

Source: SA5 (Telecom Management)
Title: 6 Rel-6 CR 32.612/3/5 Bulk CM IRP IS / CORBA / CMIP SSs
Document for: Approval
Agenda Item: 7.5.3

Doc1stLevel	Specific a	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	Workitemsl D
SP-040807	32.612	011	--	Rel-6	Partition Bulk CM IRP capabilities into packages	F	6.0.0	S5-047109	OAM-NIM
SP-040807	32.612	012	--	Rel-6	BulkCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP	B	6.0.0	S5-049020	OAM-NIM
SP-040807	32.613	010	--	Rel-6	Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields - Align with IS in 32.612	F	6.0.0	S5-047123	OAM-NIM
SP-040807	32.613	011	--	Rel-6	Partition Bulk CM IRP capabilities into separate IDL modules -- Align to IS in 32.612	F	6.0.0	S5-047124	OAM-NIM
SP-040807	32.613	012	--	Rel-6	Add Signalling Transport Network (STN) NRM IRP in BulkCM IRP CORBA SS	B	6.0.0	S5-049049	OAM-NIM
SP-040807	32.615	018	--	Rel-6	Add Signalling Transport Network (STN) NRM IRP in BulkCM IRP XML FF	B	6.0.0	S5-049050	OAM-NIM

CHANGE REQUEST

⌘ **32.612 CR 011** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ Partition Bulk CM IRP capabilities into packages		
Source:	⌘ SA5 (edwin.tse@ericsson.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ F	Release:	⌘ Rel-6
	Use <u>one</u> 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 <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)

Reason for change:	⌘ To allow vendor to implement a 3GPP compliant system that supports only the capability of uploading of configuration management (CM) and inventory management data (IM). To correct definition of event time parameter in notifications.
Summary of change:	⌘ Partition the description of Bulk CM IRP capabilities into 3 separate packages where the most complex package inherits the capabilities of the lesser-complex package. Introduce state diagram for the less-complex packages.
Consequences if not approved:	⌘ It is impossible for vendor to implement a 3GPP compliant system that (a) supports the capability of uploading CM-data and IM-data and (b) does not support the complex active configuration management (download) capabilities.

Clauses affected:	⌘ All								
Other specs affected:	<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> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> </table> Other core specifications ⌘ Test specifications O&M Specifications ⌘ 32.613	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Y	N								
<input type="checkbox"/>	<input checked="" type="checkbox"/>								
<input type="checkbox"/>	<input checked="" type="checkbox"/>								
<input checked="" type="checkbox"/>	<input type="checkbox"/>								
Other comments:	⌘ Parent to Rel-6 CR 32.613 in S5-047124.								

1 Scope

The present document (Bulk Configuration Management IRP: Information Service) defines ~~as~~ [a number of](#) Integration Reference Point (IRP) through which an 'IRPAgent' (typically an Element Manager or Network Element) can communicate bulk Configuration Management related information to one or several 'IRPManagers' (typically Network Managers).

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: "Telecommunication management; Principles and high level requirements".
- [2] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [3] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".
- [4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [5] 3GPP TS 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [6] 3GPP TS 32.652: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [7] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [8] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".
- [9] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".
- [10] 3GPP TS 32.632: "Telecommunication management; Configuration Management (CM); Core Network Resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [11] [3GPP TS 32.692 "Inventory Management \(IM\) network resource Integration Reference Point \(IRP\): Network Resource Model \(NRM\)"](#).

4 System Overview

4.1 System Context

Figure 4.1 and 4.2 identify system contexts of the IRP defined by the present specification in terms of its implementation called IRPAgent and the user of the IRPAgent, called IRPManager. For a definition of IRPManager and IRPAgent, see 3GPP TS 32.102 [2].

The IRPAgent implements and supports this IRP. The IRPAgent can reside in an Element Manager (EM) or a Network Element (NE) (see also [2] clause 8). In the former case, the interfaces (represented by a thick dotted line) between the EM and the NEs is not the subject of this IRP.

An NE can be managed via System Context A or B. The criterion for choosing System Context A or B, to manage a particular NE, is implementation dependent. An IRPAgent shall support one of the two System Contexts. By observing the interaction across the Itf-N, an IRPManager cannot deduce if the EM and NE are integrated in a single system or if they run in separate systems.

As indicated in Figure 4.1 and Figure 4.2, the subject IRP needs to be complemented with the Notification IRP 3GPP TS 32.302 [3]. (This is to allow the IRP Manager to subscribe and unsubscribe to notifications issued by the IRP Agent).

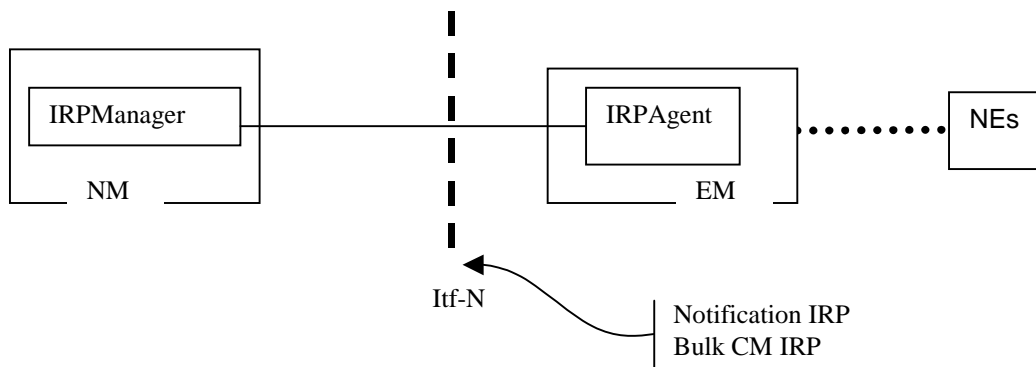


Figure 4.1: System Context A

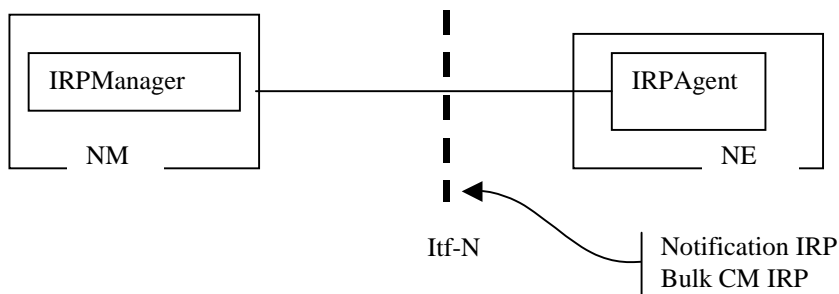


Figure 4.2: System Context B

4.2 Compliance rules

For general definitions of compliance rules related to qualifiers (Mandatory/Optional/Conditional) for *operations, notifications and parameters* (of operations and notifications) please refer to 3GPP TS 32.102 [2].

The following defines the meaning of Mandatory and Optional attributes and associations for Operations, in Solution Sets ~~to the Bulk CMIRP~~:

- The IRPManager shall support all mandatory attributes/associations. The IRPManager shall be prepared to receive information related to mandatory as well as optional attributes/associations without failure; however the IRPManager does not have to support handling of the optional attributes/associations.
- The IRPAgent shall support all mandatory attributes/associations. It may support optional attributes/associations.

An IRPAgent that incorporates vendor-specific extensions must support normal communication with an ~~3GPP SA5-compliant~~ IRPManager with respect to all mandatory and optional managed object classes, attributes, associations, operations, parameters and notifications without requiring the IRPManager to have any knowledge of the extensions.

Given that

- rules for vendor-specific extensions remain to be fully specified, and
- many scenarios under which IRPManager and IRPAgent interwork may exist,

it is recognised that ~~in R4/R5~~ the IRPManager, even though it is not required to have knowledge of vendor-specific extensions, may be required to be implemented with an awareness that extensions can exist and behave accordingly.

4.3 Scope of Bulk CM Management Specification

Within the scope of this document, it is specified how Bulk-CM-IRP ~~IS allows~~ supports the ~~configuration monitoring and provisioning of an IRPManager to actively configure~~ NEs over ~~I~~ interface-N ~~using an IRPAgent supporting Bulk CM IRP IS~~. It is not within the scope of this document to specify how Bulk-CM-IRP ~~IS~~ and the IRPAgent shall resolve any potentially conflicting CM management activities that could arise from either multiple concurrent active IRPManager management Bulk CM IRP sessions, any other IRP conflicting CM management activities, or any CM management activities outside of the scope of an IRP and interface-N. From a system perspective such potential conflicts need to be guarded against, but how this ~~is~~ done e.g. operational procedures or implementation specific recovery in an IRPManager or ~~BulkCMIRP and the IRPAgent~~, is beyond the scope of this document.

[NRMs for Bulk CM IRP are defined in other Network Resource IRP documents of CM. Within the scope of this document, the specified capabilities can manage the following network resource models:](#)

- [32.622: "Configuration Management \(CM\); Generic network resources Integration Reference Point \(IRP\); Network Resource Model \(NRM\)" \[4\].](#)
- [32.632: "Configuration Management \(CM\); Core Network Resources Integration Reference Point \(IRP\); Network Resource Model \(NRM\)" \[10\].](#)
- [32.642: "Configuration Management \(CM\); UTRAN network resources Integration Reference Point \(IRP\); Network Resource Model \(NRM\)" \[5\].](#)
- [32.652: "Configuration Management \(CM\); GERAN network resources Integration Reference Point \(IRP\); Network Resource Model \(NRM\)" \[6\].](#)
- [32.692: "Inventory Management \(IM\) network resource Integration Reference Point \(IRP\); Network Resource Model \(NRM\)" \[11\].](#)

[The above NRM documents define all the MOCs and attributes that can be configuration managed by Bulk CM IRP IS.](#)

[Editor's note: Applicability of this Bulk CM IRP for other NRM IRPs, including the decoupling of Interface IRPs from NRM IRPs, are for further study.](#)

[The types and number of NRM instances that can be managed by SimpleUploadBulkCMIRP, ControlledUploadBulkCMIRP and BulkCMIRP are dependent on network deployment scenarios. For example, an IRP instance may:](#)

- [Manage instances of IOCs defined in CN NRM IRP \[10\] and in IM-data NRM IRP \[11\];](#)
- [Manage instances of IOCs specified in IM-data NRM IRP \[11\].](#)

[Both IRP instances in the above examples are compliant with this specification since only their scope of management differs.](#)

5 Modelling approach

~~This clause identifies the modelling approach adopted and used in this IRP.~~

~~The modelling approach adopted and used in this IRP is the same as that defined in 3GPP TS 32.622: "Generic Network Resources IRP: NRM" [4].~~

[See 3GPP TS 32.102 \[2\] clause 10.](#)

6 Information Object Classes

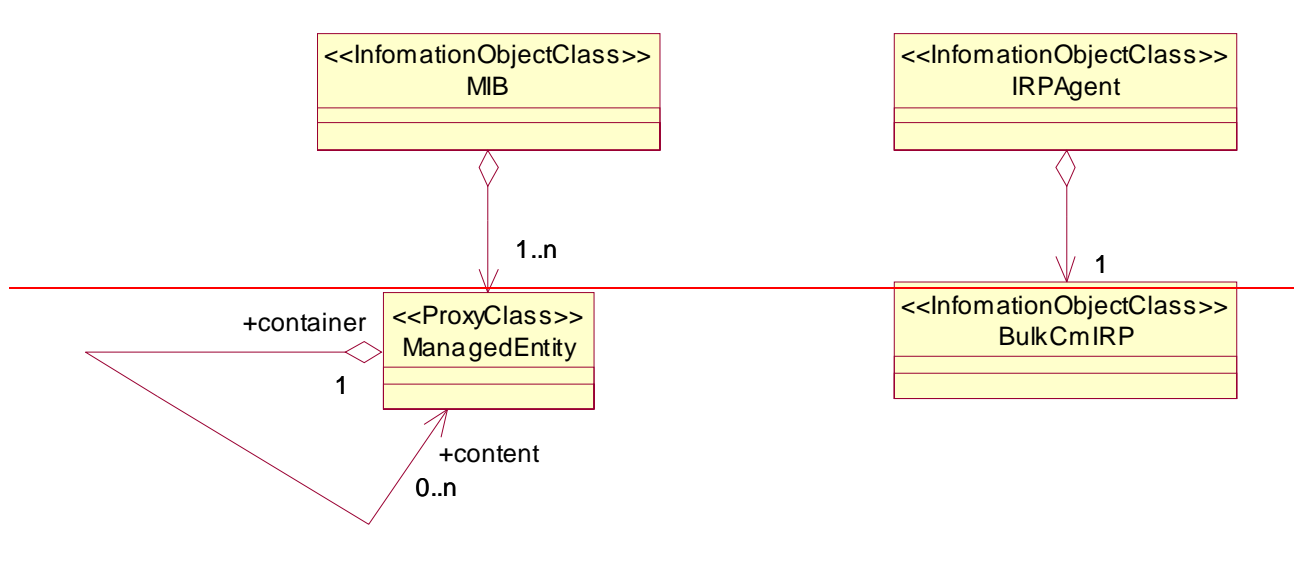
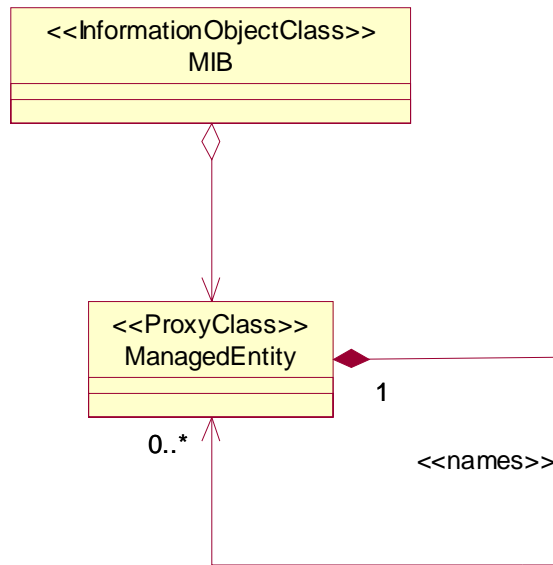
6.1 Information entities imported and local label

Label reference	Local label
32.622 [4], information object class, Top	Top
32.302 [3], information object class, NotificationIRP	NotificationIRP
32.302 [3], interface, notificationIRPNotification	NotificationIRPNotification
32.622 [4], [information object class, GenericIRP	GenericIRP
32.622 [4], information object class, IRPAgent	IRPAgent
32.312 [9], information object class, ManagedGenericIRP	ManagedGenericIRP

