Technical Specification Group Services and System Aspects
Meeting #19, Birmingham, UK, 17-20 March 2003

TSGS#19(03)0144

Source: **SA5 (Telecom Management)**

Title: **2 Rel-5 CRs 32.602 & 32.603 (Basic Configuration Management IRP information service & CORBA SS) " Add post-condition for notifications of each activeCM operation and one exception for createMO"**

Document for: **Approval**

Agenda Item: **7.5.3**

Alignment of CMCC/SA5 CM framework

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Ver-Current	Doc-2nd-Level	Workitem	Remarks
SP-030144	32.602	003	-	Rel-5	**Add post-condition for notifications of each activeCM operation and one exception for createMO**	F	5.0.0	S5-036116	OAM-NIM	**Parent CR.**
SP-030144	32.603	009	-	Rel-5	**Add description for notifications of each activeCM operation and one exception for createMO - alignment with 32.602, Information Service**	F	5.0.0	S5-036117	OAM-NIM	**Child CR.**

3GPP TSG-SA WG 5 Meeting #32bis
Sophia Antipolis, France, 20-24 Jan. 2003

Tdoc S5-036116 ⌘

CR-Form-v7

CHANGE REQUEST

⌘ **32.602** **CR** **003** ⌘**rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 [X] Core Network [X]

Title:	⌘	Add post-condition for notifications of each activeCM operation and one exception for createMO
Source:	⌘	S5
Work item code:	⌘	OAM-NIM
Date:	⌘	28/02/2003
Category:	⌘	**F**
Release:	⌘	*Rel-5*

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:	⌘	Post-condition for notifications of each activeCM operation in BasicCM IRP and one exception for createMO are needed.
Summary of change:	⌘	The notifications to be emitted when each operation has been finished and one exception for createMO are added. Clarifications of some post-conditions and removal of wrong exceptions.
Consequences if not approved:	⌘	Different interpretations of the Basic CM standard can cause interoperability problems.

Clauses affected:	⌘	2, 7.6.1, 7.6.2, 7.6.3

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
	X		O&M Specifications	32.603, 32.604

Other comments:	⌘	This CR is Parent to 32603CR009 S5-036117. Alignment of CMCC/SA5 CM framework

How to create CRs using this form:

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.

[1] 3GPP TS 32.101: "3G Telecom Management principles and high level requirements".

[2] 3GPP TS 32.102: "3G Telecom Management architecture".

[3] 3GPP TS 32.302: "Telecommunication management; Configuration Management; Part 2: Notification Integration Reference Point (IRP); Information service".

[4] 3GPP TS 32.312: "Generic Integration Reference Point (IRP) management; Information service ".

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

[6] 3GPP TS 32.600: "Configuration Management (CM); Concept and main requirements".

[7] ITU-T Recommendation X.710 (1997): "Common Management Information Service

[8] ITU-T Recommendation X.721 (02/92): "Information Technology - Open Systems Interconnection – Structure of Management Information: Definition of Management Information".

[9] ITU-T Recommendation X.730 (01/92): "Information Technology - Open Systems Interconnection – Systems Management: Object Management Function".

[10] ITU-T Recommendation X.733 (02/92): "Information Technology - Open Systems Interconnection - Alarm Reporting Function".

[11] 3GPP TS 32.662: "3G Configuration Management: Kernel CM IRP: Information Service".

...

7.6 Interface ActiveCmIRPOperations

7.6.1 ~~createMo~~ createMO (O)

7.6.1.1 Definition

This operation is invoked by IRPManager to request the IRPAgent to create a Managed Object instance in the MIB maintained by the IRPAgent. This operation will create only one Managed Object instance. This operation provides functionality that is similar to that provided by the M-CREATE service defined by CMIS (ITU-T X.710 [7]).

7.6.1.2 Input Parameters

Name	Qualifier	Information Type	Comment
managedObjectClass	M	ObjectClassIdentifier	This parameter specifies the class of the new managed object instance.
managedObjectInstance	M	DistinguishedName	This parameter specifies the instance of the managed object that is to be created and registered. This is a full Distinguished Name according to 3GPP TS 32.300 [13].
referenceObjectInstance	O	Solution Set dependant	This parameter may have a null value. When this parameter is supplied, it must specify an existing instance of a managed object, called the reference object, of the same class as the new object to be created. Attribute values associated with the reference object instance become the default values for those not specified by the attributeListIn parameter.
attributeListIn	M	LIST OF SEQUENCE< attribute name, attribute value>	This parameter may have a null value. When this parameter is supplied, it contains a list of name/value pairs specifying attribute identifiers and their values to be assigned to the new managed object. These values override the values for the corresponding attributes derived from either the reference object (if the referenceObjectInstance parameter is supplied) or the default value set specified in the definition of the managed object's class.

7.6.1.3 Output Parameters

Name	Qualifier	Matching Information	Comment
attributeListOut	M	LIST OF SEQUENCE< name OF ManagedEntity.anAttribute, value OF ManagedEntity.anAttribute>	This list of name/value pairs contains the attributes of the new managed object and the actual value assigned to each.
status	M	ENUM (OperationSucceeded, OperationFailed)	An operation may fail because of a specified or unspecified reason.

7.6.1.4 Pre-condition

managedEntityDoesNotExist

Assertion Name	Definition
managedEntityDoesNotExist	A ManagedEntity instance with the same Distinguished Name as the object specified for creation does not exist.~~The ManagedEntity instance is not being created with the same Distinguished Name as another already existing Managed Object instance.~~

7.6.1.5 Post-condition

managedEntityCreated AND objectCreationNotificationEmitted~~Sent~~

Assertion Name	Definition
managedEntityCreated	The ManagedEntity instance of the specified object class has been created with the specified Distinquished Name.
objectCreationNotificationEmitted~~Sent~~	An object creation notification (as defined in TS 32.662) is emitted~~generated~~ for the created object, if notifiable. Notifiable here means that the notification is supported and not suppressed. "Emitted" here corresponds to the association stereotyped as <<emits>> in 3GPP TS 32.662 [11].

