

---

**Source:** SA5 (Telecom Management)  
**Title:** CR 32663 Configuration Management (CM) Kernel CM IRP  
**Document for:** Approval  
**Agenda Item:** 7.5.3

---

Doc-1st-Level	Spec	CR	R	Phase	Subject	Ca	VerCr	Doc-2nd-Level	Workitem
SP-050050	32.663	012	--	Rel-6	IDL in compliant to the style guide	F	6.3.0	S5-056072	OAM-NIM

## CHANGE REQUEST

⌘ **32.663 CR 012** ⌘ 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  Core Network

<b>Title:</b>	⌘ IDL incompliant to the style guide		
<b>Source:</b>	⌘ SA5 (huangsq@zte.com.cn)		
<b>Work item code:</b>	⌘ OAM-NIM	<b>Date:</b>	⌘ 28/01/2005
<b>Category:</b>	⌘ <b>F</b>	<b>Release:</b>	⌘ Rel-6
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		Use <u>one</u> of the following releases: 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) Rel-4 (Release 4) Rel-5 (Release 5) Rel-6 (Release 6) Rel-7 (Release 7)

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

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

## 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__

```

---

## Annex B (normative): IDL specification (file name "KernelCmIRPSystem.idl")

```
// File: KernelCmIRPSystem.idl

#ifndef _KERNELCMIRPSYSTEM_IDL_
#define _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_
```

---

## Annex C (normative): IDL specification (file name "KernelCmNotifications.idl")

```
// File: KernelCmNotifications.idl

#ifndef _KERNELCMNOTIFICATIONS_IDL_
#define _KERNELCMNOTIFICATIONS_IDL_

#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 KernelCmNotifications
{
    /**
     * 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
     */

    interface NotificationCommon: NotificationIRPNotifications::Notify
    {
        /**
         * This constant defines the name of the
         * source indicator property.
         */
        const string SOURCE_INDICATOR =
            KernelCmConstDefs::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.
         * The data type for the value of this property
         * is a string.
         */
        const string ADDITIONAL_TEXT =
            KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

        /**
         * This constant defines the name of the
         * correlated notifications property.
         * The value part of the property is
         * KernelCmConstDefs::CorrelatedNotificationSetType
         */
        const string CORRELATED_NOTIFICATIONS =
            KernelCmConstDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
    };

    /**
     * Constant definitions for the MO deleted notification
     */
}
```

```

interface MODeletion : NotificationCommon
{
    const string EVENT_TYPE = "x7";

    /**
     * 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";

    /**
     * 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";

    /**
     * Information about modified 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";

    /**
     * This constant defines the name of the
     * additional text property.
     * The data type for the value of this property
     * is a string.
     */
    const string ADDITIONAL_TEXT =
        KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;
};

```

```

/**
 * This constant defines the name of the
 * base MO class property.
 * The value part of this property will carry
 * the base MO class name as a string.
 */
const string BASE_MO_CLASS =
    KernelCmConstDefs::AttributeNameValue::BASE_MO_CLASS;

/**
 * This constant defines the name of the
 * base MO instance property.
 * The value part of this property will carry
 * the base MO distinguished name as a string.
 */
const string BASE_MO_INSTANCE =
    KernelCmConstDefs::AttributeNameValue::BASE_MO_INSTANCE;

/**
 * This constant defines the name of the
 * scope property.
 * The data type for the value of this property
 * is KernelCmConstDefs::ScopePara.
 */
const string SCOPE =
    KernelCmConstDefs::AttributeNameValue::SCOPE;

};

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

};

#endif // _KERNELCMNOTIFDEFS_IDL_

```

---

## Annex 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
Dec 2004	S_26	SP-040812	011	--	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	6.2.0	6.3.0