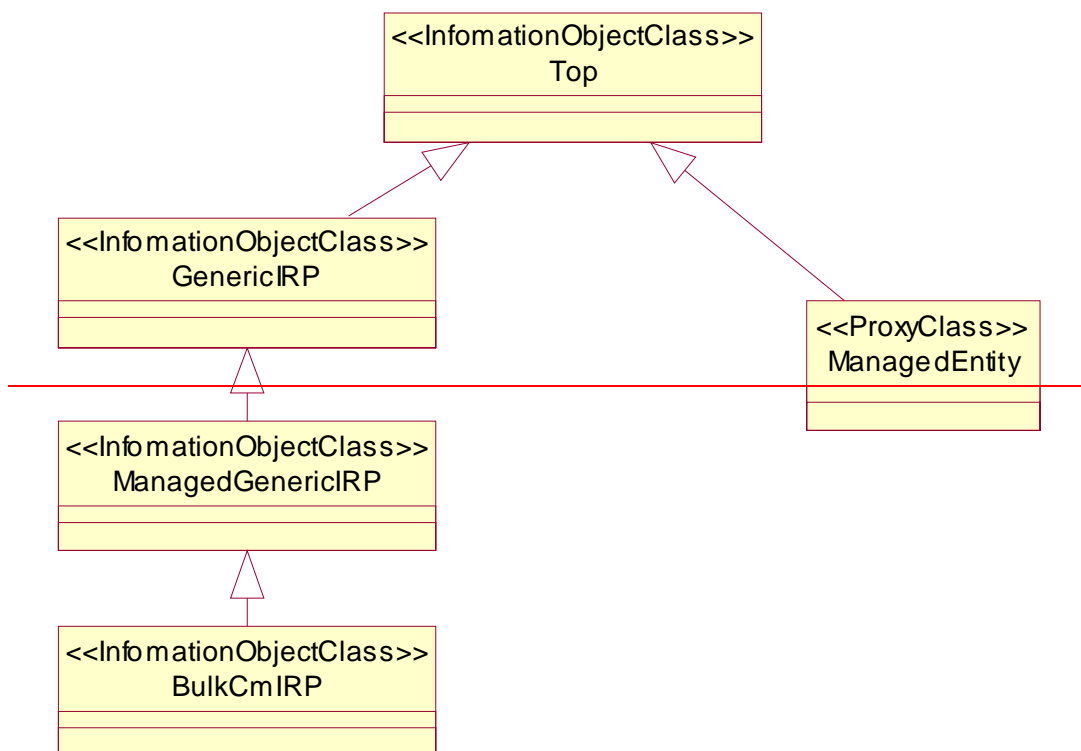
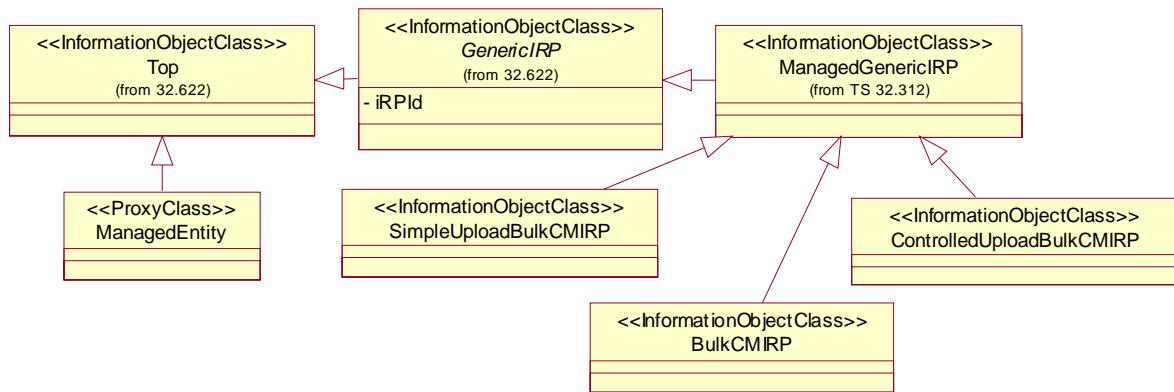
6.2 Class diagram

This clause introduces the set of information object classes (IOCs) that encapsulate [capabilities contained by information within](#) the IRPAgent. ~~The intent is to identify the information required for the BulkCMIRP Agent implementation of its operations and notification emission.~~ This clause provides the overview of all support object classes in UML. Subsequent clauses provide more detailed specification of various aspects of these support object classes.

6.2.1 Attributes and relations



6.2.2 Inheritance



6.3 Information object classes definition

6.3.1 SimpleUploadBulkCMIRP

6.3.1.1 Definition

[It is the representation of the configuration management capabilities specified in subclause 7.1.1. This IOC inherits from ManagedGenericIRP IOC specified in TS.32.312 \[9\].](#)

6.3.2 ControlledUploadBulkCMIRP

6.3.2.1 Definition

It is the representation of the configuration management capabilities specified by subclause 7.1.2. This IOC inherits from ManagedGenericIRP IOC specified in TS.32.312 [9].

6.3.34 BulkCMIRP

6.3.34.1 Definition

BulkCMIRP is the representation of the configuration management capabilities specified by subclause 7.1.3~~this specification~~. This IOC inherits from ManagedGenericIRP IOC specified in TS.32.312 [9].

6.4 Network Resource Model (NRM)

Void.

~~NRMs for Bulk CM IRP are defined in other Network Resource IRP documents of CM. For Bulk CM IRP IS these are:~~

- ~~32.622: "Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [4];~~
- ~~32.632: "Configuration Management (CM); Core Network Resources Integration Reference Point (IRP): Network Resource Model (NRM)" [10];~~
- ~~32.642: "Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [5];~~
- ~~32.652: "Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [6].~~

~~These NRM documents define all the MOCs and attributes that can be configuration managed by Bulk CM IRP IS.~~

7 Interface Definition

Clause 7.1 ~~specifies illustrates~~ the operations and notifications, ~~defined as interfaces implemented and used by~~ of an IRP Agent and IRP Manager, described using UML notation (Interface in IRP Information Model is identical to concepts conveyed by stereotype <<interface>> of UML). Parameters and return status are not indicated.

~~Configuration data files, defined in clause 10, define bulk configuration management changes exchanged between BulkCMIRP of an IRP Agent and IRP Manager.~~

~~The BulkCMIRP specifies a number of operations and notifications, grouped under Interfaces (see clause 6.3 to 6.7)The interfaces support multiple IRP Managers connected to an IRP Agent. Configuration data files defined in clause 10 define bulk configuration management changes. The following configuration data file handling operations exist in the If N.~~

- ~~•startSession~~
- ~~•endSession~~
- ~~•upload~~
- ~~•download~~
- ~~•validate~~
- ~~•preactivate~~

- ~~activate~~
- ~~fallback~~
- ~~abortSessionOperation~~
- ~~getSessionIds~~
- ~~getSessionStatus~~
- ~~getSessionLog~~
- ~~getBulkCmIRPVersion~~

~~The BulkCMIRP functions in conjunction with the NotificationIRP [3] notifications (see System Context A and System Context B of figure 4.1 and 4.2 respectively). In order for the IRPManager to receive the specified notifications, the IRPManager must use the subscribe and unsubscribe operations defined in Notification IRP [3].~~

~~the BulkCMIRP must work in conjunction with a NotificationIRP (see System Context A and System Context B of figure 4.1 and 4.2 respectively)., related operations are also associated with Bulk CM IRP (e.g. Subscribe and Unsubscribe), but these operations are described in 32.302: "Telecommunication Management; Notification Management: Part 2: Notification IRP; Information Service " [3].~~

The operations, `upload`, `_download`, `validate`, `preactivate`, `activate`, `fallback` and `getSessionLog` are performed asynchronously in that when the operations are initiated, the BulkCMIRP of the IRPAgent returns an indication that the requested activity has begun, and the IRPManager may release and continue with other tasks. If the IRPManager has subscribed on event notifications, then the IRPManager will receive a notification when the task requested in the operation is complete.

The operations `startSession`, `endSession`, `abortSessionOperation`, `getSessionIds`, `getSessionStatus` and `getBulkCmIRPVersion`, *etc.* are performed synchronously in that the result of the operation is returned as a callback to the operation, and the IRPManager will wait until the response is received before continuing. Refer to clause 4.3 for system conditions that need to be potentially managed, but are outside the scope of this document.

The operations and notifications of this document are specified and grouped under Interfaces. To allow the flexible support of the necessary and sufficient operations and notifications for various resources monitoring and provisioning needs, the operations and notifications of this specification are packaged to supporting various ~~needs~~ network management applications:

1. BulkCM SimpleUpload, enabling upload of resource information by the IRPManager without explicit session control. This requires the following Interface and Notification specified in clause 7.1.1:

- o BulkCMPassive
- o BulkCMNotification_1

2. BulkCM Controlled Upload, enabling a session controlled upload of resource information by the IRPManager. This requires the following Interfaces and Notification specified in clause 7.1.2:

- o BulkCMPassive
- o BulkCMSession
- o BulkCMNotification_1
- o BulkCMNotification_2

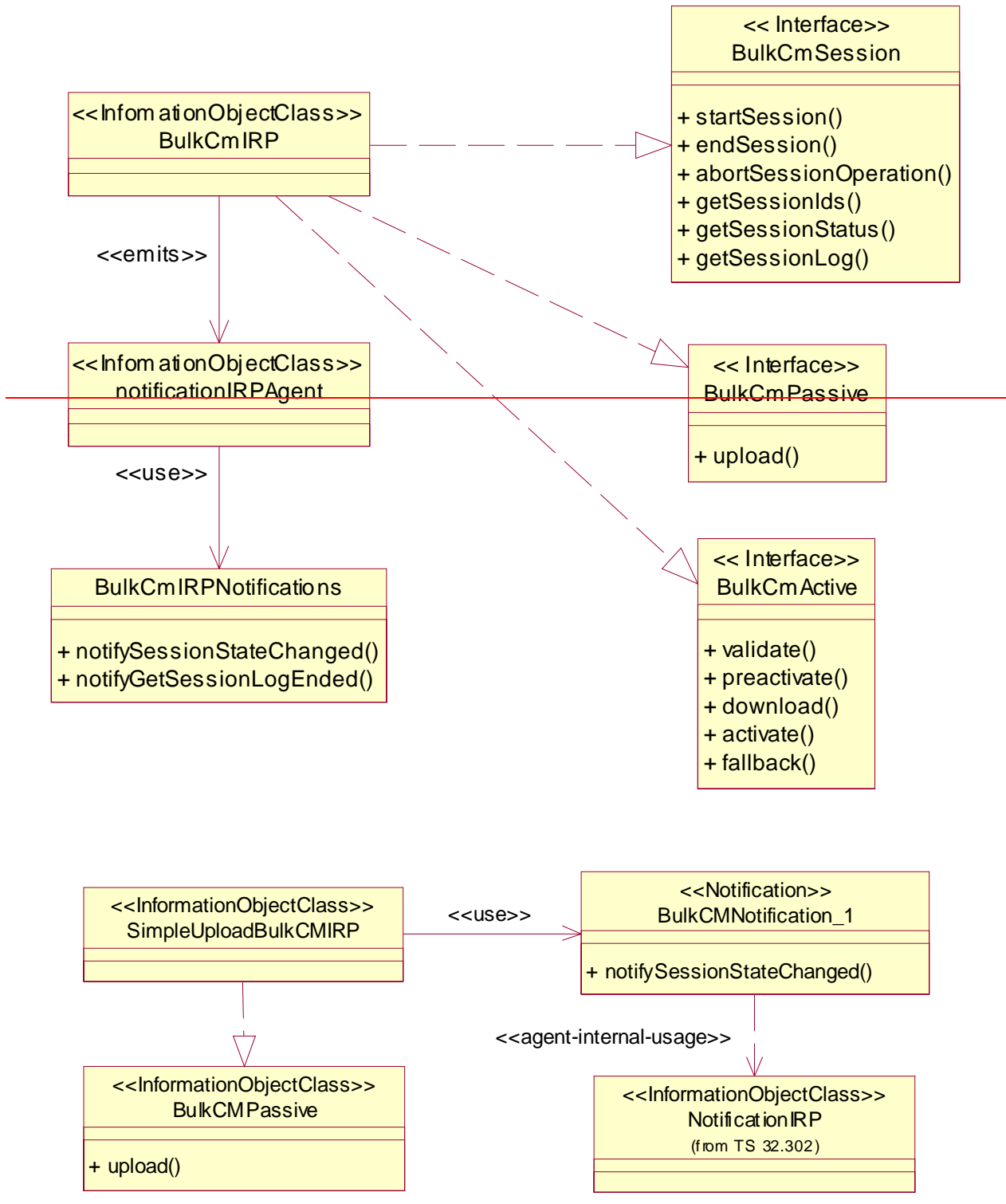
3. BulkCM Controlled Upload & Provisioning, enabling a session controlled upload and provisioning of resource information by the IRPManager. This requires ~~and requiring~~ the following Interfaces and Notification specified in clause 7.1.3:

- o BulkCMPassive
- o BulkCMSession
- o BulkCMActive
- o BulkCMNotification_1
- o ~~BulkCMNotification_2~~ BulkCMPassive
- o ~~BulkCMNotification_1~~

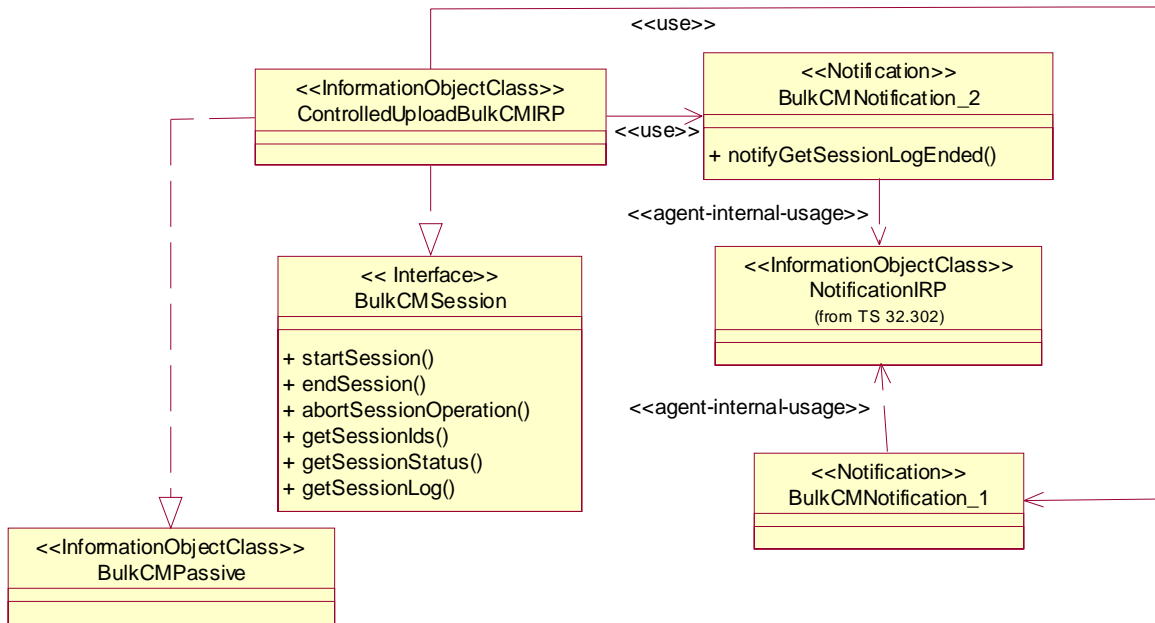
==

7.1 Class Diagram

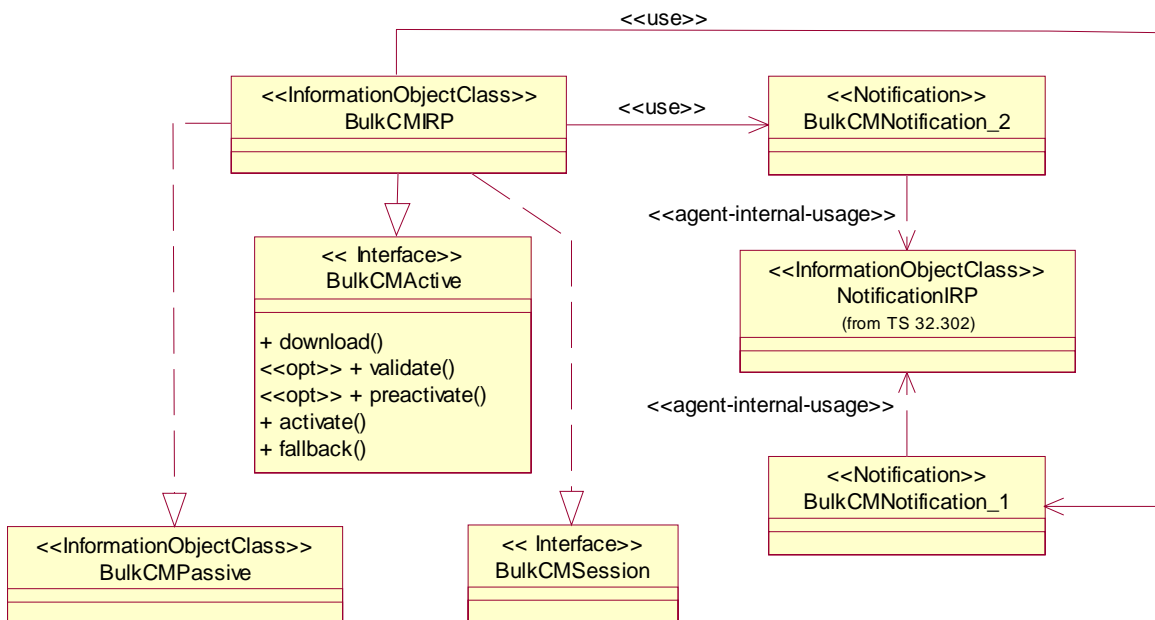
7.1.1 Operations and Notifications [for Simple Upload](#)