7.6.1.6 Exceptions

Name	Definition
operationFailed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.
objectClassSpecificationMissmatched	**Condition:** The object class named by ObjectClassIdentifier input parameter does not match the object class of the managed object specified by a non-null referenceObjectInstance input parameter. **Returned Information:** The output parameter status. **Exit state:** Entry state.
InvalidObjectInstance	**Condition:** The object instance name specified implied a violation of the naming rules; **Returned Information:** The output parameter status. **Exit state:** Entry state.
createNotAllowed	**Condition:** The object to be created may not be created over the Itf-N. **Returned Information:** The output parameter status. **Exit state:** Entry state.
noSuchObjectClass	**Condition:** The class of the specified managed object is not recognized ~~or~~. **Returned Information:** The output parameter status. **Exit state:** Entry state.
classInstanceConflict	**Condition:** The specified managed object instance may not be created as member of the specified class. **Returned Information:** The output parameter status. **Exit state:** Entry state.
noSuchAttribute	**Condition:** A specified attribute is not recognized or is not valid for specified object class. **Returned Information:** The output parameter status. **Exit state:** Entry state.
invalidAttributeValue	**Condition:** Value specified for an attribute is not valid for that attribute. **Returned Information:** The output parameter status. **Exit state:** Entry state.
missingAttributeValue	**Condition:** One or more required attribute values were not supplied and default values are not available. **Returned Information:** The output parameter status. **Exit state:** Entry state.
parentObjectDoesNotExist	**Condition:** The parent MO instance of the ManagedEntity specified to be created does not exist. **Returned Information:** The output parameter status. **Exit state:** Entry state.

7.6.2 ~~deleteMo~~ deleteMO (O)

7.6.2.1 Definition

This operation is invoked by IRPManager to request the deletion of one or more Managed Object instances from the MIB maintained by IRPAgent. This operation provides functionality that is similar to that provided by the M-DELETE service defined by CMIS (ITU-T X.710 [7]).

7.6.2.2 Input Parameters

Name	Qualifier	Information Type	Comment
baseObjectInstance	M	DistinguishedName	The MO instance that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied. This is a full Distinguished Name according to 3GPP TS 32.300 [13].
scope	M	See corresponding parameter in getMOAttributes.	See corresponding parameter in getMOAttributes.
filter	M	See comment	See corresponding parameter in getMOAttributes.

7.6.2.3 Output Parameters

Name	Qualifier	Matching Information	Comment
deletionList	M	LIST OF SEQUENCE< ManagedEntity.distinguishedName, ManagedEntity.objectClass>	If the base object alone is specified, then this parameter is optional; otherwise it contains a list of managedObjectInstance/managedObjectClass pairs identifying the managed objects deleted.
status	M	ENUM (OperationSucceeded, OperationFailed, OperationPartiallySucceeded)	An operation may fail because of a specified or unspecified reason. The operation is partially successful if some, but not all, objects selected to be deleted are actually deleted.

In lieu of a synchronization parameter, best effort synchronization will apply; that is, all managed objects selected for this operation will perform the operation if possible regardless of whether some managed objects fail to perform it.

7.6.2.4 Pre-condition

baseObjectExists AND allChildrenOfObjectsToBeDeletedSpecifiedForDeletion

Assertion Name	Definition
baseObjectExists	The ManagedEntity instance specified by the baseObjectInstance parameter exists.
allChildrenOfObjectsToBeDeletedSpecifiedForDeletion	For any ManagedEntity instance specified for deletion, all of its dependant ManagedEntity instances must be specified for deletion.

7.6.2.5 Post-condition

(selectedObjectsDeleted OR someSelectedObjectsDeleted) AND objectDeletionNotificationEmitted~~Sent~~

Assertion Name	Definition
selectedObjectsDeleted	All of the ManagedEntity instances selected for deletion are deleted.
someSelectedObjectsDeleted	Some but not all of the selected ManagedEntity instances were deleted and for any of the ManagedEntity instances deleted all of the child ManagedEntity instances of that ManagedEntity instance is deleted.
objectDeletionNotificationEmitted~~Sent~~	An object deletion notification (as defined in TS 32.662) is emitted~~generated~~ for each notifiable deleted object. Notifiable here means that the notification is supported and not suppressed. “Emitted” here corresponds to the association stereotyped as <<emits>> in 3GPP TS 32.662 [11]. An object deletion notification of a managed object containing a sub-tree implies deletion of all managed objects in the sub-tree. IRPAgent should make the best effort to reduce the number of object deletion notifications, for example by sending only one notification for the sub-tree root object in the event of a successful deletion of an entire sub-tree.

7.6.2.6 Exceptions

Name	Definition
operationFailed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.
invalidObjectInstance	**Condition:** The object instance name specified implied a violation of the naming rules; **Returned Information:** The output parameter status. **Exit state:** Entry state.
deleteNotAllowed	**Condition:** Some of the object instances to be deleted may not be deleted over the Itf-N. **Returned Information:** The output parameter status. **Exit state:** Entry state.
resourceLimitation	**Condition:** Operation not performed due to resource limitation. **Returned Information:** The output parameter status. **Exit state:** Entry state.
complexityLimitation	**Condition:** Operation not performed because a parameter was too complex. **Returned Information:** The output parameter status. **Exit state:** Entry state.

7.6.3 setMOoAttributes (O)

7.6.3.1 Definition

This operation is invoked by IRPManager to request the modification of management information (Managed Object attribute values) in the MIB maintained by IRPAgent. Attributes of one or several Managed Objects may be modified - based on the containment hierarchy. This operation provides functionality that is similar to that provided by the M-SET service defined by CMIS (ITU-T X.710 [7]).

