as possible be independent of any specific definitions of MOCs, attributes etc. in the NRMs referred to in item 1.

End of Change in Clause 4.1
End of the Document

CR-Form-v7

CHANGE REQUEST

⌘ 32.663 CR 011 ⌘rev - ⌘ Current version: 6.2.0 ⌘

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

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

Title:	⌘	Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields) - Align with IS in 32.662
Source:	⌘	**SA5** (China Mobile llrui@bupt.edu.cn; Ericsson john.power@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)

Reason for change:	⌘	The IS-defined non-filterable parameters are not mapped into remaining_body of CORBA structured event. Generic IRP reference is missing. IRPVersion definition is incorrect. R99 text remains. Redundant text remains. Align IDL style with IDL Style Guide.
Summary of change:	⌘	Place IS-defined non-filterable parameters into remaining_body of CORBA structured event. Add Generic IRP reference and definition of IRPVersion. Remove R99 text. Remove redundant text, correct miscellaneous errors. Align IDL style with IDL Style Guide.
Consequences if not approved:	⌘	SS would not be aligned with IS. IRPAgent wastes CPU cycles on non-filterable parameters before emission of notification. Specification is ambiguous.

Clauses affected:	⌘	2, 3.1, 4, 5.2, 6.2, 6.3, 6.4, 7, 8.1, Annex A, Annex B, Annex C

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

Change in Clause 2

2 References

- …

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".

[4] 3GPP TS 32.662: "Telecommunication management; Configuration Management (CM); Kernel CM; Information Service (IS)".

[5] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[6] Object Management Group 98 (November 1998): "*Notification Service: Joint Revised Submission OMG TC Document telecom/98-11-01*".

[7] OMG CORBA Services (November 1996): "Common Object Services Specification" (clause 4 contains the Event Service specification).

[8] The Common Object Request Broker: Architecture and Specification (for specification of valid version, see [1]).

[9] 3GPP TS 32.303: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[10] 3GPP TS 32.111-3: "Telecommunication management; Fault Management; Part 3: Alarm Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[11] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[12] 3GPP TS 32.673: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[13] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirements

End of Change in Clause 2

Change in Clause 3.1

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.600 [3] and 3GPP TS 32.662 [4] apply.

IRP document version number string (or "IRPVersion"): see 3GPP TS 32.311 [13].

End of Change in Clause 3.1

Change in Clause 4

4 IRP document version number string

~~The IRP document version number (sometimes called "IRPVersion" or "SS version number") string is used to identify this specification. The string is derived using a rule described in 3GPP TS 32.312 [11]. The value of this string is defined by a constant in annex A.~~

~~This string (or sequence of strings, if more than one version is supported) is returned in `getKernelCmIRPVersion` method and is carried in the first field of the notification header of all notifications related to this IRP.~~

Void.

End of Change in Clause 4

Change in Clause 5.2

5.2 Filter language

The filter language used in the SS is the Extended Trader Constraint Language (see OMG Notification Service [6]). IRPAgents may throw a FilterComplexityLimit exception when a given filter is too complex. ~~However, for 3GPP Release 99 an "empty filter" shall be used i.e. a filter that satisfies all MOs of a scoped search (this does not affect the filter for notifications as defined in the Notification IRP - see 3GPP TS 32.303 [9]).~~

End of Change in Clause 5.2

Change in Clause 5.3

5.3 Syntax for Distinguished Names and Versions

The format of a Distinguished Name is defined in 3GPP TS 32.300 [5].

The version of this IRP is represented as a string (see also clause ~~4~~3.1).