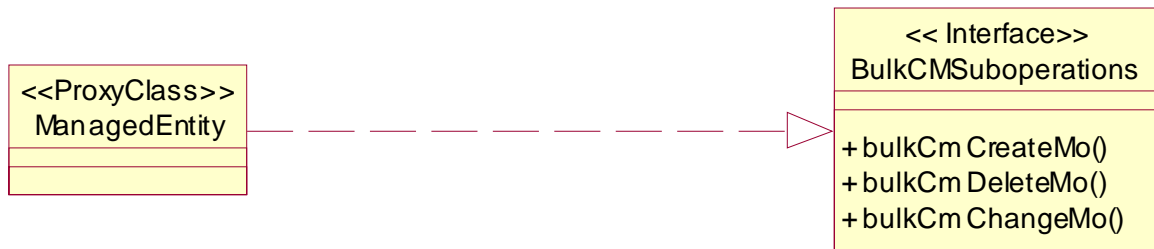
7.1.2 Operations and Notifications for Controlled Upload



7.1.3 Main Operations and Notifications for Controlled Upload & Provisioning



7.1.42 Suboperations [for Controlled Upload & Provisioning](#) (of clause 10)



7.3 ~~bulkCmSession~~ Interface [BulkCMSession](#)

7.3.5 Operation getSessionStatus (M)

7.3.5.1 Definition

The IRPManager invokes this operation to request the IRPAgent to send the current state of the bulk configuration data file operation. The IRPAgent returns the current state. See clause 9.

This operation can be invoked in any session state and does not change the session state.

7.3.5.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which the current status is required.

7.3.5.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
sessionState	M	List of ENUM(Idle, Upload In Progress, Upload Failed, Upload Completed, Down Load In Progress, Download Failed, Download Completed, Validation In Progress, Validation Failed, Validation Completed, Preactivation In Progress, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation In Progress, Activation Failed, Activation Partly Realised, Activation Completed, Fallback In Progress, Fallback Failed, Fallback Partly Realised, Fallback Completed)	Indicates current state of the configuration data file operation. See clause 9, i.e. will be one of: Idle, Upload In Progress, Upload Failed, Upload Completed, Down Load In Progress, Download Failed, Download Completed, Validation In Progress, Validation Failed, Validation Completed, Preactivation In Progress, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation In Progress, Activation Failed, Activation Partly Realised, Activation Completed, Fallback In Progress, Fallback Failed, Fallback Partly Realised, Fallback Completed.
status	M		Indicates (a) start of operation is successful or (b) operation failed because of specified or unspecified reasons

7.4 ~~bulkCmPassive~~ Interface BulkCMPassive

7.4.1 Operation upload (M)

7.4.1.1 Definition

An IRPManager invokes this operation to request the IRPAgent to create a file containing bulk configuration data (clause 10) and transfer the file to the indicated globally unique data file reference.

7.4.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
s SessionId	M	String identifying the session	Identifies this specific session and process associated with the requested bulk data upload.
u UploadDataFileReference	M	String of complete path of file and name.	This specifies a globally unique file reference to where the specified scope of bulk data is to be uploaded and stored .
b BaseObjectInstance	M	DistinguishedName	The DN of the MO where the search (see parameter scope below) starts. The DN is in accordance a full Distinguished Name according to 3GPP TS 32.300 [7].
s Scope	M	SEQUENCE < ENUM { BASE_OBJECT_ONLY, NTH_LEVEL_SUBORDINATES, BASE_NTH_LEVEL, BASE_ALL}, Integer >	This parameter defines how many levels of the containment hierarchy to search (i.e. apply the filter defined below). The search starts from the MO given by the baseObjectInstance parameter. The levels of search that may be performed are: <ul style="list-style-type: none"> • the base object alone (default); • the n-th level subordinates of the base object; • the base object and all of its subordinates down to and including the n-th level; • the base object and all of its subordinates.
filter	M	See comment.	This parameter defines a filter test to be applied to the scoped Managed Object(s). If the filter is empty, all of the managed objects included by the scope are selected. The actual syntax and capabilities of the filter is Solution Set specific. However, each Solution Set support a filter consisting of one or several assertions that may be grouped using the logical operators AND, OR and NOT. Each assertion is a logical expression of attribute existence, attribute value comparison ("equal to X, less than Y" etc.) and MO Class.

7.4.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM(OperationSucceeded, OperationFailed).	indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

7.4.1.4 Pre-condition

sessionIdle

Assertion Name	Definition
sessionIdle	State as defined in clause 9. The Bulk CM IRP Agent has successfully opened the session either explicitly (as in the case of Controlled Upload, see 7.1.2, and Controlled Upload & Provisioning, see 7.1.3) or implicitly (as in the case of Simple Upload, see 7.1.1) and is ready to handle the first operations of the session or repeat this operation.

7.4.1.5 Post-condition

uploadInProgress

Assertion Name	Definition
upload in progress	State = UPLOAD_IN_PROGRESS, as defined in clause 9. The Bulk CM IRP Agent has successfully started the upload of the request configuration data.

7.4.1.6 Exceptions

7.4.1.6.1 operationFailed

Exception Name	Definition
operationFailed	<p>Condition: Pre-condition is false or post-condition is false.</p> <p>Returned information: The output parameter status.</p> <p>Exit state: Entry state.</p>

7.5 ~~bulkCmActive~~ Interface BulkCMActive

...

7.5.2 Operation validate (O)

7.5.2.1 Definition

An IRPManager invokes this operation to request an IRPAgent to validate previously downloaded bulk configuration data (clause.10), see clause 7.5.1. Use of this optional operation enables an IRPManager to detect errors with regard to the previously downloaded bulk configuration data before requesting preactivation or activation. See clause 7.5.6 for scope and types of errors attempted to be detected.

Specifying an activation mode is optional. There can only be one activation mode for a session. If an activation mode is specified for the validate, it shall be when the first validate operation is requested. If an activation mode was specified for the first validate operation, it is not possible to change the activation mode initially specified with any subsequent validate retries. (If another activation mode is required; a new session, download, validate, preactivate and activate should be started.). If no activation mode is specified for the first validate, it cannot be subsequently specified with any subsequent validate retries. (If specification of an activation mode is required; a new session, download, validate, preactivate and activate should be started.) If an activation mode is specified for the validate, it cannot be specified for the preactivation or activation. If no activation mode is specified for the validate operation, it cannot be specified for the preactivation or activation. See also clauses 7.5.3 and 7.5.4.

Use of the validate operation shall have no influence on the fallback behaviour of a session.

Invoking the validate operation shall not result in any of the suboperations specified in the downloaded bulk configuration data being applied (clause 10). The operation is essentially passive.

7.5.2.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	Identifies this specific session and process associated with the requested bulk data download.
activationMode	O		Identifies whether a specific activation mode is required. See also clauses 7.5.3 and 7.5.4. The valid choices are defined in the parameter table in clause 7.5.4.

7.5.2.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM(OperationSucceeded, OperationFailed).	indicates (a) start of operation is successful or (b) operation failed because of specified or unspecified reasons

7.5.2.4 Pre-condition

downloaded

Assertion Name	Definition
downloaded downloaded	State as defined in clause 9. The Bulk CM IRP Agent has successfully opened the session and download had been attempted or repeat this operation.

7.5.2.5 Post-condition

validationInProgress

Assertion Name	Definition
validationInProgress	State = VALIDATE_IN_PROGRESS, as defined in clause 9. The Bulk CM IRP Agent has successfully started the validation of the downloaded configuration data.

7.5.2.6 Exceptions

7.5.2.6.1 operationFailed

Exception Name	Definition
operationFailed	Condition: Pre-condition is false or post-condition is false. Returned information: The output parameter status. Exit state: Entry state.

7.5.3 Operation preactivate (O)

...

7.5.3.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	Identifies this specific session and process associated with an earlier bulk data download that is required to be activated.
VerificationMode	O		Selects the mode of checking. One of two choices may be selected: "full checking", "limited checking", see clause 7.5.6.3.
activationMode	O		Identifies whether a specific activation mode is required. See also clauses 7.5.2 and 7.5.4. The valid choices are defined in the parameter table in clause 7.5.4.
fallbackEnabled	M		Indicates whether or not it is required to initialise and enable fallback option prior to the preactivation. This option is only open for the first preactivate operation of a session. For any subsequent preactivate operation retries within a session the fallbackEnabled parameter must be set to indicate it is not required to initialise fallback otherwise the pre-activate operation retry shall fail.

7.5.3.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM(OperationSucceeded, OperationFailed).	indicates (a) start of operation is successful or (b) operation failed because of specified or unspecified reasons

7.5.3.4 Pre-condition

Assertion Name	Definition
downLoaded downloaded	State as defined in clause 9. The Bulk CM IRP Agent has successfully opened the session and download had been attempted or repeat this operation.

7.5.2.5 Post-condition

preactivationInProgress

Assertion Name	Definition
preactivationInProgress	State = PREACTIVATION_IN_PROGRESS, as defined in clause 9. The Bulk CM IRP Agent has successfully started the validation of the downloaded configuration data.

7.5.3.6 Exceptions

7.5.3.6.1 operationFailed

Exception Name	Definition
operationFailed	Condition: Pre-condition is false or post-condition is false. Returned information: The output parameter status. Exit state: Entry state.

7.5.4 Operation activate (M)

...

7.5.4.4 Pre-condition

downLoaded

Assertion Name	Definition
downLoaded downloaded	State as defined in clause 9. The Bulk CM IRP Agent has successfully opened the session and download had been attempted or repeat this operation.

...

7.5.5 Operation fallback (M)

...

7.5.5.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which the current log is required.

7.5.5.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
sStatus	M	ENUM(OperationSucceeded, OperationFailed).	indicates (a) start of operation is successful or (b) operation failed because of specified or unspecified reasons

7.5.5.4 Pre-condition

fallbackEnabled

Assertion Name	Definition
fallbackEnabled	State as defined in clause 9. The Bulk CM IRP Agent has successfully opened the session and fallbackEnables=True by either, preactivate or activate operations being successfully invoked, as defined in clauses 7.5.3 and 7.5.4.

7.5.5.5 Post-condition

fallbackInProgress

Assertion Name	Definition
fallbackInProgress	State = FALLBACK_IN_PROGRESS, as defined in clause 9. The Bulk CM IRP Agent has successfully started the fallback.

7.5.5.6 Exceptions

7.5.5.6.1 operationFailed

Exception Name	Definition
operationFailed	Condition: Pre-condition is false or post-condition is false. Returned information: The output parameter status. Exit state: Entry state.

7.5.6 Validation and Checking Functions

...

7.5.6.4 Activate Checks

During the activation the same checks as for validate and preactivate should be performed if these operations have not previously been successfully performed. These checks may also be repeated if the context may have changed.

[7.6 Interface BulkCMIRPNotification 1](#)

[7.6.1 Notification notifySessionStateChanged \(M\)](#)

[7.6.1.1 Definition](#)

[The IRP Agent notifies the IRP Manager that a state change has occurred on a bulk -configuration data file sessionId operation subscribed to by the IRP Manager. E.g. a configuration data file is available for processing after an upload, a download is complete. See clause 9 for a further description of states.](#)

7.6.1.2 Input Parameters

<u>Parameter Name</u>	<u>Qualifiers</u>	<u>Matching Information</u>	<u>Comment</u>
objectClass	O, Y	ManagedEntity.objectClass	Notification header - see [3].
objectInstance	O, Y	ManagedEntity.objectInstance.	Notification header - see [3].
notificationId	O, N	This carries the semantics of notification identifier.	Notification header - see [3].
eventTime	M, Y	--	Notification header - see [3].
systemDN	O, C, Y	IRPAgent.systemDN where the IRPAgent is related to the KernelCmIRP.	Notification header - see [3].
notificationType	M, Y	Mapped to notificationType in [3].	Notification header - see [3]. For this notification it indicates notification type is Notify Session State Changed.
sessionId	M, N	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which the current status is required.
sourceIndicator	O, N		This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: resource operation: The notification was generated in response to an internal operation of the resource; management operation: The notification was generated in response to a management operation applied across the managed object boundary external to the managed object; unknown: It is not possible to determine the source of the operation.
sessionState	M, N	ENUM(Upload Failed, Upload Completed, Download Failed, Download Completed, Validation Failed, Validation Completed, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation Failed, Activation Partly Realised, Activation Completed, Fallback Failed, Fallback Partly Realised, Fallback Completed)	Indicates the state transition that caused the Notification. See clause 7. i.e. : Upload Failed, Upload Completed, Download Failed, Download Completed, Validation Failed, Validation Completed, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation Failed, Activation Partly Realised, Activation Completed, Fallback Failed, Fallback Partly Realised, Fallback Completed. (Note: as per clause 7.2 "in-progress" transition states are not notified)

7.6.1.3 Triggering events

[State transitions as defined in clause 9.](#)

7.7 Interface BulkCMIRPNotification 2

7.7.1 Notification NotifyGetSessionLogEnded (M)

7.7.1.1 Definition

[The IRPAgent notifies the IRPManager that a requested GetSessionLog for a bulk data configuration file sessionId operation subscribed to by the IRPManager has ended successfully or unsuccessfully.](#)

7.7.1.2 Input parameters

<u>Parameter Name</u>	<u>Qualifiers</u>	<u>Matching Information</u>	<u>Comment</u>
objectClass	O, Y	ManagedEntity.objectClass	Notification header - see [3].
objectInstance	O, Y	ManagedEntity.objectInstance.	Notification header - see [3].
notificationId	O, N	This carries the semantics of notification identifier.	Notification header - see [3].
eventTime	M, Y	--	Notification header - see [3].
systemDN	O, C, Y	IRPAgent.systemDN where the IRPAgent is related to the KernelCmIRP.	Notification header - see [3].
notificationType	M, Y	Mapped to notificationType in [3]	Notification header - see [3].For this notification it indicates notification type is Notify Bulk CM Log State.
sessionId	M, N	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which Log State is required.
sourceIndicator	O, N		This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: resource operation: The notification was generated in response to an internal operation of the resource; management operation: The notification was generated in response to a management operation applied across the managed object boundary external to the managed object; unknown: It is not possible to determine the source of the operation.
sessionLogStatus	M, N	Boolean = GetSessionLog completed successfully or GetSessionLog completed unsuccessfully	Indicates event that caused the Notification i.e. GetSessionLog completed successfully, GetSessionLog completed unsuccessfully.