7.6.3.2 Input Parameters

Name	Qualifier	Information Type	Comment
baseObjectInstance	M	DistinguishedName	The MO instance that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied. This is a full Distinguished Name according to 3GPP TS 32.300 [13].
scope	M	See corresponding parameter in getMOAttributes.	See corresponding parameter in getMOAttributes.
filter	M	See comment	See corresponding parameter in getMOAttributes.
modificationList	M	LIST OF SEQUENCE <attribute identifier, [attribute values], ENUM(replace, add values, remove values, set to default)> See Comment for when attribute values are require and when they are optional.	This parameter contains a set of attribute modification specifications, each of which contains: 1. attribute identifier: the identifier of the attribute whose value(s) is(are) to be modified. 2. attribute value: the value(s) to be used in the modification of the attribute. The use of this parameter is defined by the modify operator. This parameter is optional when the set to default modify operator is specified and if supplied, shall be ignored. 3. modify operator: the way in which the attribute values(s) (if supplied) is(are) to be applied to the attribute. The possible operators are: replace: the attribute value(s) specified shall be used to replace the current values(s) of the attribute; add values: the attribute values(s) specified shall be added to the current value(s) of the of the attribute. This operator shall only be applied to a set-valued attribute and shall perform a set union (in the mathematical sense) between the current values(s) of the attribute and the attribute value(s) specified. Value(s) specified in the attribute value parameter which is(are) already in the current values of the attribute shall not cause an error to be returned. remove values: the attribute value(s) specified shall be removed from the current values(s) of the attribute. This operator shall only be applied to a set-valued attribute and shall perform a set difference (in the mathematical sense) between the current value(s) of the attribute and the attribute values(s) specified. Value(s) specified in the attribute value parameter which is(are) not in the current value(s) of the attribute shall not cause an error to be returned; set to default: when this operator is applied to a single-valued attribute, the value of the attribute shall be set to its default value. When this operator is applied to a set–valued attribute, the value(s) of the attribute shall be set to their default value(s) and only as many values as defined by the default shall be assigned. If there is no default value defined, an error shall be returned. Note: Set is used here in the mathematical sense so that a set-valued attribute is an unordered set of unique values. The modify operator is optional, and if it is not specified, the replace operator shall be assumed. The modificationList parameter contains a single set of attribute modification specifications and this same set is applied to each managed object instance to be modified.

7.6.3.3 Output Parameters

Name	Qualifier	Matching Information	Comment
modificationListOut	M	LIST OF SEQUENCE< ManagedEntity.distinguishedName, ManagedEntity.objectClass, LIST OF SEQUENCE<name OF ManagedEntity.anAttribute, value OF ManagedEntity.anAttribute>>	This parameter will provide for each managed object instance the full Distinguished Name of the managed object instance, the managedObjectClass, and a list of name/value pairs with the values of all the attributes of the modified managed object instance after modification. The form of this information is solution set dependant and may be provided in one or many data structures.
status	M	ENUM (OperationSucceeded, OperationFailed, OperationPartiallySucceeded)	An operation may fail because of a specified or unspecified reason and no attributes have been updated. The operation is only successful if all specified attributes of all selected objects are actually modified. Otherwise, the operation is partially successful.

In lieu of a synchronization parameter, best effort synchronization will apply; that is, all managed objects selected for this operation will perform the operation if possible regardless of whether some managed objects fail to perform it.

7.6.3.4 Pre-condition

baseObjectExists

Assertion Name	Definition
baseObjectExists	The ManagedEntity instance specified by the baseObjectInstance parameter exists.

7.6.3.5 Post-condition

(selectedObjectsModified OR someSelectedObjectsModified) AND attributeValueChangeNotificationEmitted~~Sent~~

Assertion Name	Definition
selectedObjectsModified	All of the attributes of all of the ManagedEntity instances selected for modification are modified as specified.
someSelectedObjectsModified	Some attributes of some of the selected ManagedEntity instances were modified but not all attributes of all selected ManagedEntity instances.
attributeValueChangeNotificationEmitted~~Sent~~	A notifyAttributeValueChange notification (as defined in TS 32.662) is ~~generated~~emitted for ~~each~~the notifiable attributes of each modified object instance. ~~.~~ Notifiable here means that the notification is supported and not suppressed. “Emitted” here corresponds to the association stereotyped as <<emits>> in 3GPP TS 32.662 [11].

7.6.3.6 Exceptions

Name	Definition
operationFailed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.
modifyNotAllowed	**Condition:** The object to be modified may not be modified over the Itf-N. **Returned Information:** The output parameter status. **Exit state:** Entry state.
~~noSuchObjectClass~~	~~**Condition:** The class of the specified managed object is not recognized or.~~ ~~**Returned Information:** The output parameter status.~~ ~~**Exit state:** Entry state.~~
~~classInstanceConflict~~	~~**Condition:** The specified managed object instance may not be created as member of the specified class.~~ ~~**Returned Information:** The output parameter status.~~ ~~**Exit state:** Entry state.~~
noSuchAttribute	**Condition:** A specified attribute is not recognized or is not valid for specified object class. **Returned Information:** The output parameter status. **Exit state:** Entry state.
invalidAttributeValue	**Condition:** Value specified for an attribute is not valid for that attribute. **Returned Information:** The output parameter status. **Exit state:** Entry state.
missingAttributeValue	**Condition:** One or more required attribute values were not supplied and default values are not available. **Returned Information:** The output parameter status. **Exit state:** Entry state.
resourceLimitation	**Condition:** Operation not performed due to resource limitation. **Returned Information:** The output parameter status. **Exit state:** Entry state.
complexityLimitation	**Condition:** Operation not performed because a parameter was too complex. **Returned Information:** The output parameter status. **Exit state:** Entry state.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2001	S_12	SP-010283	--	--	New document 32.602 based on 32.106-5 V3.1.0 Approved at TSG SA #12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	S_13	SP-010476	001	--	Replace the current parameter invokeIdentifier with the two parameters invokeIdentifierIn and invokeIdentifierOut in the operations getMoAttributes() and getContainment()	4.0.0	4.1.0
Sep 2002	S_17	SP-020483	002	--	Add Active CM and new methodology, Remove CM Notifications (moved to Kernel CM - 32.66x)	4.1.0	5.0.0