End of Change in Clause 5.3

Change in Clause 6.1

6.1 General mappings

~~The IS parameter name managedObjectInstance is mapped into DN.~~

~~Attributes modelling associations as defined in the NRM (here also called "reference attributes") are in this SS mapped to attributes. The names of the reference attributes in the NRM are mapped to the corresponding attribute names in the MOC. When the cardinality for an association is 0..1 or 1..1 the datatype for the reference attribute is defined as an MOReference. The value of an MO reference contains the distinguished name of the associated MO. When the cardinality for an association allows more than one referred MO, the reference attribute will be of type MOReferenceSet, which contains a sequence of MO references.~~

~~If a reference attribute is changed, an AttributeValueChange notification is emitted.~~

Void.

End of Change in Clause 6.1

Change in Clause 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~~kernel_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 Clause 6.2

Change in Clause 6.3

6.3 Operation parameter mapping

The Kernel CM IRP: IS (see 3GPP TS 32.662 [4]) defines semantics of parameters carried in operations across the Kernel Configuration Management IRP. The following tables in this subclause indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

…

Table 6.3.2: Mapping from IS get~~Kernel~~~~Cm~~IRPVersion parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
versionNumberList	return of type ManagedGenericIRPConstDefs::VersionNumberSet	M
status	exception GetKernelCmIRPVersionsException	M

…

End of Change in Clause 6.3

Change in Clause 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
NotifyAttributeValueChange	push_structured_event	O
NotifyStateChange	push_structured_event	O
NotifyCMSynchronizationRecommended	push_structured_event	~~M~~O

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
~~managedO~~objectClass	KernelCmNotifications~~Defs~~::NotificationCommon::MANAGED_OBJECT_CLASS	M
~~managedO~~objectInstance	KernelCmNotifications~~Defs~~::NotificationCommon::MANAGED_OBJECT_INSTANCE	M
notificationId	KernelCmNotifications~~Defs~~::NotificationCommon::NOTIFICATION_ID	~~O~~M
eventTime	KernelCmNotifications~~Defs~~::NotificationCommon::EVENT_TIME	M
systemDN	KernelCmNotifications~~Defs~~::NotificationCommon::SYSTEM_DN	O
~~eventType~~notificationType	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	KernelCmNotifications~~Defs~~::MOCreation::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications~~Defs~~::MOCreation::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifications~~Defs~~::MOCreation::SOURCE_INDICATOR	O
attributeList	KernelCMNotifications~~Defs~~::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	KernelCmNotifications~~Defs~~::MODeletion::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications~~Defs~~::MODeletion::ADDITIONAL_ TEXT	O
sourceIndicator	KernelCmNotifications~~Defs~~::MODeletion::SOURCE_INDICATOR	O
attributeList	KernelCMNotifications~~Defs~~::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 `notifyAttributeValueChange` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
notificationHeader	See table 6.4.2	M
correlatedNotifications	KernelCmNotifications~~Defs~~::AttributeValueChange::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications~~Defs~~::AttributeValueChange::ADDITIONAL_TEXT	~~M~~O
sourceIndicator	KernelCmNotifications~~Defs~~::AttributeValueChange::SOURCE_INDICATOR	O
attributeValueChangeDefinition	KernelCMNotifications~~Defs~~:: 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	KernelCmNotifications~~Defs~~::CMSynchronizationRecommended::BASE_MO_CLASS	M
baseMOInstance	KernelCmNotifications~~Defs~~::CMSynchronizationRecommended::BASE_MO_INSTANCE	M
scope	KernelCmNotifications~~Defs~~::CMSynchronizationRecommended::SCOPE	M
additionalText	KernelCmNotifications~~Defs~~::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~~stateChange	~~State Management Attribute value pairs defined in the interface~~ StateManagementIRPConstDefs::AttributeNameValue (See note 1)	M
correlatedNotifications	KernelCmNotifications~~Defs~~::StateChange::CORRELATED_NOTIFICATIONS	O
additionalText	KernelCmNotifications~~Defs~~::StateChange::ADDITIONAL_TEXT	O
sourceIndicator	KernelCmNotifications~~Defs~~::StateChange::SOURCE_INDICATOR	O

NOTE 1: The ~~stateDefinition~~ stateChange attribute is mapped into name-value pairs that ~~-~~contain~~s~~ the state identifier in the name and the new and optional old state values in the attribute field (See TS 32.673 [12] StateManagementIRPConstDefs IDL *<State Name>OldNewValue* structures).