7.7.1.3 Triggering event

[Attempt to transfer session log to destinations completed successfully or failed. Session state independent, see clause 9.](#)

8 ~~Bulk Configuration Data File~~ Void

8 —

8.1 Interface BulkCmIRPNotifications#1

8.1.1 Notification notifySessionStateChanged (M)

8.1.1.1 Definition

The IRPAgent notifies the IRPManager that a state change has occurred on a bulk-configuration data file sessionId operation subscribed to by the IRPManager. E.g. a configuration data file is available for processing after an upload, a download is complete. See clause 9 for a further description of states.

Parameter Name	Qualifiers	Matching Information	Comment
objectClass	O,F	ManagedEntity.objectClass	Notification header – see [3].
objectInstance	O,F	ManagedEntity.objectInstance.	Notification header – see [3].
notificationId	O	This carries the semantics of notification identifier.	Notification header – see [3].
eventTime	M,F	ManagedEntity.creationTime	Notification header – see [3].
systemDN	O,C,F	IRPAgent.systemDN where the IRPAgent is related to the KernelCmIRP.	Notification header – see [3].
NotificationType	M,F	Mapped to notificationType in [3].	Notification header – see [3]. For this notification it indicates notification type is Notify Session State Changed.
sessionId	M	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which the current status is required.
sourceIndicator	O		This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: resource operation: The notification was generated in response to an internal operation of the resource; management operation: The notification was generated in response to a management operation applied across the managed object boundary external to the managed object; unknown: It is not possible to determine the source of the operation.
sessionState	M	ENUM(Upload Failed, Upload Completed, Download Failed, Download Completed, Validation Failed, Validation Completed, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation Failed, Activation Partly Realised, Activation Completed, Fallback Failed, Fallback Partly Realised, Fallback Completed)	Indicates the state transition that caused the Notification. See clause 9. i.e. Upload Failed, Upload Completed, Download Failed, Download Completed, Validation Failed, Validation Completed, Preactivation Failed, Preactivation Partly Realised, Preactivation Completed, Activation Failed, Activation Partly Realised, Activation Completed, Fallback Failed, Fallback Partly Realised, Fallback Completed. (Note: as per clause 9.2 "in-progress" transition states are not notified)

8.1.1.3 Triggering events

State transitions as defined in clause 9.

8.1.1 Notification NotifyGetSessionLogEnded (M)

8.1.1.1 Definition

The IRPAgent notifies the IRPManager that a requested GetSessionLog for a bulk data configuration file sessionId operation subscribed to by the IRPManager has ended successfully or unsuccessfully.

8.1.1.2 Input parameters

Parameter Name	Qualifiers	Matching Information	Comment
objectClass	O,F	ManagedEntity.objectClass	Notification header -- see [3].
objectInstance	O,F	ManagedEntity.objectInstance.	Notification header -- see [3].
notificationId	O	This carries the semantics of notification identifier.	Notification header -- see [3].
eventTime	M,F	ManagedEntity.creationTime	Notification header -- see [3].
systemDN	O,C,F	IRPAgent.systemDN where the IRPAgent is related to the KernelCmIRP.	Notification header -- see [3].
NotificationType of notificationHeader	M,F	Mapped to notificationType in [3]	Notification header -- see [3]. For this notification it indicates notification type is Notify Bulk CM Log State.
SessionId	M	String identifying the session	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which Log State is required.
SourceIndicator	O		This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: resource operation: The notification was generated in response to an internal operation of the resource; management operation: The notification was generated in response to a management operation applied across the managed object boundary external to the managed object; unknown: It is not possible to determine the source of the operation.
SessionLogStatus	M	Boolean = GetSessionLog completed successfully or GetSessionLog completed unsuccessfully	Indicates event that caused the Notification i.e. GetSessionLog completed successfully, GetSessionLog completed unsuccessfully.

8.1.1.3 Triggering event

Attempt to transfer session log to destinations completed successfully or failed. Session state independent, see clause 9.

9 State Machine

...

9.2 State Machine Description

The IRPAgent progresses Bulk CM operations and associated configuration data changes (clause 10) within a session according to the state machine defined here. The IRPManager can manage a configuration session using session state change notifications which are triggered by the IRPAgent. Not all state changes defined here are notified to the IRPManager. The transition states (UPLOAD_IN_PROGRESS, DOWNLOAD_IN_PROGRESS, VALIDATION_IN_PROGRESS, PREACTIVATION_IN_PROGRESS, ACTIVATION_IN_PROGRESS) are not notified to the IRPManager as they are not required.

If the IRPManager becomes unaware or needs to confirm the current state of a configuration session it can request this by invoking getSessionStatus operation. It is not required to know the history of the state machine. The getSessionStatus operation will provide the "actual" current status.

An IRPManager may request the status when it detects loss of control, for example because of the following reasons:

- 1) Session state change notifications are not being received as expected, e.g. because IRPAgent is blocked in a transition state, e.g. ACTIVATION_IN_PROGRESS;
- 2) IRPManager gets disconnected from the IRPAgent, e.g. session state notification are not received.

The session state notification events are considered a subset of the state machine (without transition state). The actual configuration state can be requested via getSessionStatus. Because of this common behaviour it is reasonable to define one interface type for the state machine handling which is used in the session state notification and in the getSessionStatus operation.

The IRPManager will only receive notifications if it registered itself at the IRPAgent with the subscribe operation.

For ease of description the state machine of a configuration session is introduced with the notion of substate machines but state itself are named unique. This kind of notion is not to be interpreted as providing implementation directions.

Within the description of the substate machines it is becoming clear that they have the following state symmetries:

- The state of the UPLOAD_PHASE, the DOWNLOAD_PHASE and the VALIDATION_PHASE are similar.
- The state of the ACTIVATION_PHASE, PREACTIVATION_PHASE and the FALLBACK_PHASE are similar.

The startSession operation creates a state machine. The initial state of the configuration session in the IDLE_PHASE is IDLE. The endSession deletes a state machine which is not in a transition state, more details are defined in the substate machines.

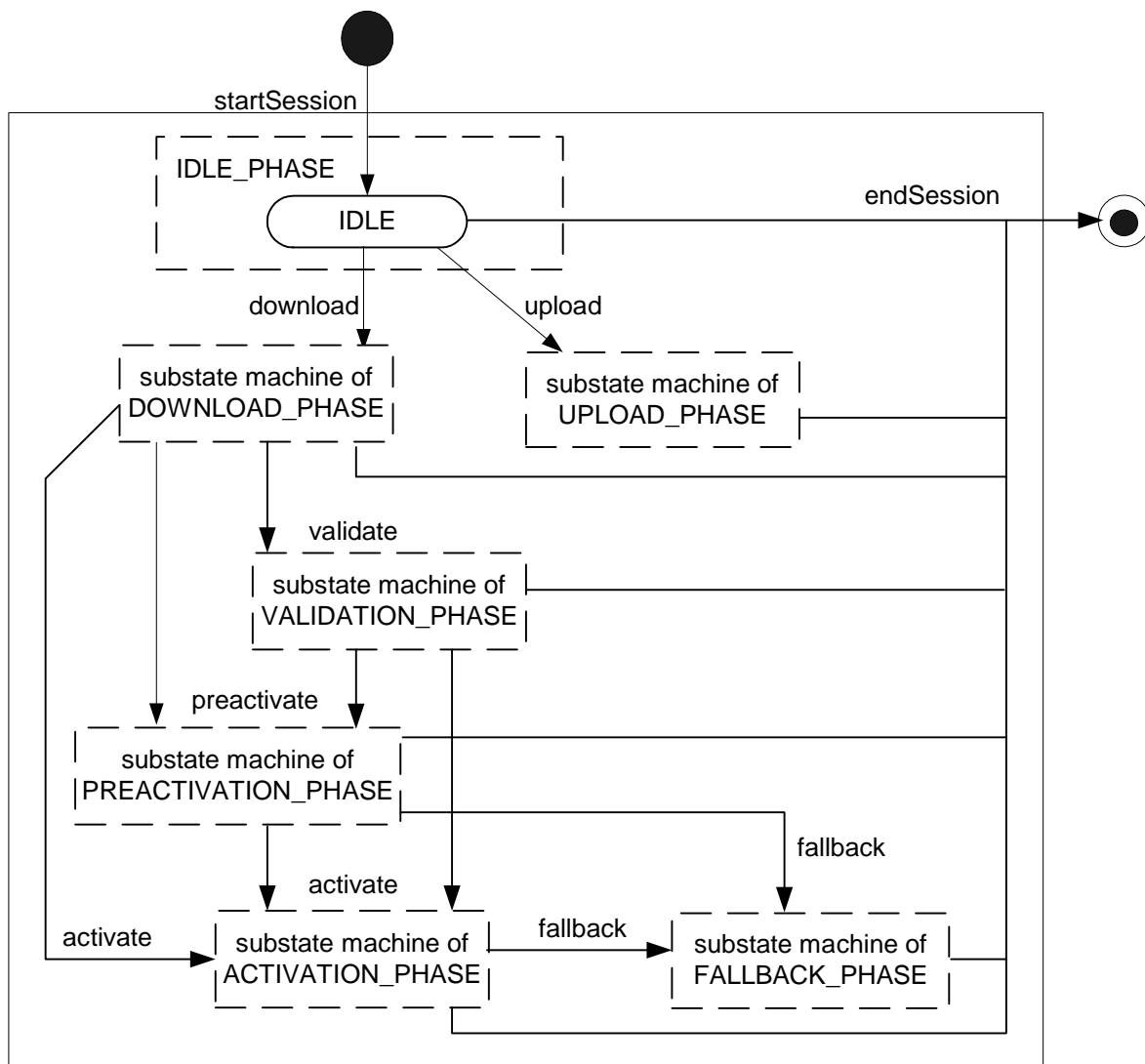


Figure 1: State Machine for Controlled Upload and Controlled Upload & Provisioning

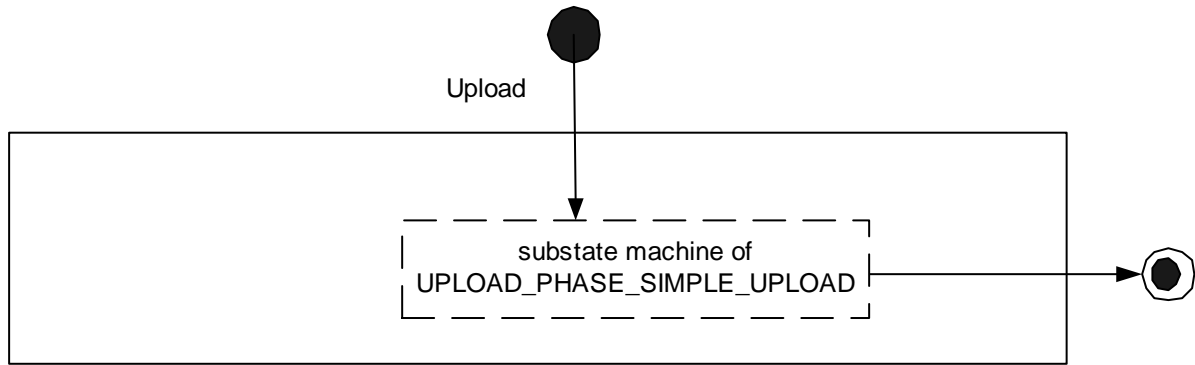


Figure 2: State Machine for Simple Upload

Figure - State Machine

The following figures describe the substate machine of a configuration session. The transition states, DOWNLOAD_IN_PROGRESS, UPLOAD_IN_PROGRESS, VALIDATION_IN_PROGRESS, PREACTIVATION_IN_PROGRESS and ACTIVATION_IN_PROGRESS, are either left implicit if the IRP Agent finished the processing or explicit via an abortSessionOperation operation from the IRPManager.

In these figures solid transition lines indicate the transition is caused by an external event and dashed transition lines indicate the transition is caused by an internal event or decision as depicted in the following figure.

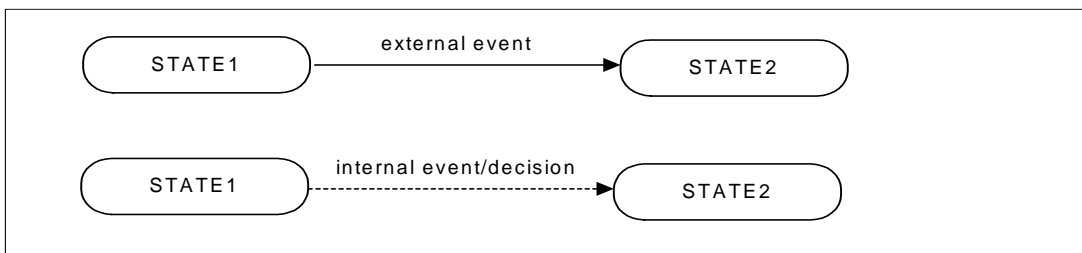


Figure 3: Depicting State Transition Lines for Internal and External Events and Decision

Figure - Depicting State Transition Lines for Internal and External Events and Decision

9.2.1 Upload Phase

When the upload is triggered the IRP Agent writes the requested configuration data into a configuration data file and copies to the file reference provided by the IRP Manager. If the process succeeds the state UPLOAD_COMPLETED is indicated. If the upload fails a retry can be triggered in state UPLOAD_FAILED.

Once a session is associated with an upload none of the other state changes phases outside of the upload phase, i.e., download, validate, preactivate and activate phases cannot be triggered for the session.