3GPP TSG-SA WG 5 Meeting #32bis
Sophia Antipolis, France, 20-24 January 2003

Tdoc S5- 036117⌘

CR-Form-v7

CHANGE REQUEST

⌘ **32.603** **CR** **009** ⌘**rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 [X] Core Network [X]

Title:	⌘	Add description for notifications of each activeCM operation and one exception for createMO - alignment with 32.602, Information Service
Source:	⌘	S5
Work item code:	⌘	OAM-NIM
Date:	⌘	28/02/2003
Category:	⌘	**F**
Release:	⌘	*Rel-5*

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:	⌘	One exception for createMO needs to be added to align with the IS in 32.602.
Summary of change:	⌘	one exception for createMO is added.
Consequences if not approved:	⌘	Solution Set is not aligned with Information Service.

Clauses affected:	⌘	6.3, Annex A

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

⌘

Other comments: ⌘ This CR is Child of 32602CR003 S5-036116.

Alignment of CMCC/SA5 CM framework

How to create CRs using this form:

1 Scope

The purpose of this *Basic Configuration Management (CM) IRP: CORBA Solution Set* is to define the mapping of the Basic CM IRP: IS (see 3GPP TS 32.602 [4]) to the protocol specific details necessary for implementation of this IRP in a CORBA/IDL environment.

This document defines NRM independent data types and methods.

This Solution Set specification is related to 3G TS 32.602 V5.~~0~~1.X.

6 Mapping

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 (see TS 32.663 [11]) is emitted.

6.2 Operation mapping

The Basic CM IRP: IM (see 3GPP TS 32.602 [4]) defines semantics of operation visible across the Basic Configuration Management IRP. Table 1 indicates mapping of these operations to their equivalents defined in this SS.

Table 1: Mapping from IS Operation to SS equivalents

IS Operation (3GPP TS 32.602 [4])	SS Method	Qualifier
getMoAttributes	BasicCmIrpOperations::find_managed_objects BasicCmInformationIterator::next_basicCmInformations	M
getContainment	BasicCmIrpOperations::find_managed_objects BasicCmInformationIterator::next_basicCmInformations	O
getBasicCmIRPVersion	get_basicCm_IRP_version	M
cancelOperation	BasicCmInformationIterator::destroy	O
CreateMo	BasicCmIrpOperations::create_managed_object	O
deleteMo	BasicCmIrpOperations::delete_managed_objects	O
setMoAttributes	BasicCmIrpOperations::modify_managed_objects	O

6.3 Operation parameter mapping

The Basic CM IRP: IS (see 3GPP TS 32.602 [4]) defines semantics of parameters carried in operations across the Basic Configuration Management IRP. Tables 2 through 8 indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

The SS operation find_managed_objects is equivalent to the IS operation getMoAttributes when called with ResultContents set to NAMES_AND_ATTRIBUTES. Iterating the BasicCmInformationIterator is used to fetch the result.

Table 2: Mapping from IS getMoAttributes parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
invokeIdentifier	- (No equivalence)	-
invokeIdentifierOut	Return value of type BasicCmInformationIterator	M
baseObjectInstance	in DN baseObject	M
scope	in SearchControl searchControl (SearchControl.type and SearchControl.level)	M
filter	in SearchControl searchControl (SearchControl.filter)	M
attributeListIn	in AttributeNameSet requestedAttributes	M
managedObjectClass managedObjectInstance attributeListOut	Return value of type BasicCmInformationIterator - parameter out ResultSet fetchedElements of method next_basicCmInformations	M
status	Exceptions: FindManagedObjects, ManagedGenericIRPSystem::InvalidParameter, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit	M

The SS operation find_managed_objects is equivalent to the IS operation getContainment when called with ResultContents set to NAMES. Iterating the BasicCmInformationIterator is used to fetch the result.

Table 3: Mapping from IS getContainment parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
invokeIdentifier	- (No equivalence)	-
invokeIdentifierOut	Return value of type BasicCmInformationIterator	M
baseObjectInstance	in DN baseObject	M
scope	in SearchControl searchControl (SearchControl.type and SearchControl.level)	O
Not specified in IS	in SearchControl searchControl (SearchControl.filter)	M
containment	Return value of type BasicCmInformationIterator - parameter out ResultSet fetchedElements of method next_basicCmInformations	M
status	Exceptions: FindManagedObjects, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, ManagedGenericIRPSystem::ValueNotSupported, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit	M

Table 4: Mapping from IS getBasicCmIRPVersion parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
versionNumberSet	Return value of type ManagedGenericIRPConstDefs::VersionNumberSet	M
status	Exceptions: GetBasicCmIRPVersion	M

Table 5: Mapping from IS cancelOperation parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
invokeIdentifier	- (Not applicable, the BasicCmInformationIterator instance identifies the ongoing operation)	M
status	Exceptions: DestroyException	M

Table 6: Mapping from IS createMo parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
managedObjectClass managedObjectInstance	in DN objectName	M
referenceObjectInstance	in DN referenceObject	O
attributeListIn attributeListOut	inout MoAttributeSet attributes	M
status	out AttributeErrorSeq attributeErrors Exceptions: CreateManagedObject, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMOException, IllegalDNFormatException, DuplicateMO, CreateNotAllowed, ObjectClassMismatch, NoSuchObjectClass, ParentObjectDoesNotExist	M

Table 7: Mapping from IS deleteMo parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
baseObjectInstance	in DN baseObject	M
scope	in SearchControl searchControl (SearchControl.type and SearchControl.level)	M
filter	in SearchControl searchControl (SearchControl.filter)	M
deletionList	Return value of type DeleteResultIterator - parameter out ResultSet fetchedElements of method next_basicCmInformations	M
status	Return value of type DeleteResultIterator - parameter out DeleteErrorSeq fetchedDeleteErrors of method next_deleteErrors Exceptions: DeleteManagedObjects, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit	M