End of Change in Clause 6.4

Change in Clause 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 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
here is no orresponding SS ttribute	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.
vent 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 KernelCmNotifications~~Defs~~::MOCreation::EVENT_TYPE, KernelCmNotifications~~Defs~~::MODeletion::EVENT_TYPE, KernelCmNotifications~~Defs~~::AttributeValueChange::EVENT_TYPE, KernelCmNotifications~~Defs~~::StateChange::EVENT_TYPE or KernelCmNotifications~~Defs~~::CMSynchronizationRecommended::EVENT_TYPE
	event_name	Shall be set to an empty string
here is no orresponding SS ttribute	variable Header	
anaged Object lass, Managed bject 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, KernelCmNotifications~~Defs~~::<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]).
otificationId	One NV pair of filterable_ body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifications~~Defs~~::<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]).
ventTime	One NV pair of remainder_of_body~~filterable_ body_fields~~	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifications~~Defs~~::<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]).
ystemDN	One NV pair of filterable_ body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifications~~Defs~~::<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].

SS Attribute	OMG CORBA Structured Event attribute	Comment
orrelated otifications	One NV pair of remainder_of_bodyof 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, KernelCmNotifications Defs::<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].
dditional Text	One NV pair of remainder_of_bodyfilterable_ 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, KernelCmNotifications Defs::<interface>::ADDITIONAL_TEXT where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a string.
ource Indicator	One NV pair of remainder_of_bodyfilterable_ body_fields	It is an attribute of the Object Creation, Object Deletion and Attribute Value Change notifications. Name of NV pair is a string, KernelCmNotifications Defs::<interface>::SOURCE_INDICATOR where <interface> is either MOCreation, MODeletion, StateChange or AttributeValueChange. Value of NV pair is a string with values of either KernelCmNotifications Defs::<interface>::RESOURCE_OPERATION, KernelCmNotifications Defs::<interface>::MANAGEMENT_OPERATION or KernelCmNotifications Defs::<interface>::UNKNOWN_OPERATION where <interface> is either MODeletion, MOCreation, StateChange or AttributeValueChange.
here is no orrespnding SS ttribute		Is used to transport attribute information. For Object Creation notification, this is defined by KernelCmNotifications Defs::MOCreation::InitialAttributeValues. For Object Deletion notification, this is defined by KernelCmNotifications Defs::MODeletion::AttributeValues. For Attribute Value Change notification, this is defined by KernelCmNotifications Defs::AttributeValueChange::ModifiedAttributeSet. The name component of InitialAttributeValues, AttributeValues and ModifiedAttributeSet will be set to attribute names defined in KernelCmNRMDefs.
tateManagement RPConstDefs:: ttributeNameValue	A set of up to 9 Name-value pairs See table 7.1.12. All these 9 NV pairs are part of the remainder_of_body	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::AttributeNameValue
ase MO Class	One NV pair of remainder_of_bodyof filterable_body_fields	It is an attribute of the CM-Synchronization-Recommended notifications. Name of NV pair is a string, KernelCmNotifications Defs::CMSynchronizationRecommended::BASE_MO_CLASS. Value of NV pair is a string. This string conveys the semantics of the Managed Object Class.
ase MO Instance	One NV pair of remainder_of_bodyfilterable_ body_fields	It is an attribute of the CM-Synchronization-Recommended_notifications. Name of NV pair is a string, KernelCmNotifications Defs::CMSynchronizationRecommended::BASE_MO_INSTANCE. Value of NV pair is a string. This is the DN string of the Managed Object Instance.
cope	One NV pair of remainder_of_bodyfilterable_ body_fields	It is an attribute of the CM-Synchronization-Recommended_notifications. Name of NV pair is a string, KernelCmNotifications Defs::CMSynchronizationRecommended::SCOPE. Value of NV pair is KernelCmConstNotifDefs::ScopePara.