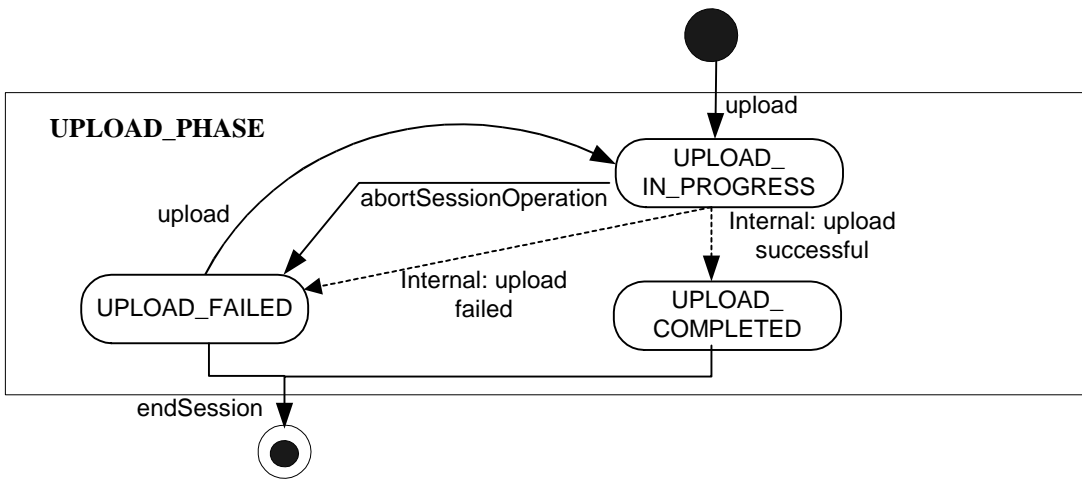


Figure 4: Substate Machine - UPLOAD_PHASE

~~Figure : Substate Machine – UPLOAD_PHASE~~

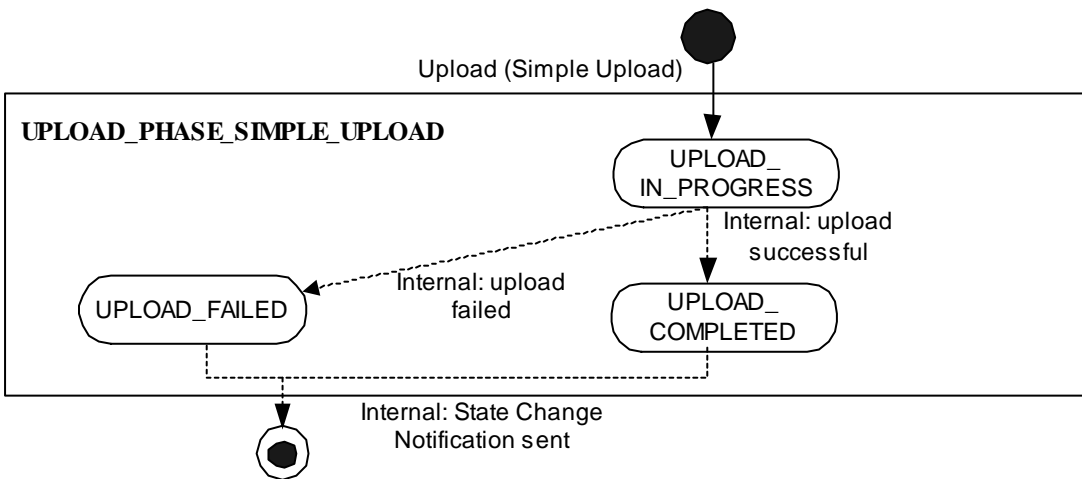


Figure 5: Substate Machine - UPLOAD_PHASE_SIMPLE_UPLOAD

~~Figure : Substate Machine – UPLOAD_PHASE_P1~~

9.2.2 Download Phase

When the download is triggered the IRP Agent copies the configuration data file (clause 10) from a given file area. The file is parsed and validated. If valid the state `DOWNLOAD_COMPLETED` is indicated. If the download fails a retry can be triggered in state `DOWNLOAD_FAILED`.

Once a session is associated with a download/validate/preactivate/activation behaviour then an upload phase cannot be triggered within this session.

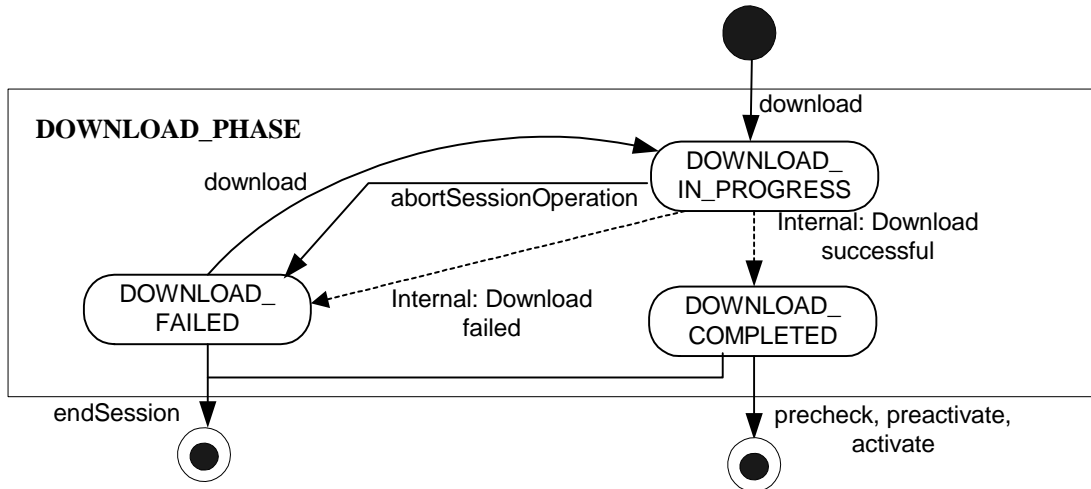


Figure 6: Substate Machine - DOWNLOAD_PHASE

Figure : Substate Machine – DOWNLOAD_PHASE

9.2.4 Validation Phase

After a download had been completed the configuration data can be semantically validated before being preactivated or activated into the real subnetwork of an IRP Agent. (see clause 7.5.6.2). A best effort strategy shall be applied. If validation was successful the state `VALIDATION_COMPLETED` is indicated. If the validate fails a retry can be triggered in state `VALIDATION_FAILED`.

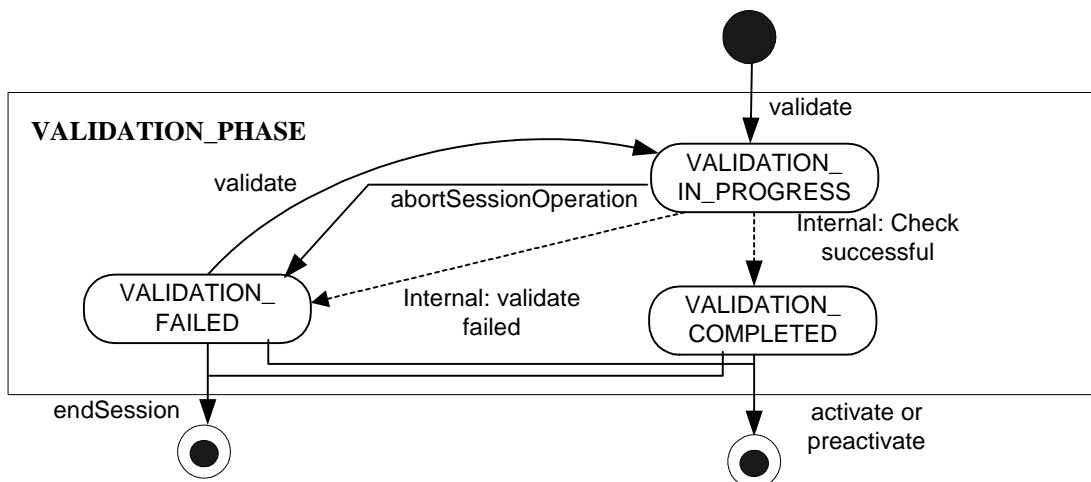


Figure 7: Substate Machine - VALIDATION_PHASE

Figure : Substate Machine – VALIDATION_PHASE

9.2.5 Preactivation Phase

After a download had been completed and optionally validated the configuration data can be preactivated before being activated into the real subnetwork of an IRPAgent. If the process fully succeeds the preactivation is completed.

For preactivation a best effort strategy shall be employed.

If the IRPAgent is unable to successfully complete all pre-MIB changes that were actioned in the configuration data file (clause10) the state PREACTIVATION_PARTLY_REALISED is indicated. This state is not an error condition because the preactivation of configuration data changes follows a best effort strategy. If the preactivation fails completely i.e. there are no pre-MIB changes the state PREACTIVATION_FAILED is indicated. A retry of the preactivate can be performed in states PREACTIVATION_PARTLY_REALISED and PREACTIVATION_FAILED. The PREACTIVATION_FAILED state cannot be entered if previously during the session the state had become PREACTIVATION_PARTLY_REALISED. The PREACTIVATION_PARTLY_REALISED state should be re-entered instead. A retry of the preactivate is allowed so that it is possible to recover after transient condition that caused a preactivate to fail or partly realise are no longer present.

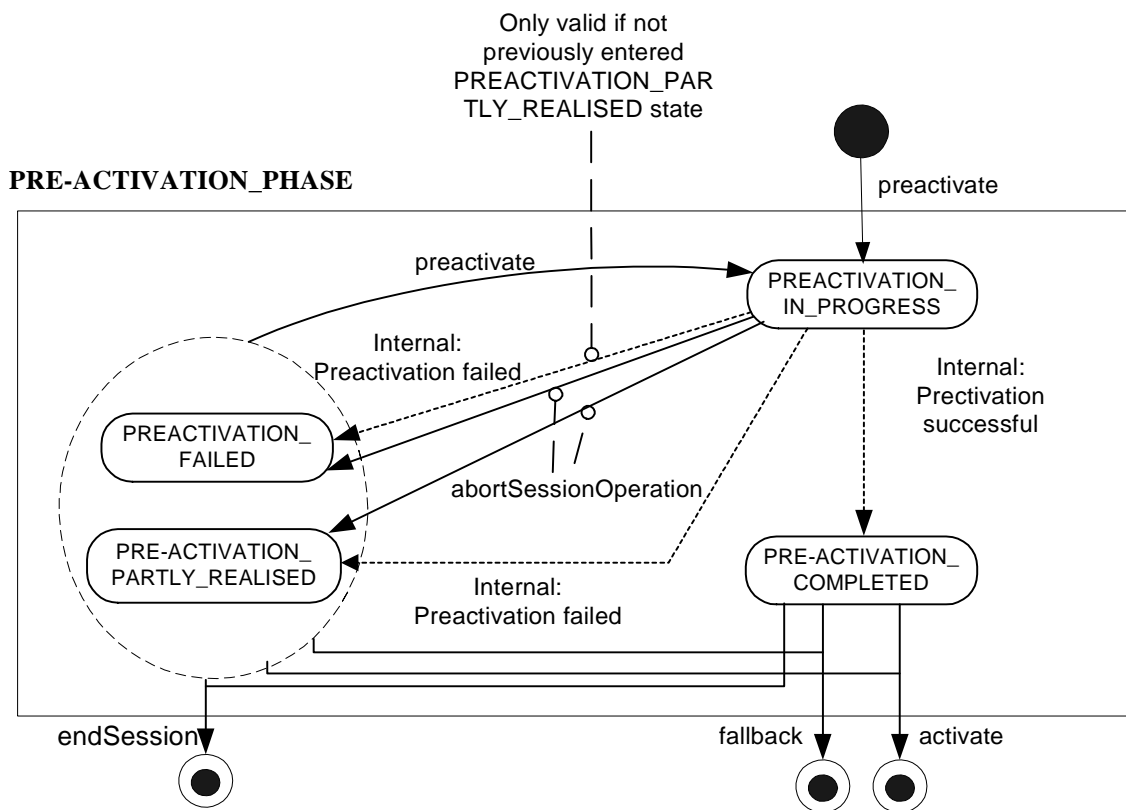


Figure 8: Substate Machine - PREACTIVATION_PHASE

Figure : Substate Machine – PREACTIVATION_PHASE

9.2.6 Activation Phase

After a download has been completed and optionally validated and/or preactivated the configuration data can be activated into the real subnetwork of an IRPAgent. If the process fully succeeds the activation is completed.

For activation a best effort strategy shall be employed.

If the IRPAgent is unable to successfully complete all MIB changes and corresponding changes in the network elements that were actioned in the configuration data file (clause 10) the state `ACTIVATION_PARTLY_REALISED` is indicated. This state is not an error condition because the activation of configuration data changes follows a best effort strategy. If the activate fails completely i.e. there are no MIB changes or corresponding changes in the network elements, the state `ACTIVATION_FAILED` is indicated. A retry of the activate can be performed in states `ACTIVATION_PARTLY_REALISED` and `ACTIVATION_FAILED`. The `ACTIVATION_FAILED` state cannot be entered if previously during the session the state had become `ACTIVATION_PARTLY_REALISED`. The `ACTIVATION_PARTLY_REALISED` state should be re-entered instead. A retry of the activate is allowed so that it is possible to recover after transient condition that caused an activate to fail or partly realise are no longer present.

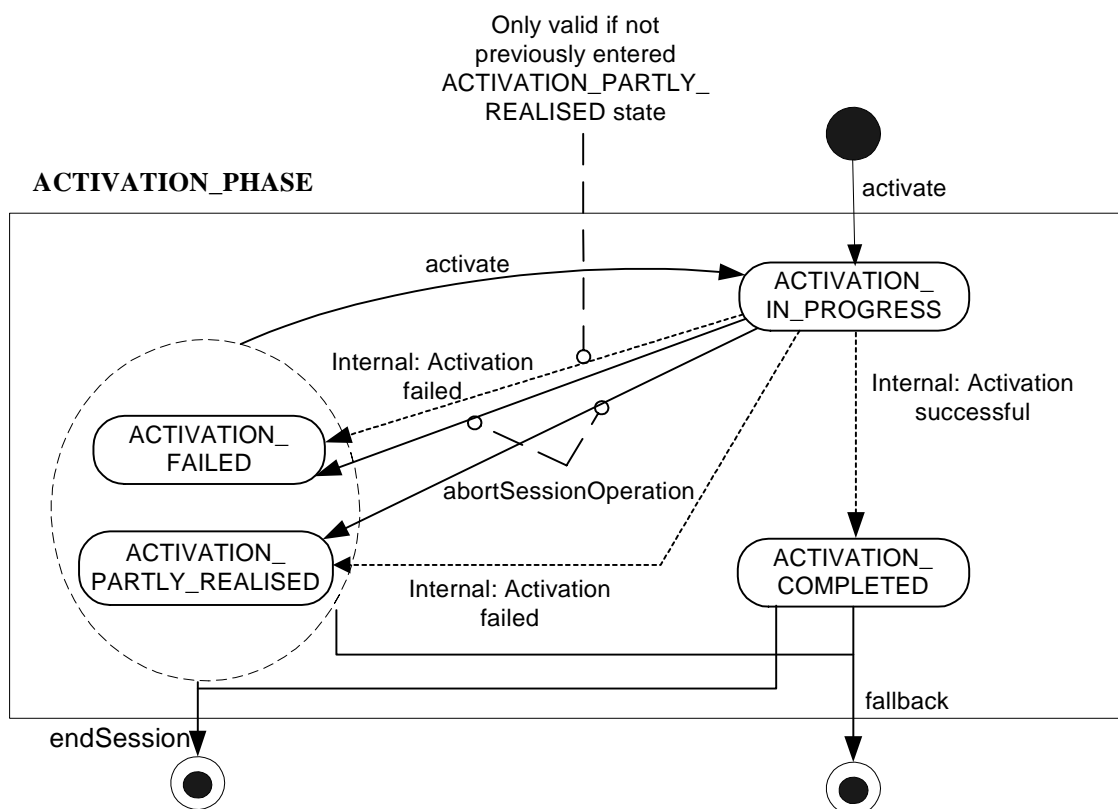


Figure 9: Substate Machine - ACTIVATION_PHASE

Figure : Substate Machine – ACTIVATION_PHASE

9.2.7 Fallback Phase

If an activate or preactivate operation was requested with the fallback option enabled and was successfully or partially completed then a fallback operation can be requested. If the process of a fallback fully succeeds then the related MIB and subnetwork is reverted back to its former configuration prior to first configuration data file preactivation or activation of a session.

For fallback a best effort strategy shall be employed.

In case that not all MIB changes and corresponding changes in the network elements that were actioned in configuration data file (clause 8) were successfully reverted back the state FALLBACK_PARTLY_REALISED is indicated. This state is not an error condition as the fallback to the former configuration follows a best effort strategy. If the fallback fails completely i.e. no MIB changes or corresponding changes in the network elements can be reverted back then the state FALLBACK_FAILED is indicated. A retry of fallback can be performed in the states FALLBACK_PARTLY_REALISED and FALLBACK_FAILED. The FALLBACK_FAILED state cannot be entered if previously during the session the state had become FALLBACK_PARTLY_REALISED. The FALLBACK_PARTLY_REALISED state should be re-entered instead. A retry of the fallback is allowed so that it is possible to recover after transient condition that caused a fallback to fail or partly realise are no longer present.