Table 8: Mapping from IS setMoAttributes parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
baseObjectInstance	in DN baseObject	M
scope	in SearchControl searchControl (SearchControl.type and SearchControl.level)	M
filter	in SearchControl searchControl (SearchControl.filter)	M
modificationList	in AttributeModificationSet modifications	M
modificationListOut	Return value of type ModifyResultIterator - parameter out ResultSet fetchedElements of method next_basicCmInformations	M

IS Operation parameter	SS Method parameter	Qualifier
status	Return value of type ModifyResultIterator - parameter out ModifyAttributeErrorsSeq fetchedModifyErrors of method next_modifyErrors Exceptions: ModifyManagedObjects, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit	M

Annex A (normative):
CORBA IDL, Access Protocol

```
#ifndef BasicCmIRPSystem_idl
#define BasicCmIRPSystem_idl

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

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

module BasicCmIRPSystem
{

   /**
    * Defines the name of a Managed Object Class
    */
   typedef string MOClass;


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


   /**
    * Defines the name of an attribute of a Managed Object
    */
   typedef string MOAttributeName;


   /**
    * Defines the value of an attribute of a Managed Object in form of a CORBA
    * Any. Apart from basic datatypes already defined in CORBA, the allowed
    * attribute value types are defined in the AttributeTypes module.
    */
   typedef any MOAttributeValue;


   /**
    *  This module adds datatype definitions for types
    *  used in the NRM which are not basic datatypes defined
    *  already in CORBA.
    */
   module AttributeTypes
   {

      /**
       * An MO reference referres to an MO instance.
       * "otherMO" contains the  distinguished name of the referred MO.
       * A conceptual "null" reference (meaning no MO is referenced)
       * is represented as an empty string ("").
       *
       */
      struct MOReference
      {
         DN otherMO;
```

```
    };

    /**
     * MOReferenceSet represents a set of MO references.
     * This type is used to hold 0..n MO references.
     * A referred MO is not allowed to be repeated (therefore
     * it is denoted as a "Set")
     */
    typedef sequence<MOReference> MOReferenceSet;


    /**
     *  A set of strings.
     */
    typedef sequence<string> StringSet;

};

 exception IllegalFilterFormatException {
    string reason;
 };
 exception IllegalDNFormatException {
    string reason;
 };
 exception IllegalScopeTypeException {
    string reason;
 };
 exception IllegalScopeLevelException {
    string reason;
 };
 exception UndefinedMOException {
    string reason;
 };

exception UndefinedScopeException {
    string reason;
};

exception FilterComplexityLimit {
   string reason;
};

exception DuplicateMO {};

exception CreateNotAllowed {};

exception ObjectClassMismatch {};

exception NoSuchObjectClass {
   MOClass objectClass;
};

exception ParentObjectDoesNotExist {};

/**
 * System otherwise fails to complete the operation.  System can provide
 * reason to qualify the exception. The semantics carried in reason
 * is outside the scope of this IRP.
 */
exception NextBasicCmInformations { string reason; };
exception NextDeleteErrors { string reason; };
exception NextModifyErrors { string reason; };
exception DestroyException { string reason; };
```

```
exception GetBasicCmIRPVersion { string reason; };
exception FindManagedObjects { string reason; };
exception CreateManagedObject { string reason; };
exception DeleteManagedObjects { string reason; };
exception ModifyManagedObjects { string reason; };


/**
 *
 * In this version the only allowed filter value is "TRUE" i.e. a filter that
 * matches everything.
 */
typedef string FilterType;

 /**
  * ResultContents is used to tell how much information to get back
  * from the find_managed_objects operation.
  *
  * NAMES: Used to get only Distinguished Name
  *        for MOs.
  *        The name contains both the MO class
  *        and the names of all superior objects in the naming
  *        tree.
  *
  * NAMES_AND_ATTRIBUTES: Used to get both NAMES plus
  *    MO attributes (all or selected).
  */
 enum ResultContents
 {
    NAMES,
    NAMES_AND_ATTRIBUTES
 };

 /**
  * ScopeType defines the kind of scope to use in a search
  * together with SearchControl.level, in a SearchControl value.
  *
  * SearchControl.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
 };

 /**
  * SearchControl controls the find_managed_object search,
  * and contains:
  * the type of scope ("type" field),
  * the level of scope ("level" field), level 0 means the "baseObject",
  *    level 1 means baseobject including its sub-ordinates etc..
  * the filter ("filter" field),
  * the result type ("contents" field).
```

```
  * The type, level and contents fields are all mandatory.
  * The filter field contains the filter expression.
  *  The string "TRUE" indicates "no filter",
  * i.e. a filter that matches everything.
  */
 struct SearchControl
 {
    ScopeType type;
    unsigned long level;
    FilterType filter;
    ResultContents contents;
 };

 /**
  * Represents an attribute: "name" is the attribute name
  * and "value" is the attribute value.
  */
struct MOAttribute
{
   MOAttributeName name;
   MOAttributeValue value;
};

typedef sequence<MOAttribute> MOAttributeSet;



 struct Result
 {
    DN mo;
    MOAttributeSet attributes;
 };

 typedef sequence<Result> ResultSet;


 /**
  * AttributeErrorCategory defines the categories of errors, related to
  * attributes, that can occur during creation or modification of MOs.
  *
  * NO_SUCH_ATTRIBUTE: The specified attribute does not exist.
  * INVALID_ATTRIBUTE_VALUE: The specified attribute value is not valid.
  * MISSING_ATTRIBUTE_VALUE: An attribute value is required but none was
  *   provided and no default value is defined for the attribute.
  * INVALID_MODIFY_OPERATOR: The specified modify operator is not valid
  *   (e.g. operator ADD_VALUES applied to a non multi-valued attribute
  *   or operator SET_TO_DEFAULT applied where no default value is defined).
  * MODIFY_NOT_ALLOWED: The modification of the attribute is not allowed.
  * MODIFY_FAILED: The modification failed because of an unspecified reason.
  */
 enum AttributeErrorCategory
 {
    NO_SUCH_ATTRIBUTE,
    INVALID_ATTRIBUTE_VALUE,