…

End of Change in Clause 7

Change in Clause 8.1

8.1 Allowed extensions

~~Vendor specific MOCs may be supported. The vendor specific MOCs may support new types of attributes. The 3GPP SA5 specified notifications may be issued referring to the vendor specific MOCs and vendor specific attributes. New MOCs shall be distinguishable from 3GPP SA5 MOCs by name. 3GPP SA5 specified and vendor specific attributes may be used in vendor specific MOCs. Vendor specific attribute names shall be distinguishable from existing attribute names.~~

~~NRM MOCs may be subclassed. Subclassed MOCs shall maintain the specified behaviour of the 3GPP SA5's superior classes. They may add vendor specific behaviour with vendor specific attributes. When subclassing, naming attributes cannot be changed. The subclassed MOC shall support all attributes of its superior class. Vendor specific attributes cannot be added to 3GPP SA5 NRM MOCs without subclassing.~~

~~When subclassing, the 3GPP SA5 specified containment rules and their specified cardinality shall still be followed. As an example, `ManagementNode` (or its subclasses) shall be contained under `SubNetwork` (or its subclasses). Also, in Rel 4, there may only be 0 or 1 `ManagementNode` (or its subclasses) contained under `SubNetwork` (or its subclasses).~~

~~Managed Object Instances may be instantiated as CORBA objects. This requires that the MOCs be represented in IDL. 3GPP SA5's NRM MOCs are not currently specified in IDL, but may be specified in IDL for instantiation or subclassing purposes. However, management information models should not require that IRPManagers access the instantiated managed objects other than through supported methods in the present document.~~

~~Extension rules related to notifications (Notification categories, Event Types, Extended Event Types etc.) are for further study.~~Void.

End of Change in Clause 8.1

Change in Annex A

Annex A (normative): IDL specification (file name "KernelCmConstDefs.idl")

```
// File: KernelCmConstDefs.idl

#ifndef _KERNELCMCONSTDEFS_IDL_
#define _KERNELCMCONSTDEFS_IDL_


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

module KernelCmConstDefs
{

      /**
       * 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;
    };

    /* The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
    "Name Conventions for Managed Objects".
    */
    typedef string DN;

   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 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";
   };

};

#endif _KERNELCMNOTIFDEFS_IDL_
```

End of change in Annex A

Change in Annex B

Annex BA (normative): ~~CORBA IDL, Access Protocol~~IDL specification (file name "KernelCmIRPSystem.idl")

```
// File: KernelCmIRPSystem.idl

#ifndef KernelCmIRPSystem_idl_KERNELCMIRPSYSTEM_IDL_
#define KernelCmIRPSystem_idl_KERNELCMIRPSYSTEM_IDL_

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

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

module KernelCmIRPSystem
{

    exception GetKernelCMIRPNotificationProfileException { string reason; };
    exception GetKernelCMIRPOperationProfileException { string reason; };
    exception GetNRMIRPVersion { string reason; };
    exception GetKernelCMIRPVersionsException { string reason; };

    /**
     * The KernelCmIrpOperations interface.
     * Supports a number of Resource Model versions.
     */
    interface KernelCmIrpOperations
    {
      /*
      Return the list of all supported Kernel CM IRP versions.
      */
      ManagedGenericIRPConstDefs::VersionNumberSet get_kernel_CM_IRP_versions (
      )
      raises (GetKernelCMIRPVersionsException);


     /**
      * Get the version(s) of the interface
      *
      * @raises GetNRMIRPVersion when the system for some reason
      *   can not return the supported versions.
      * @returns all supported versions.
      */
      void get_NRM_IRP_version
      (
         out ManagedGenericIRPConstDefs::VersionNumberSet versionNumberList,
         out ManagedGenericIRPConstDefs::VersionNumberSet vSEVersionNumberList
      )
         raises (GetNRMIRPVersion);

      /*
      Return the list of all supported operations and their supported
      parameters for a specific KernelCM IRP version.
      */
      ManagedGenericIRPConstDefs::MethodList get_kernel_CM_IRP_operation_profile (
         in ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
      )
      raises (GetKernelCMIRPOperationProfileException,
              ManagedGenericIRPSystem::OperationNotSupported,
              ManagedGenericIRPSystem::InvalidParameter);

      /*
      Return the list of all supported notifications and their supported
```

```
        parameters for a specific KernelCM IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
           get_kernel_CM_IRP_notification_profile
        (
           in ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
        )
        raises (GetKernelCMIRPNotificationProfileException,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);

    };
};
#endif _KERNELCMIRPSYSTEM_IDL_
```