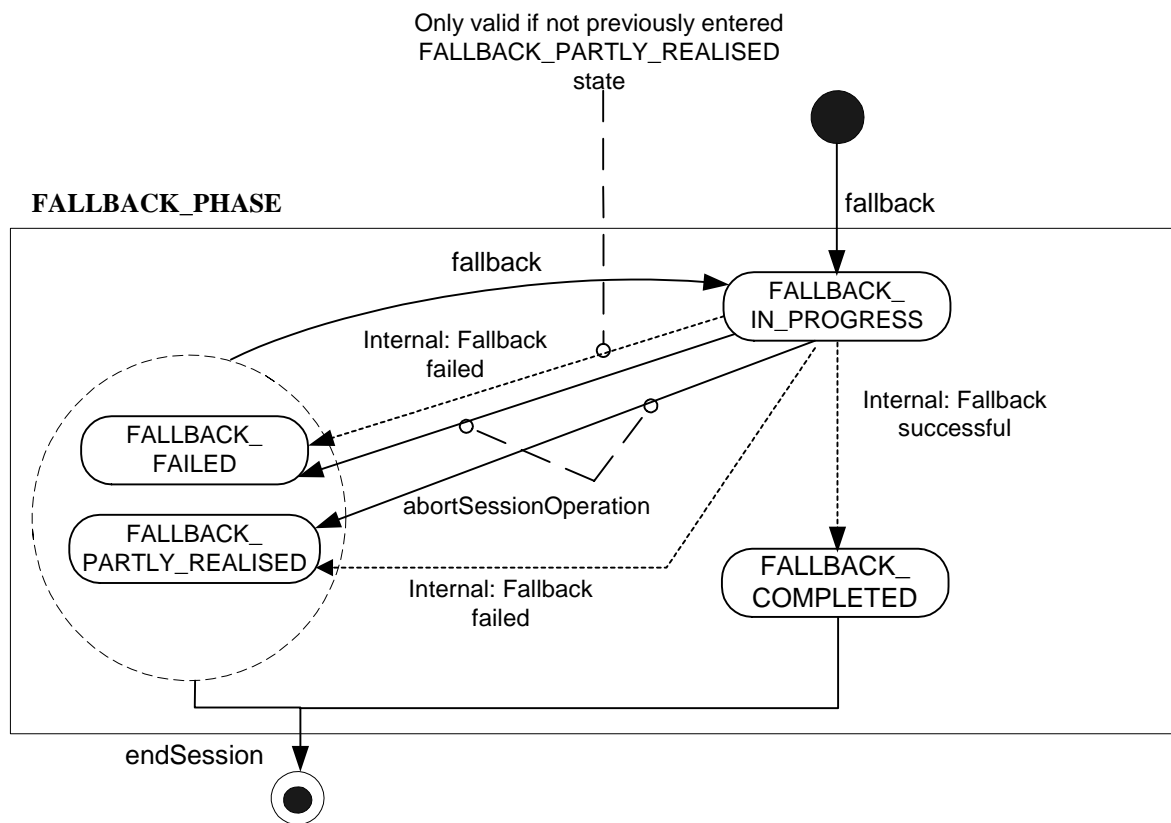


Figure 10: Substate Machine – FALLBACK_PHASE

Figure : Substate Machine – FALLBACK_PHASE

9.3 State Machine Pre and Post Conditions Tables

For each operation Table 9.1 identifies the state machine pre and post conditions.

Table 9.1: State Machine Pre and Post Conditions ([Controlled Upload and Controlled Upload & Provisioning](#))

Operation	Pre-condition	Post Condition
startSession	No state – input sessionId provided by an IRPManager is not already in use in the IRPAgent by this or any other IRPManager	State = IDLE
endSession	not in a Transition status i.e. state <>.*_IN_PROGRESS	sessionId is released - No state.
upload	State = IDLE or UPLOAD_FAILED	Initially while operation is being performed: State= UPLOAD_IN_PROGRESS Finally when operation has completed: State = UPLOAD_COMPLETED or UPLOAD_FAILED
download	State = IDLE or DOWNLOAD_FAILED	Initially while operation is being performed: State= DOWNLOAD_IN_PROGRESS Finally when operation has completed: State = DOWNLOAD_COMPLETED or DOWNLOAD_FAILED
validate	State = DOWNLOAD_COMPLETED or VALIDATION_FAILED	Initially while operation is being performed: State= VALIDATION_IN_PROGRESS Finally when operation has completed: State = VALIDATION_COMPLETED or VALIDATION_FAILED
preactivate	State = DOWNLOAD_COMPLETED or VALIDATION_COMPLETED or PREACTIVATION_PARTLY_REALISED or PREACTIVATION_FAILED	Initially while operation is being performed: State= PREACTIVATION_IN_PROGRESS Finally when operation has completed: State = PREACTIVATION_COMPLETED or PREACTIVATION_PARTLY_REALISED or PREACTIVATION_FAILED
activate	State = DOWNLOAD_COMPLETED or VALIDATION_COMPLETED or ACTIVATION_PARTLY_REALISED or ACTIVATION_FAILED or PREACTIVATION_COMPLETED or PREACTIVATION_PARTLY_REALISED or PREACTIVATION_FAILED	Initially while operation is being performed: State= ACTIVATION_IN_PROGRESS Finally when operation has completed: State = ACTIVATION_COMPLETED or ACTIVATION_PARTLY_REALISED or ACTIVATION_FAILED
fallback	State = PREACTIVATION_COMPLETED or PREACTIVATION_PARTLY_REALISED or ACTIVATION_COMPLETED or ACTIVATION_PARTLY_REALISED or FALLBACK_PARTLY_REALISED or FALLBACK_FAILED or FALLBACK_PARTLY_REALISED or FALLBACK_FAILED	Initially while operation is being performed: State= FALLBACK_IN_PROGRESS Finally when operation has completed: State = FALLBACK_COMPLETED or FALLBACK_PARTLY_REALISED or FALLBACK_FAILED
abortSessionOperation	State = UPLOAD_IN_PROGRESS or DOWNLOAD_IN_PROGRESS or VALIDATION_IN_PROGRESS or PREACTIVATION_IN_PROGRESS or ACTIVATION_IN_PROGRESS or FALLBACK_IN_PROGRESS	State = UPLOAD_FAILED or DOWNLOAD_FAILED or VALIDATE_FAILED or PREACTIVATION_PARTLY_REALISED or PREACTIVATION_FAILED or ACTIVATION_PARTLY_REALISED or ACTIVATION_FAILED or FALLBACK_PARTLY_REALISED or FALLBACK_FAILED
getSessionIds	N/A – State Machine independent	N/A
getSessionStatus	None	None
getSessionLog	None	None
getBulkCmIRPVersion	N/A – State Machine independent	N/A

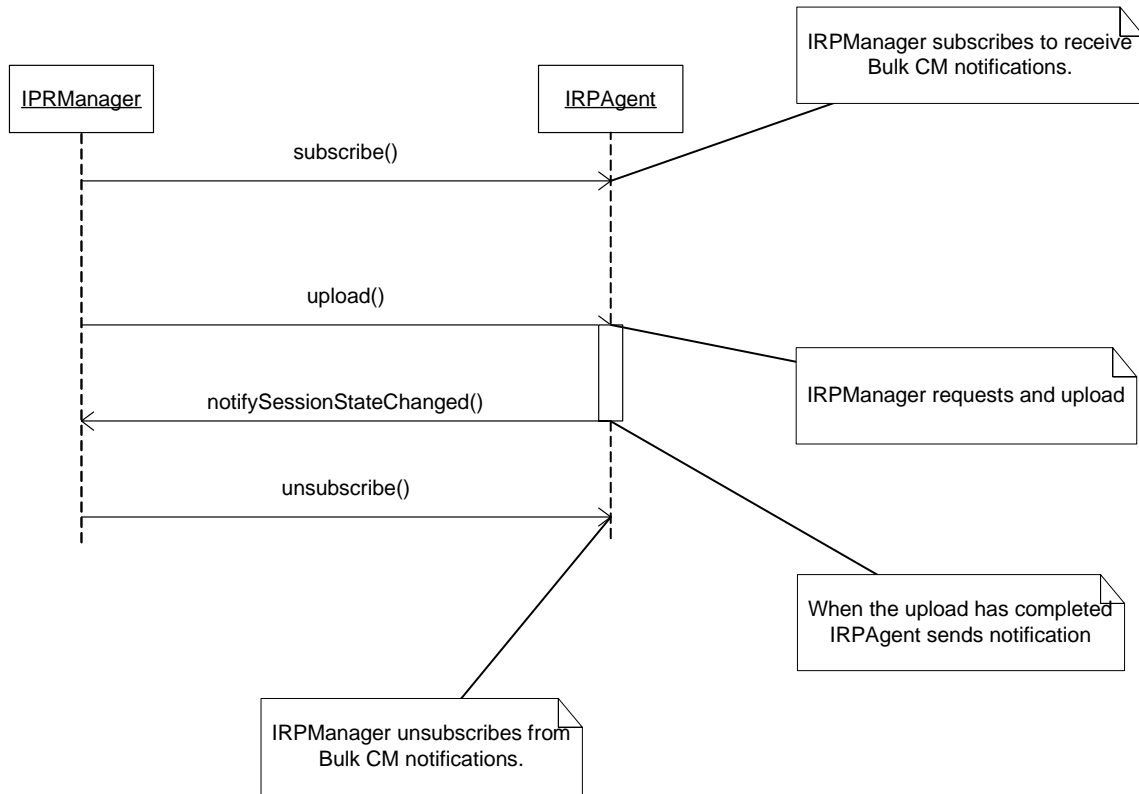
Table 9.2: State Machine Pre and Post Conditions (Simple Upload)

Operation	Pre-condition	Post Condition
<u>upload</u>	<u>No state – input sessionId provided by an IRPManager is not already in use in the IRPAgent by this or any other IRPManager</u>	<u>Initially while operation is being performed: State= UPLOAD_IN_PROGRESS When operation has completed: State = UPLOAD_COMPLETED or UPLOAD_FAILED until SessionStateChangeNotification sent then finally sessionId released and State becomes – no state</u>

Annex A (informative): Scenarios

Supporting background informational only.

...



[Figure A.6: Example 6: Successful Upload Session – Simple Upload](#)

CHANGE REQUEST

⌘ **32.612 CR 012** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ BulkCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP		
Source:	⌘ SA5 (wangrui@ln.chinamobile.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ B	Release:	⌘ Rel-6
	Use <u>one</u> 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 <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)

Reason for change:	⌘ Since Signalling Transport Network (STN) NRM IRP is introduced in R6, the scope of NRM IRP which can use BulkCMIRP needs to be extended.
Summary of change:	⌘ BulkCMIRP 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, 6.4								
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">Y</td> <td style="width: 20px;">N</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table> Other core specifications ⌘ Test specifications ⌘ O&M Specifications ⌘ 32.613, 32.615	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Y	N								
<input type="checkbox"/>	<input checked="" type="checkbox"/>								
<input type="checkbox"/>	<input checked="" type="checkbox"/>								
<input checked="" type="checkbox"/>	<input type="checkbox"/>								
Other comments:	⌘ Parent to Rel-6 CRs 32.613, 32.615 in S5-049021, S5-049023								

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.

- [1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".
- [2] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [3] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".
- [4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [5] 3GPP TS 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [6] 3GPP TS 32.652: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [7] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [8] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".
- [9] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".
- [10] 3GPP TS 32.632: "Telecommunication management; Configuration Management (CM); Core Network Resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [11] [3GPP TS 32.742: "Configuration Management \(CM\); Signalling Transport Network \(STN\) interface Network Resource Model \(NRM\)".](#)

End of Change in Clause 2

Change in Clause 6.4

6.4 Network Resource Model (NRM)

NRMs for Bulk CM IRP are defined in other Network Resource IRP documents of CM, For Bulk CM IRP IS these are:

- 32.622: "Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [4],
- 32.632: "Configuration Management (CM); Core Network Resources Integration Reference Point (IRP): Network Resource Model (NRM)" [10],

32.642: "Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [5],

32.652: "Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)" [6].

[32.742: "Configuration Management \(CM\); Signalling Transport Network \(STN\) interface Network Resource Model \(NRM\)" \[11\]](#)

These NRM documents define all the MOCs and attributes that can be configuration managed by Bulk CM IRP IS

**End of Change in Clause 6.4
End of the Document**

CHANGE REQUEST

⌘ **32.613 CR 010** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ Correct mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields - Align with IS in 32.612		
Source:	⌘ SA5 (robert.petersen@ericsson.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ F	Release:	⌘ Rel-6
	Use <u>one</u> 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 <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)

Reason for change:	⌘ Alignment with IS, i.e. correcting the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields).
Summary of change:	⌘ Place IS-defined non-filterable parameters into remaining_body of CORBA structured event.
Consequences if not approved:	⌘ The spec would not be aligned with the IS. Bulk CM IRP process wastes CPU filter CPU cycles on non-filterable parameters before emission of notification.

Clauses affected:	⌘ 4.4						
Other specs affected:	<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="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> Test 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="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> O&M Specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Y	N						
<input type="checkbox"/>	<input checked="" type="checkbox"/>						
Other comments:	⌘						

Change in Clause 4.4

4.4 Notification parameter mapping

Reference 3G TS 32.612 [3] defines semantics of parameters carried in notifications. The following tables indicate the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [6]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [6], is:

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

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the Bulk CM IRP: IS [3] defined notification parameters.

Table 17: Mapping from IS notifyGetSessionLogEnded parameters to SS equivalents

IS Parameter	OMG CORBA Structured Event Attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See sub-clause 3.3. It indicates the syntax and semantics of the Structure Event as defined by this specification.
notificationType	type_name	M	It carries the string NOTIFY_GET_SESSION_LOG_ENDED.
sessionLogStatus	event_name	M	It carries either the string GET_SESSION_LOG_COMPLETED_SUCCESSFUL GET_SESSION_LOG_COMPLETED_UNSUCCESSFUL In the case of the latter, the NV pair indicating ERROR_INFORMATION may be present.
There is no corresponding IS parameter	Variable Header		
managedObjectClass, managedObjectInstance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of pairs is not significant. The name of NV-pair is always encoded in string. Name of NV pair is the MANAGED_OBJECT_INSTANCE interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See encoding of this string [5]. These are attributes of Header defined in the IS.
notificationId	One NV pair of remaining_bodyfilterable_body_fields	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs Value of NV pair is a long. This is an attribute of Header defined in the IS.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs Value of NV pair is a IRPTime. This is an attribute of Header of the IS.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs Value of NV pair is a string. This is an attribute of Header defined in the IS.
sessionId	One NV pair of remaining_bodyfilterable_body_fields	M	Name of NV pair is the SESSION_ID of interface AttributeNameValue of module BulkCMIRPConstDefs Value of NV pair is a string.
sourceIndicator	One NV pair of remaining_bodyfilterable_body_fields	O	Name of NV pair is the SOURCE_INDICATOR of interface AttributeNameValue of module BulkCMIRPConstDefs Value of NV pair is a string.
There is no corresponding IS attribute.	One NV pair of remaining_bodyfilterable_body_fields	M	Name of NV pair is the ERROR_INFORMATION of interface AttributeNameValue of module BulkCMIRPConstDefs Value of NV pair is a string.