    MISSING_ATTRIBUTE_VALUE,
    INVALID_MODIFY_OPERATOR,
    MODIFY_NOT_ALLOWED,
    MODIFY_FAILED
 };


 /**
  * DeleteErrorCategory defines the categories of errors that can occur
```

```
 * during deletion of MOs.
 *
 * SUBORDINATE_OBJECT: The MO cannot be deleted due to subordinate MOs.
 * DELETE_NOT_ALLOWED: The deletion of the MO is not allowed.
 * DELETE_FAILED: The deletion failed because of an unspecified reason.
 */
enum DeleteErrorCategory
{
   SUBORDINATE_OBJECT,
   DELETE_NOT_ALLOWED,
   DELETE_FAILED
};


/**
 * AttributeError represents an error, related to an attribute, that occured
 * during creation or modification of MOs.
 * It contains:
 * - the name of the indicted attribute ("name" field),
 * - the category of the error ("error" field),
 * - optionally, the indicted attribute value ("value" field),
 * - optionally, additional details on the error ("reason" field).
 */
struct AttributeError
{
   MOAttributeName name;
   AttributeErrorCategory error;
   MOAttributeValue value;
   string reason;
};

typedef sequence<AttributeError> AttributeErrorSeq;


/**
 * DeleteError represents an error that occured during deletion of MOs.
 * It contains:
 * - the distinguished name of the indicted MO ("object" field),
 * - the category of the error ("error" field),
 * - optionally, additional details on the error ("reason" field).
 */
struct DeleteError
{
   DN object;
   DeleteErrorCategory error;
   string reason;
};

typedef sequence<DeleteError> DeleteErrorSeq;


/**
 * ModifyAttributeErrors represents errors that occured during
 * modification of attributes of a MO.
 * It contains:
 * - the distinguished name of the indicted MO ("object" field),
 * - a sequence containing the attribute errors ("errors" field).
 */
struct ModifyAttributeErrors
{
   DN object;
   AttributeErrorSeq errors;
};
```

```
typedef sequence<ModifyAttributeErrors> ModifyAttributeErrorsSeq;


/**
 The BasicCmInformationIterator is used to iterate through a snapshot of
  Managed Object Information when IRPManager invokes find_managed_objects.
  IRPManager uses it to pace the return of Managed Object Information.

 IRPAgent controls the life-cycle of the iterator. However, a destroy
  operation is provided to handle the case where IRPManager wants to stop
  the iteration procedure before reaching the last iteration.
 */
interface BasicCmInformationIterator
{

   /**
    This method returns between 1 and "how_many" Managed Object information.
     The IRPAgent may return less than "how_many" items even if there are
     more items to return. "how_many" must be non-zero. Return TRUE if there
     may be more Managed Object information to return. Return FALSE if there
     are no more Managed Object information to be returned.

     If FALSE is returned, the IRPAgent will automatically destroy the
      iterator.

     @parm how_many how many elements to return in the "fetchedElements" out
      parameter.
     @parm fetchedElements the elements.
     @returns A boolean indicating if any elements are returned.
      "fetchedElements" is empty when the BasicCmInformationIterator is
      empty.
    */

    boolean next_basicCmInformations (
      in unsigned short how_many,
      out ResultSet fetchedElements
    )
    raises (NextBasicCmInformations,
            ManagedGenericIRPSystem::InvalidParameter);

    /**
    This method destroys the iterator.
    */

    void destroy ()
    raises (DestroyException);

}; // end of BasicCmInformationIterator


/**
 The DeleteResultIterator is used to iterate through the list of deleted MOs
  when IRPManager invokes method "delete_managed_objects".
  IRPManager uses it to pace the return of Managed Object Information.

 IRPAgent controls the life-cycle of the iterator. However, a destroy
  operation is provided to handle the case where IRPManager wants to stop
  the iteration procedure before reaching the last iteration.
 */
interface DeleteResultIterator : BasicCmInformationIterator
{
```

```
   /**
    Inherited method "next_basicCmInformations" has the same behaviour as
     for interface BasicCmInformationIterator, except that:
     - The Managed Object information returned in parameter
       "fetchedElements" contains only the DNs of the deleted MOs
       (no attributes are returned).
     - If FALSE is returned, the IRPAgent will not automatically destroy the
       iterator.
    */

   /**
    This method returns between 0 and "how_many" deletion errors. The
     IRPAgent may return less than "how_many" items even if there are more
     items to return. "how_many" must be non-zero. Return TRUE if there are
     more deletion errors to return. Return FALSE if there are no more
     deletion errors to be returned.

     If FALSE is returned and last call to inherited method
     "next_basicCmInformations" also returned FALSE (i.e. no more Managed
     Object information to be returned), the IRPAgent will automatically
     destroy the iterator.

     @parm how_many: how many deletion errors to return in the
      "fetchedDeleteErrors" out parameter.
     @parm fetchedDeleteErrors: the deletion errors.
     @returns: a boolean indicating if any deletion errors are returned.
    */

    boolean next_deleteErrors (
      in unsigned short how_many,
      out DeleteErrorSeq fetchedDeleteErrors
    )
    raises (NextDeleteErrors,
            ManagedGenericIRPSystem::InvalidParameter);

}; // end of DeleteResultIterator


/**
 The ModifyResultIterator is used to iterate through the list of modified
  MOs when IRPManager invokes method "modify_managed_objects".
  IRPManager uses it to pace the return of Managed Object Information.

 IRPAgent controls the life-cycle of the iterator. However, a destroy
  operation is provided to handle the case where IRPManager wants to stop
  the iteration procedure before reaching the last iteration.
 */
interface ModifyResultIterator : BasicCmInformationIterator
{

   /**
    Inherited method "next_basicCmInformations" has the same behaviour as
     for interface BasicCmInformationIterator, except that:
     - The Managed Object information returned in parameter
       "fetchedElements" contains DNs and attributes of the modified MOs.
     - If FALSE is returned, the IRPAgent will not automatically destroy the
       iterator.
    */

   /**
    This method returns between 0 and "how_many" modification errors. The
     IRPAgent may return less than "how_many" items even if there are more