End of Change in Annex B

Change in Annex C

Annex CB (normative): ~~CORBA IDL, Notification Definitions~~IDL specification (file name "KernelCmNotifications.idl")

```
// File: KernelCmNotifications.idl

#ifndef KernelCmNotifDefs_idl_KERNELCMNOTIFICATIONS_IDL_
#define KernelCmNotifDefs_idl_KERNELCMNOTIFICATIONS_IDL_

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

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

module KernelCmNotificationsDefs
{
      /**
       * 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;
    };

    /* The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
    "Name Conventions for Managed Objects".
    */
    typedef string DN;

   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: NotificationIRPNotifications::Notify
    {
        /**
         * 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 =
          KernelCmNotifDefsConstDefs::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 =
             KernelCmNotifDefsConstDefs::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
            *    KernelCmConstNotifDefs::CorrelatedNotificationSetType
            */
           const string CORRELATED_NOTIFICATIONS =
             KernelCmNotifDefsConstDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
       };


       /**
        *  Constant definitions for the MO deleted notification
        */
       interface MODeletion : NotificationCommon
       {
           const string EVENT_TYPE = "x7";
           KernelCmNotifDefs ::ET_OBJECT_DELETION;

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



       /**
        *  Constant definitions for the MO created notification
        */
       interface MOCreation : NotificationCommon
       {
           const string EVENT_TYPE = "x6";
           KernelCmNotifDefs::ET_OBJECT_CREATION;

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


       /**
        *  Constant definitions for the Attribute Value Change
        *  notification
        */
       interface AttributeValueChange : NotificationCommon
       {
           const string EVENT_TYPE = "x8";
           KernelCmNotifDefs::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: NotificationIRPNotifications::Notify
      {
        const string EVENT_TYPE = "x9";
          KernelCmNotifDefs::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 =
```

```
            KernelCmNotifDefsConstDefs::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 =
            KernelCmNotifDefsConstDefs::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 =
            KernelCmNotifDefsConstDefs::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 KernelCmConstNotifDefs::ScopePara.
           */
          const string SCOPE =
            KernelCmNotifDefsConstDefs::AttributeNameValue::SCOPE;

      };

      /**
       *  Constant definitions for the State Change notification
       */
      interface StateChange : NotificationCommon
      {
         const string EVENT_TYPE = "xA";
          KernelCmNotifDefs::ET_STATE_CHANGE;
      };


};

#endif _KERNELCMNOTIFDEFS_IDL_
```

End of change in Annex C
End of Document

Annex ~~C~~ D (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
Sep 2004	S_25	SP-040568	007	--	Add missing DN definition	6.1.0	6.2.0
Sep 2004	S_25	SP-040568	009	--	Add missing IDL for get_kernel_CM_IRP_versions	6.1.0	6.2.0
Sep 2004	S_25	SP-040569	010	--	Add State Management Support to Kernel CM IRP CORBA SS	6.1.0	6.2.0