Table 18: Mapping from IS notifySessionStateChanged parameters to SS equivalents

IS Parameter	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute	domain_name	M	It carries the IRP document version number string. See sub-clause 3.3. It indicates the syntax and semantics of the Structured Event as defined by this specification.
notificationType	type_name	M	It carries the string NOTIFY_SESSION_STATE_CHANGED. This is an attribute of Header defined in the IS.
sessionState	event_name	M	It carries one of the following: UPLOAD_FAILED UPLOAD_COMPLETED, DOWNLOAD_FAILED, DOWNLOAD_COMPLETED, ACTIVATION_FAILED, ACTIVATION_PARTLY_REALISED, ACTIVATION_COMPLETED, FALLBACK_FAILED, FALLBACK_PARTLY_REALISED, FALLBACK_COMPLETED, VALIDATION_FAILED, VALIDATION_COMPLETED, Preactivation_FAILED, Preactivation_PARTLY_REALISED, Preactivation_COMPLETED In the case of XXX_FAILED and XXX_PARTLY_REALISED, the NV pair indicating ERROR_INFORMATION may be present.
There is no corresponding IS attribute	Variable Header		
managedObject Class, managedObject Instance	One NV pair of filterable_body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See encoding of this string in [5]. These are attributes of Header defined in the IS.
notificationId	One NV pair of remaining_bodyfilterable_body_fields	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. This is an attribute of Header defined in the IS.
eventTime	One NV pair of filterable_body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a IRPTime. This is an attribute of Header of the IS.
systemDN	One NV pair of filterable_body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. This is an attribute of Header defined in the IS.
sessionId	One NV pair of remaining_bodyfilterable_body_fields	M	Name of NV pair is the SESSION_ID of interface AttributeNameValue of module BulkCMIRPConstDefs. Value of NV pair is a string.
sourceIndicator	One NV pair of remaining_bodyfilterable_body_fields	O	Name of NV pair is the SOURCE_INDICATOR of interface AttributeNameValue of module BulkCMIRPConstDefs. Value of NV pair is a string.
There is no corresponding IS attribute.	One NV pair of remainingfilterable_body_fields	<u>M</u>	Name of NV pair is the ERROR_INFORMATION of interface AttributeNameValue of module BulkCMIRPConstDefs. Value of NV pair is a string.

End of Change in Clause 4.4
End of Document

CHANGE REQUEST

⌘ **32.613 CR 011** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ Partition Bulk CM IRP capabilities into separate IDL modules – Align to IS in 32.612		
Source:	⌘ SA5 (edwin.tse@ericsson.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ F	Release:	⌘ Rel-6
	Use <u>one</u> 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 <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)

Reason for change:	⌘ To allow vendor to implement a 3GPP compliant system that supports only the capability of uploading of configuration management (CM) and inventory management data (IM).
Summary of change:	⌘ Partition the Bulk CM IRP capabilities into separate IDL modules.
Consequences if not approved:	⌘ It is impossible for vendor to implement a 3GPP compliant system that (a) supports the capability of uploading CM-data and IM-data and (b) does not support the complex active configuration management (download) capabilities.

Clauses affected:	⌘ 1, 2, 3, 4, Annex A										
Other specs affected:	<table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">Y</td> <td style="width: 20px;">N</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </table> Other core specifications Test specifications O&M Specifications	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	⌘	
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
Other comments:	⌘ Child to 32.612 CR in S5-047109										

1 Scope

The purpose of this *Bulk CM IRP: CORBA Solution Set* is to define the mapping of the IRP information service (see 3GPP TS 32.612 [3]) to the protocol specific details necessary for implementation of this IRP in a CORBA/IDL environment.

~~The present document does not describe any Network Resource Model (NRM)—they are described in Generic Network Resources IRP: NRM 3GPP TS 32.622 [4], UTRAN Network Resources IRP: NRM 3GPP TS 32.642 [11], GERAN Network Resources IRP: NRM 3GPP TS 32.652 [12].~~

This Solution Set specification is related to 3GPP TS 32.612 V6.10.X.

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 32.612: "Telecommunication management; Configuration Management (CM); Bulk CM Integration Reference Point (IRP); Information Service (IS)".
- [4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [5] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [6] OMG Notification Service, Version 1.0.
- [7] OMG CORBA services: Common Object Services Specification, Update: November 22, 1996.
- [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 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [12] 3GPP TS 32.652: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [13] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[14] [3GPP TS 32.692 “Inventory Management \(IM\) network resources Integration Reference Point \(IRP\): Network Resource Model \(NRM\)”](#).

3 Definitions and abbreviations

3.1 Definitions

For terms and definitions please refer to 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.612 [3], 3GPP TS 32.622 [4], 3GPP TS 32.642 [11], ~~and~~ 3GPP TS 32.652 [12] ~~and~~ [3GPP TS 32.692 \[14\]](#).

- IRP document version number string (or "IRPVersion"): See 3GPP TS 32.312 [13].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CORBA	Common Object Request Broker Architecture
DN	Distinguished Name
IS	Information Service
IDL	Interface Definition Language (OMG)
IRP	Integration Reference Point
MO	Managed Object
MOC	Managed Object Class
NRM	Network Resource Model
OMG	Object Management Group
SS	Solution Set

~~3.3 IRP document version number string~~

~~The IRP document version number (sometimes called "IRPVersion" or "version number") string is used to identify this specification. The string is derived using a rule described in definition "IRP document version number string".~~

~~This string is returned in `getBulkCmIRPVersion` method and is carried in the first field of the notification header of all notifications related to this IRP.~~

~~Take the 3GPP document number on the front page of this specification, such as "3GPP TS 32.613 V5.0.0 (2002-09)". Discard the leading "3GPP TS ". Discard all characters after and including the last period. Eliminate leading and trailing spaces. Reduce multiple consecutive spaces with one space. Express the resultant in a string. Capitalised the string. For example, if the 3GPP document version number is "3GPP TS 32.613 V5.0.0 (2002-09)", then the IRP document version number shall be "32.613 V5.0".~~

4 Mapping

..

4.2 Operation and Notification mapping

The IS part of Bulk CM: IRP defines semantics of operations and notifications visible across the Bulk Configuration IRP. The table below indicates mapping of these operations and notifications to their equivalents defined in this document.

[There are 3 qualifications for each row of the following mapping table. The 3 qualifications correspond to the three IS-defined packages: Controlled Upload & Provisioning, Controlled Upload and Simple Upload. Not all operations/notifications specified in the following table are required for all 3 packages. An '-' indicates that the subject operation or notification is not allowed by that corresponding package.](#)

Table 1: Mapping from IM Notification/Operation to SS equivalent

IS Operation/ notification	SS Method	Qualifier
startSession	start_session	M,M,-
endSession	end_session	M,M,-
upload	upload	M,M,M
download	download	M,-,-
activate	activate	M,-,-
getSessionStatus	get_session_status	M,M,-
getSessionIds	get_session_ids	M,M,-
getSessionLog	get_session_log	M,M,-
fallback	fallback	M,-,-
abortSessionOperation	abort_session_operation	M,M,-
getIRPVersion	get_bulk_CM_IRP_versions get_controlled_upload_bulk_CM_IRP_versions get_simple_upload_bulk_CM_IRP_versions	M,-,- -,M,- -,-M
notifySessionStateChanged	push_structured_event Note that OMG Notification Service OMG Notification Service [1] defines this method. See clause 5.1	M,M,M
notifyGetSessionLogEnded	push_structured_event Note that OMG Notification Service OMG Notification Service [1] defines this method. See clause 5.1.	M,M,-
preactivate	preactivate	O,-,-
validate	validate	O,-,-
getOperationProfile	get_bulk_CM_IRP_operation_profile get_controlled_upload_bulk_CM_IRP_operation_profile get_simple_upload_bulk_CM_IRP_operation_profile	O,-,- -,O,- -,-O
getNotificationProfile	get_bulk_CM_IRP_notification_profile get_controlled_upload_bulk_CM_IRP_notification_profile get_simple_upload_bulk_CM_IRP_notification_profile	O,-,- -,O,- -,-O

4.3 Operation Parameter Mapping

Reference Bulk CM IRP; Information Service [3] defines semantics of parameters carried in operations. The tables below indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table 2: Mapping from IS startSession parameters to SS equivalents

IS Operation parameter	SS parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
status	exception StartSessionException, exception SessionIdInUseException, exception MaxSessionReachedException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 3: Mapping from IS endSession parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
status	exception EndSessionException, exception UnknownSessionIdException, exception NotValidInCurrentStateException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 4: Mapping from IS upload parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
uploadDataFile Reference	BulkCmIRPConstDefs::FileDestination sink	M
baseObjectInstance	BulkCmIRPConstDefs::DistinguishedName base_object	M
scope, filter	BulkCmIRPConstDefs::SearchControl search_control	M
status	exception UploadException, exception UnknownSessionIdException, exception MaxSessionReachedException, exception NotValidInCurrentStateException, exception ConcurrencyException, exception IllegalDNFormatException, exception IllegalFilterFormatException, exception IllegalScopeTypeException, exception IllegalScopeLevelException, exception IllegalURLFormatException, exception ManagedGenericIRPSystem::InvalidParameter	M
NOTE:	The IllegalURLFormatException does not imply that the transfer protocol used must be a URL. The transfer protocol is dependant on the file format definition, i.e. in the case of XML, FileDestination will be a URL.	

Table 5: Mapping from IS download parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
downloadDataFileReference	BulkCmIRPConstDefs::FileDestination source	M
status	exception DownloadException, exception UnknownSessionIdException, exception MaxSessionReachedException, exception NotValidInCurrentStateException, exception IllegalURLFormatException, exception ManagedGenericIRPSystem::InvalidParameter	M
NOTE:	The IllegalURLFormatException does not imply that the transfer protocol used must be a URL. The transfer protocol is dependant on the file format definition, i.e. in the case of XML, FileDestination will be a URL.	

Table 6: Mapping from IS activate parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPCConstDefs::SessionId session_id	M
activationMode	BulkCmIRPCConstDefs::ActivationModeTypeOpt activation_mode	O
fallbackEnabled	boolean fallback	M
status	exception ActivateException, exception UnknownSessionIdException, exception NotValidInCurrentStateException, exception ConcurrencyException, exception IllegalActivationModeException, exception ManagedGenericIRPSystem::ParameterNotSupported, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 7: Mapping from IS fallback parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPCConstDefs::SessionId session_id	M
status	exception FallbackException, exception UnknownSessionIdException, exception NoFallbackException, exception NotValidInCurrentStateException, exception ConcurrencyException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 8: Mapping from IS abortSessionOperation parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPCConstDefs::SessionId session_id	M
status	exception AbortSessionOperationException, exception UnknownSessionIdException, exception NotValidInCurrentStateException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 9: Mapping from IS getSessionIds parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionIdList	return of type BulkCmIRPCConstDefs::SessionIdList	M
status	exception GetSessionIdsException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 10: Mapping from IS getSessionStatus parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPCConstDefs::SessionId session_id	M
sessionState	return of type BulkCmIRPCConstDefs::SessionState	M
Not specified in IS	BulkCmIRPCConstDefs::ErrorInformation error_information	M
status	exception GetSessionStatusException, exception UnknownSessionIdException, exception ManagedGenericIRPSystem::InvalidParameter	M

Table 11: Mapping from IS getSessionLog parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
logFileReference	BulkCmIRPConstDefs::FileDestination sink	M
contentType	boolean only_error_info	M
status	exception GetSessionLogException, exception UnknownSessionIdException, exception IllegalURLFormatException, exception ManagedGenericIRPSystem::InvalidParameter	M
NOTE: The IllegalURLFormatException does not imply that the transfer protocol used must be a URL. The transfer protocol is dependant on the file format definition, i.e. in the case of XML, FileDestination will be a URL.		

Table 12: Mapping from IS getBulkCmIRPVersion parameters to SS equivalents

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

Table 13: Mapping from IS validate parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
activationMode	BulkCmIRPConstDefs::ActivationModeTypeOpt activation_mode	O
status	exception ValidateException , exception UnknownSessionIdException, exception NotValidInCurrentStateException, exception ConcurrencyException, exception IllegalActivationModeException, exception ManagedGenericIRPSystem::ParameterNotSupported, exception ManagedGenericIRPSystem::InvalidParameter, exception ManagedGenericIRPSystem::OperationNotSupported	M

Table 14: Mapping from IS preactivate parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
sessionId	BulkCmIRPConstDefs::SessionId session_id	M
verificationMode	BulkCmIRPConstDefs::VerificationModeTypeOpt verification_mode	O
activationMode	BulkCmIRPConstDefs::ActivationModeTypeOpt activation_mode	O
fallbackEnabled	boolean fallback	M
status	exception PreactivateException, exception UnknownSessionIdException, exception NotValidInCurrentStateException, exception ConcurrencyException, exception IllegalActivationModeException, exception IllegalVerificationModeException, exception ManagedGenericIRPSystem::ParameterNotSupported, exception ManagedGenericIRPSystem::InvalidParameter, exception ManagedGenericIRPSystem::OperationNotSupported	M

Table 15: Mapping from IS getOperationProfile parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
irpVersion	ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version	M
operationNameProfile, operationParameterProfile	Return value of type ManagedGenericIRPConstDefs::MethodList	M
status	Exceptions: GetBulkCMIRPOperationProfileException, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter	M

Table 16: Mapping from IS getNotificationProfile parameters to SS equivalents

IS Operation parameter	SS Method parameter	Qualifier
irpVersion	ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version	M
notificationNameProfile, notificationParameterProfile	Return value of type ManagedGenericIRPConstDefs::MethodList	M
status	Exceptions: GetBulkCMIRPNotificationProfileException, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter	M

...

4.7 Package Mapping

[The Bulk CM IRP: IS \[3\] clause 7 specifies packages of capabilities. The IS-defined packages are mapped into IDL module constructs. Specifically:](#)

- [The operations named in the IS-defined packages Simple Upload, Controlled Upload and Controlled Upload & Provisioning are mapped to methods in SimpleUploadBulkCMIRSystem::SimpleUploadBulkCMIRP, ControlledUploadBulkCMIRPSystem::ControlledUploadBulkCMIRP and BulkCmIRPSystem::BulkCmIRP respectively. See A.2.](#)
- [The notifications named in the IS-defined Simple Upload, Controlled Upload and Controlled Upload & Provisioning are mapped to SS Interfaces defined in IDL module BulkCMIRPNotifications. See A.3.](#)

5 BulkCMIRPNotifications Interface

OMG CORBA Notification push operation is used to realise the notification of BulkCMIRPNotifications. All the notifications in this interface are implemented using this push_structured_event method.

5.1 Method push (M)

```

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

```

```
    ...  
}; // SequencePushConsumer  
  
    ...  
}; // CosNotifyComm
```

NOTE 1: The `push_structured_events` method takes an input parameter of type `EventBatch` as defined in the OMG `CosNotification` module (OMG Notification Service [6]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to `IRPManager` by `IRPAgent` to which it is connected.

NOTE 2: The maximum number of events that will be transmitted within a single invocation of this operation is controlled by `IRPAgent` wide configuration parameter.

NOTE 3: The amount of time the supplier (`IRPAgent`) of a sequence of Structured Events will accumulate individual events into the sequence before invoking this operation is controlled by `IRPAgent` wide configuration parameter as well.

NOTE 4: `IRPAgent` may push `EventBatch` with only one Structured Event.