     items to return. "how_many" must be non-zero. Return TRUE if there are
```

```
      more modification errors to return. Return FALSE if there are no more
      modification errors to be returned.

      If FALSE is returned and last call to inherited method
      "next_basicCmInformations" also returned FALSE (i.e. no more Managed
      Object information to be returned), the IRPAgent will automatically
      destroy the iterator.

      @parm how_many: how many modification errors to return in the
       "fetchedModifyErrors" out parameter.
      @parm fetchedModifyErrors: the modification errors.
      @returns: a boolean indicating if any modification errors are returned.
     */

     boolean next_modificationErrors (
       in unsigned short how_many,
       out ModifyAttributeErrorsSeq fetchedModifyErrors
     )
     raises (NextModifyErrors,
             ManagedGenericIRPSystem::InvalidParameter);

 }; // end of ModifyResultIterator


typedef sequence<MOAttributeName> AttributeNameSet;


 /**
  * ModifyOperator defines the way in which an attribute value is to be
  * applied to an attribute in a modification of MO attributes.
  *
  * REPLACE: replace the current value with the provide value
  * ADD_VALUES: for a multi-valued attribute, add the provided values to the
  *   current list of values
  * REMOVE_VALUES: for a multi-valued attribute, remove the provided values
  *   from the current list of values
  * SET_TO_DEFAULT: set the attribute to its default value
  */
 enum ModifyOperator
 {
    REPLACE,
    ADD_VALUES,
    REMOVE_VALUES,
    SET_TO_DEFAULT
 };


 /**
  * AttributeModification defines an attribute value and the way it is to
  * be applied to an attribute in a modification of MO attributes.
  * It contains:
  * - the name of the attribute to modify ("name" field),
  * - the value to apply to this attribute ("value" field),
  * - the way the attribute value is to be applied to the attribute
  *   ("operator" field).
 struct AttributeModification
 {
    MOAttributeName name;
    MOAttributeValue value;
    ModifyOperator operator;
 };

 typedef sequence<AttributeModification> AttributeModificationSet;
```

```
/**
  * The BasicCmIrpOperations interface.
  * Supports a number of Resource Model versions.
  */
 interface BasicCmIrpOperations
 {

    /**
     * Get the version(s) of the interface
     *
     * @raises GetBasicCmIRPVersion when the system for some reason
     *   can not return the supported versions.
     * @returns all supported versions.
     */
    ManagedGenericIRPConstDefs::VersionNumberSet get_basicCm_IRP_version()
       raises (GetBasicCmIRPVersion);


    /**
     * Performs a containment search, using a SearchControl to
     * control the search and the returned results.
     *
     * All MOs in the scope constitute a set that the filter works on.
     * The result BasicCmInformationIterator contains all matched MOs,
     * with the amount of detail specified in the SearchControl.
     * For the special case when no managed objects are matched in
     * find_managed_objects, the BasicCmInformationIterator will be returned.
     * Executing the next_basicCmInformations in the
     * BasicCmInformationIterator will return FALSE for
     * completion.
     *
     * @parm baseObject The start MO in the containment tree.
     * @parm searchControl the SearchControl to use.
     * @parm requestedAttributes defines which attributes to get.
     *   If this parameter is empty (""),  all attributes shall
     *   be returned. In this version this is the only supported semantics.
     *   Note that this argument is only
     *   relevant if ResultContents in the search control is
     *   specifed to NAMES_AND_ATTRIBUTES.
     *
     *
     * @raises ManagedGenericIRPSystem::ValueNotSupported if a valid but
     * unsupported parameter value is passed. E.g. the contents
     * field in the searchcontrol parameter contains the value NAMES and
     * the optional getContainment IS operation is not supported.
     * @raises UndefinedMOException The MO does not exist.
     * @raises IllegalDNFormatException The dn syntax string is
     * malformed.
     * @raises IllegalScopeTypeException The ScopeType in scope contains
     * an illegal value.
     * @raises IllegalScopeLevelException The scope level is negative
     * (<0).
     * @raises IllegalFilterFormatException The filter string is
     * malformed.
     * @raises FilterComplexityLimit if the filter syntax is correct,
     *   but the filter is too complex to be processed by the IRP agent.
     * @see SearchControl
     * @see BasicCmInformationIterator
     */
    BasicCmInformationIterator find_managed_objects(in DN baseObject,
                                  in SearchControl searchControl,
```

```
                                    in AttributeNameSet requestedAttributes)
      raises (FindManagedObjects,
              ManagedGenericIRPSystem::ParameterNotSupported,
              ManagedGenericIRPSystem::InvalidParameter,
              ManagedGenericIRPSystem::ValueNotSupported,
              UndefinedMOException,
              IllegalDNFormatException,
              UndefinedScopeException,
              IllegalScopeTypeException,
              IllegalScopeLevelException,
              IllegalFilterFormatException,
              FilterComplexityLimit);

    /**
     * Performs the creation of a MO instance in the MIB maintained
     * by the IRPAgent.
           *
     * @parm objectName: the distinguished name of the MO to create.
     * @parm referenceObject: the distinguished name of a reference MO.
     * @parm attributes: in input, initial attribute values for the MO to
     *   create; in output, actual attribute values of the created MO.
     * @parm attributeErrors: errors, related to attributes, that caused the
     *   creation of the MO to fail.
     *
     * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
     *   is not supported.
     * @raises ManagedGenericIRPSystem::ParameterNotSupported: An optional
     *   parameter is not supported.
     * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
     *   parameter value has been provided.
     * @raises UndefinedMOException: The MO does not exist.
     * @raises IllegalDNFormatException: The DN syntax string is malformed.
     * @raises DuplicateMO: A MO already exist with the same DN as the one
     *   to create.
     * @raises CreateNotAllowed: The creation of the MO is not allowed.
     * @raises ObjectClassMismatch: The object class of the MO to create does
     *   not match with the object class of the provided reference MO.
     * @raises NoSuchObjectClass: The class of the object to create is not
     *   recognized.
     * @raises ParentObjectDoesNotExist: The parent MO instance of the
     *   ManagedEntity specified to be created does not exist.
     */
    void create_managed_object (
        in DN objectName,
        in DN referenceObject,
        inout MoAttributeSet attributes,
        out AttributeErrorSeq attributeErrors
    )
    raises (CreateManagedObject,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter,
            UndefinedMOException,
            IllegalDNFormatException,
            DuplicateMO,
            CreateNotAllowed,
            ObjectClassMismatch,
            NoSuchObjectClass,
            ParentObjectDoesNotExist);

    /**
     * Performs the deletion of one or more MO instances from the MIB
     * maintained by the IRPAgent, using a SearchControl to control the
```

```
 * instances to be deleted.
 *
 * All MOs in the scope constitute a set that the filter works on.
 * All matched MOs will be deleted by this operation.
 * The returned DeleteResultIterator is used to retrieve the DNs of the
 * MOs deleted and the errors that may have occurred preventing deletion
 * of some MOs.
 * For the special case when no managed objects are matched in
 * delete_managed_objects, the DeleteResultIterator will be returned.
 * Executing the next_basicCmInformations in the DeleteResultIterator
        * will return FALSE for completion.
 *
 * @parm baseObject: the start MO in the containment tree.
 * @parm searchControl: the SearchControl to use; field "contents" has no
 *   meaning here and shall be ignored.
  @returns: a DeleteResultIterator (see above).
 *
 * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
 *   is not supported.
 * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
 *   parameter value has been provided.
 * @raises UndefinedMOException: The MO does not exist.
 * @raises IllegalDNFormatException: The DN syntax string is malformed.
 * @raises IllegalScopeTypeException: The ScopeType in scope contains
 *   an illegal value.
 * @raises IllegalScopeLevelException: The scope level is negative (<0).
 * @raises IllegalFilterFormatException: The filter string is malformed.
 * @raises FilterComplexityLimit: The filter syntax is correct,
 *   but the filter is too complex to be processed by the IRPAgent.
 */
DeleteResultIterator delete_managed_objects (
   in DN baseObject,
   in SearchControl searchControl
)
raises (DeleteManagedObjects,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        UndefinedMOException,
        IllegalDNFormatException,
        UndefinedScopeException,
        IllegalScopeTypeException,
        IllegalScopeLevelException,
        IllegalFilterFormatException,
        FilterComplexityLimit);

/**
 * Performs the modification of MO attributes. One or more MOs attributes
 * may be modified according to a SearchControl.
 *
 * All MOs in the scope constitute a set that the filter works on.
 * All matched MOs will have their attributes modified by this operation.
 * The returned ModifyResultIterator is used to retrieve the DNs of the
 * modified MOs together with the values of the modified attributes, and
 * the errors that may have occurred preventing modification of some
 * attributes.
 * For the special case when no managed objects are matched in
 * modify_managed_objects, the ModifyResultIterator will be returned.
 * Executing the next_basicCmInformations in the ModifyResultIterator
 * will return FALSE for completion.
        *
 * @parm baseObject: the start MO in the containment tree.
 * @parm searchControl: the SearchControl to use; field "contents" has no
    meaning here and shall be ignored.
```

```
         * @parm modifications: the values for the attributes to modify and
                the way those values are to be applied to the attributes.
          @returns: a ModifyResultIterator (see above).
         *
         * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
         *   is not supported
         * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
         *   parameter value has been provided
         * @raises UndefinedMOException: The MO does not exist.
         * @raises IllegalDNFormatException: The DN syntax string is malformed.
         * @raises IllegalScopeTypeException: The ScopeType in scope contains
         *   an illegal value.
         * @raises IllegalScopeLevelException: The scope level is negative (<0).
         * @raises IllegalFilterFormatException: The filter string is malformed.
         * @raises FilterComplexityLimit: The filter syntax is correct,
         *   but the filter is too complex to be processed by the IRPAgent.
         */
        ModifyResultIterator modify_managed_objects (
           in DN baseObject,
           in SearchControl searchControl,
           in AttributeModificationSet modifications
        )
        raises (ModifyManagedObjects,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter,
                UndefinedMOException,
                IllegalDNFormatException,
                UndefinedScopeException,
                IllegalScopeTypeException,
                IllegalScopeLevelException,
                IllegalFilterFormatException,
                FilterComplexityLimit);

    };
};
#endif
```

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2001	S_12	SP-010283	--	--	Approved at TSG SA #12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	S_13	SP-010476	001	--	Correction of invokeIdentifier usage	4.0.0	4.1.0
Mar 2002	S_15	SP-020019	002	--	Correction of erroneous CORBA module names and mapping tables	4.1.0	4.2.0
Mar 2002	S_15	SP-020019	003	--	Corrections to Basic CM IRP CORBA Solution Set IDLs	4.1.0	4.2.0
Mar 2002	S_15	SP-020038	004	--	Addition of missing CORBA exception "ManagedGenericIRPSystem::ValueNotSupported" onto CORBA method "find_managed_objects"	4.1.0	4.2.0
Jun 2002	S_16	SP-020294	005	--	Correcting IDL definitions of notification structured event Name Value pair names	4.2.0	4.3.0
Jul 2002	--	--	--	--	Updated the Version number (420->431) and the Date on the cover page	4.3.0	4.3.1
Sep 2002	S_17	SP-020483	006	--	Add Active Basic CM feature - CORBA Solution Set	4.3.1	5.0.0