Annex A (normative): IDL Specifications

A.1 IDL specification (file name :-BulkCmIRPConstDefs.idl)

// File: BulkCmIRPConstDefs.idl

```
#ifndef BULKCMIRPCONSTDEFS_IDL BulkCmIRPConstDefs_IDL
#define BULKCMIRPCONSTDEFS_IDL BulkCmIRPConstDefs_IDL

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

/* ## Module: BulkCmIRPConstDefs
This module contains type definitions for the Bulk CM IRP
=====
*/
module BulkCmIRPConstDefs
{

    /*
    This block identifies the notification types defined by
    this Bulk CM IRP version.
    This string is used in the second field of the Structured
    Event.
    */
    interface NotificationType
    {
        const string NOTIFY_SESSION_STATE_CHANGED = "x1";
        const string NOTIFY_GET_SESSION_LOG_ENDED = "x2";
    };

    /*
    This block assigns value for the name of the NV of the Structured Event.
    */
    interface AttributeNameValue
    {
        const string SESSION_ID = "k";
        const string SOURCE_INDICATOR = "m";
        const string ERROR_INFORMATION = "n";
    };

    /*
    This block defines all possible values for sessionState.
    One of these strings appear in the event_name of the
    Structured Event of notifySessionStateChanged notification.
    */
    interface SessionStateChangeNotification
    {
        const string UPLOAD_FAILED = "x1";
        const string UPLOAD_COMPLETED = "x2";
        const string DOWNLOAD_FAILED = "x3";
        const string DOWNLOAD_COMPLETED = "x4";
        const string ACTIVATION_FAILED = "x5";
        const string ACTIVATION_PARTLY_REALISED = "x6";
        const string ACTIVATION_COMPLETED = "x7";
    };
};
```

```

    const string FALLBACK_FAILED = "x8";
    const string FALLBACK_PARTLY_REALISED = "x9";
    const string FALLBACK_COMPLETED = "x10";
    const string VALIDATION_FAILED = "x11";
    const string VALIDATION_COMPLETED = "x12";
    const string PREACTIVATION_FAILED = "x13";
    const string PREACTIVATION_PARTLY_REALISED = "x14";
    const string PREACTIVATION_COMPLETED = "x15";
};

/*
This block defines all possible values for sessionLogStatus
One of these strings appear in the event_name of the Structured
Event of notifyGetSessionLogEnded notification.
*/
interface LogStateNotification
{
    const string GET_SESSION_LOG_COMPLETED_SUCCESSFULLY = "x1";
    const string GET_SESSION_LOG_COMPLETED_UNSUCESSFULLY = "x2";
};

/*
For each started configuration session a unique identifier is generated
by the IRPManager. An sessionId can not be used for an upload if it is
already in use of a download configuration and vice versa.
*/
typedef string SessionId;

/*
This string field is used in order to provide additional error information
if an operation has failed.
*/
typedef string ErrorInformation;

/*
Defines the different subphases of a configuration session
e.g. thus it is easy to implement a detection of an upload
or a download/activate session.
*/
enum SubPhase {IdlePhase, DownloadPhase, UploadPhase, ActivationPhase,
                FallbackPhase, PreactivationPhase, ValidationPhase};

/*
Defines the different substates of a configuration session. This includes
the transition state as well.
*/
enum SubState {Completed, Failed, PartlyRealised, InProgress};

/*
Defines state of a configuration session with the phase and the substate
of the configuration.
*/
struct SessionState
{
    SubPhase sub_phase;
    SubState sub_state;
};

/*
Contains the list of all current sessionIds
*/
typedef sequence <BulkCmIRPConstDefs::SessionId> SessionIdList;

```

```

/*
Specifies a complete destination path (including filename).
*/
typedef string FileDestination;

/*
The format of Distinguished Name is specified in
the Naming Conventions for Managed Objects; 3GTS 32.300-Annex H.
e.g. "SubNetwork=10001,ManagedElement=400001" identifies an
G3ManagedElement instance of the object model.
*/
typedef string DistinguishedName;

/*
Used within the upload method to give filter criteria
*/
typedef string FilterType;

/*
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.
*/
enum ScopeType {BaseOnly, BaseNthLevel, BaseSubtree, BaseAll};

/*
Controls the searching for MOs during upload, and contains:
the type of scope ("type" field),
the level of scope ("level" field),
the filter ("filter" field),
The type and level fields are mandatory.
The filter field is mandatory (The filter will have to be
set to an empty string if it has no other value).
*/
struct SearchControl
{
    ScopeType type;
    unsigned long level;
    FilterType filter;
};

/*
This indicates how the activation is executed, either with least service
impact or least elapsed time.
*/
enum ActivationMode {LeastServiceImpact, LeastElapsedTime};

/*
This indicates the level of verification of bulk configuration data done,
either full or limited checking.
*/
enum VerificationMode {FullChecking, LimitedChecking};

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

/* VerificationModeTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, the value is present.

```

```
    Otherwise, the value is absent.  
*/  
union VerificationModeTypeOpt switch(boolean)  
{  
    case TRUE: VerificationMode verification_mode;  
};  
  
};  
  
#endif
```

A.2 Annex B (normative): IDL specification (file Name: BulkCmIRPSystem.idl)

```
// File: BulkCmIRPSystem.idl

#ifndef BULKCMIRPSYSTEM_IDL BulkCmIRPSystem_IDL
#define BULKCMIRPSYSTEM_IDL BulkCmIRPSystem_IDL

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

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

/* ## Module: BulkCmIRPSystem
This module implements capabilities of Bulk CM IRP.
=====
*/
module BulkCmIRPSystem
{
    /*
    The request cannot be processed due to a situation of concurrency.
    E.g. two concurrent activation requests involving the same ManagedElement
    instance. The semantics carried in reason is outside the scope of this IRP.
    */
    exception ConcurrencyException { string reason; };

    /*
    The provided filter is malformed or invalid. The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception IllegalFilterFormatException { string reason; };

    /*
    The provided Distinguished Name is malformed or invalid. The semantics
    carried in reason is outside the scope of this IRP.
    */
    exception IllegalDNFormatException { string reason; };

    /*
    The provided scope type is illegal. The semantics carried in reason is
    outside the scope of this IRP.
    */
    exception IllegalScopeTypeException { string reason; };

    /*
    The provided scope level is illegal. The semantics carried in reason is
    outside the scope of this IRP.
    */
    exception IllegalScopeLevelException { string reason; };

    /*
    The request cannot be processed because no fallback data is available, i.e.
    fallback capability was previously not asked for.
    */
    exception NoFallbackException {};

    /*
    The provided sessionId value is already used for another configuration

```

```

session. The semantics carried in reason is outside the scope of this IRP.
*/
exception SessionIdInUseException { string reason; };

/*
The provided URL is malformed or invalid. The semantics carried in reason is
outside the scope of this IRP.
*/
exception IllegalURLFormatException{ string reason; };

/*
The provided sessionId value does not identify any existing configuration
session.
*/
exception UnknownSessionIdException {};

/*
The request cannot be processed because it is not valid in the current state
of the configuration session.
*/
exception NotValidInCurrentStateException
{
    BulkCmIRPConstDefs::SessionState current_state;
};

/*
The request cannot be processed because the maximum number of simultaneously
running configuration sessions has been reached. The semantics carried in
reason is outside the scope of this IRP.
*/
exception MaxSessionReachedException { string reason; };

/*
The provided ActivationMode type is illegal. The semantics carried in reason
is outside the scope of this IRP.
*/
exception IllegalActivationModeException { string reason; };

/*
The provided VerificationMode type is illegal. The semantics carried in
reason is outside the scope of this IRP.
*/
exception IllegalVerificationModeException { string reason; };

/*
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 GetBulkCmIRPVersionsException { string reason; };
exception UploadException { string reason; };
exception DownloadException { string reason; };
exception ActivateException { string reason; };
exception ValidateException { string reason; };
exception PreactivateException { string reason; };
exception GetBulkCMIRPOperationProfileException { string reason; };
exception GetBulkCMIRPNotificationProfileException { string reason; };
exception GetSessionLogException { string reason; };
exception StartSessionException { string reason; };
exception GetSessionStatusException { string reason; };
exception FallbackException { string reason; };
exception EndSessionException { string reason; };
exception AbortSessionOperationException { string reason; };

```



```

exception GetSessionIdsException { string reason; };

/*
Defines the System interface of a EM. It defines all methods which are
necessary to control a configuration session from a IRPManager.
*/
interface BulkCmIRP
{
    /*
    Return the list of all supported Bulk CM IRP versions.
    */
    ManagedGenericIRPConstDefs::VersionNumberSet get_bulk_CM_IRP_versions (
    )
    raises (GetBulkCmIRPVersionsException);

    /*
    Return the list of all supported operations and their supported
    parameters for a specific BulkCM IRP version.
    */
    ManagedGenericIRPConstDefs::MethodList get_bulk_CM_IRP_operation_profile (
        in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version
    )
    raises (GetBulkCMIRPOperationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

    /*
    Return the list of all supported notifications and their supported
    parameters for a specific BulkCM IRP version.
    */
    ManagedGenericIRPConstDefs::MethodList
        get_bulk_CM_IRP_notification_profile
    (
        in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version
    )
    raises (GetBulkCMIRPNotificationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

    /*
    Uploads a configuration from the subnetwork. The result is put in a
    configuration data file in an area specified by the IRPManager.
    The MIB of the subnetwork is iterated by means of containment search,
    using a SearchControl to control the search and the returned results.
    All MOs in the scope constitutes a set that the filter works on.
    In case of a concurrent running session the function will
    return an exception. If the value of the given baseObject or FilterType
    does not exist then this asynchronous error condition will be notified.
    */
    void upload (
        in BulkCmIRPConstDefs::SessionId session_id,
        in BulkCmIRPConstDefs::FileDestination sink,
        in BulkCmIRPConstDefs::DistinguishedName base_object,
        in BulkCmIRPConstDefs::SearchControl search_control
    )
    raises (UploadException, UnknownSessionIdException,
        MaxSessionReachedException, NotValidInCurrentStateException,
        ConcurrencyException,
        IllegalDNFormatException, IllegalFilterFormatException,
        IllegalScopeTypeException, IllegalScopeLevelException,
        IllegalURLFormatException,
        ManagedGenericIRPSystem::InvalidParameter);
}

```

```

/*
Indicates the EM that it can download a configuration data file from
a given configuration data file storage area. The EM will check the
consistence of the configuration data and the software compatibilty.
*/
void download (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::FileDestination source
)
raises (DownloadException, UnknownSessionIdException,
        MaxSessionReachedException, NotValidInCurrentStateException,
        IllegalURLFormatException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Activates a previously downloaded and sucessfully parsed configuration
inside a session. This means that the configuration will be introduced
in the live sub-network. In case of a concurrent running session
the function will return an exception.
*/
void activate (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::ActivationModeTypeOpt activation_mode,
    in boolean fallback
)
raises (ActivateException, UnknownSessionIdException,
        NotValidInCurrentStateException, ConcurrencyException,
        IllegalActivationModeException,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Uploads a log from the subnetwork which is usally used for error
analysis. The log is put in a logfile in the filesystem which can
be accessed by the EM. If there are no log entries an empty log file
is uploaded.
*/
void get_session_log (
    in BulkCmIRPConstDefs::FileDestination sink,
    in BulkCmIRPConstDefs::SessionId session_id,
    in boolean only_error_info
)
raises (GetSessionLogException, UnknownSessionIdException,
        IllegalURLFormatException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Creates an instance of the configuration session state machine. The
IDLE_PHASE & COMPLETED is notified
*/
void start_session (
    in BulkCmIRPConstDefs::SessionId session_id
)
raises (StartSessionException, SessionIdInUseException,
        MaxSessionReachedException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Returns the state of a configuration session.
*/
BulkCmIRPConstDefs::SessionState get_session_status (
    in BulkCmIRPConstDefs::SessionId session_id,
    out BulkCmIRPConstDefs::ErrorInformation error_information

```

```

)
raises (GetSessionStatusException, UnknownSessionIdException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Activates a fallback area. Each time a configuration is activated a
fallback area can be created, s. activate parameter.
This area is backup of the complete configuration which can be
restored by this method. The process is as follows:
1. When the method activate(..., ..., TRUE) is used,
   a copy of the valid area is taken before the activation
   of the new planned data has started. Only one fallback area can
   exist at a time for a specific scope of the subnetwork.
2. When a fallback area is available and triggered by this method, the
   previous valid area is replaced with the data stored in
   the fall back area.
If the EM detects that the former configuration has never been
changed it returns an exception because it does not trigger an
activation of the former data.
*/
void fallback (
    in BulkCmIRPConstDefs::SessionId session_id
)
raises (FallbackException, UnknownSessionIdException, NoFallbackException,
        NotValidInCurrentStateException, ConcurrencyException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
The IRPManager invokes this operation to delete all its temporary
entities and the related sessionId which belong to the scope of
a configuration session. This includes the related error and log
informationen too.
*/
void end_session (
    in BulkCmIRPConstDefs::SessionId session_id
)
raises (EndSessionException, UnknownSessionIdException,
        NotValidInCurrentStateException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
The IRPManager invokes this operation to abort an active operation
during a configuration session. It is only effecting
a configuration session in state IN_PROGRESS. In this case the
current session task is interrupted, e.g. the activating in progress,
using best effort strategy, and a state change is notified
*/
void abort_session_operation (
    in BulkCmIRPConstDefs::SessionId session_id
)
raises (AbortSessionOperationException, UnknownSessionIdException,
        NotValidInCurrentStateException,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Returns a list all sessionIds of current running configuration sessions.
*/
BulkCmIRPConstDefs::SessionIdList get_session_ids (
)
raises (GetSessionIdsException);

/*
Validates previously downloaded bulk configuration data inside a session.

```

```

Detects errors in the data prior to requesting preactivation or
activation.
*/
void validate (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::ActivationModeTypeOpt activation_mode
)
raises (ValidateException, UnknownSessionIdException,
        NotValidInCurrentStateException, ConcurrencyException,
        IllegalActivationModeException,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        ManagedGenericIRPSystem::OperationNotSupported);

/*
Preactivates previously downloaded bulk configuration data inside a
session. This operation validates configuration data changes in the
context of the current data and pre-processes the configuration data
changes.
*/
void preactivate (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::VerificationModeTypeOpt verification_mode,
    in BulkCmIRPConstDefs::ActivationModeTypeOpt activation_mode,
    in boolean fallback
)
raises (PreactivateException, UnknownSessionIdException,
        NotValidInCurrentStateException, ConcurrencyException,
        IllegalActivationModeException, IllegalVerificationModeException,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter,
        ManagedGenericIRPSystem::OperationNotSupported);
};
};

```

```

module SimpleUploadBulkCMIRPSystem
{
    exception GetSimpleUploadBulkCmIRPVersionsException { string reason; };
    exception GetSimpleUploadBulkCMIRPOperationProfileException
        { string reason; };
    exception GetSimpleUploadBulkCMIRPNotificationProfileException
        { string reason; };

    interface SimpleUploadBulkCMIRP
    {
        /*
        Return the list of all supported Bulk CM IRP versions.
        */
        ManagedGenericIRPConstDefs::VersionNumberSet
        get_simple_upload_bulk_CM_IRP_versions (
        )
        raises (GetSimpleUploadBulkCmIRPVersionsException);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific BulkCM IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
        get_simple_upload_bulk_CM_IRP_operation_profile (
            in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version

```

```

)
raises (GetSimpleUploadBulkCMIRPOperationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Return the list of all supported notifications and their supported
parameters for a specific BulkCM IRP version.
*/
ManagedGenericIRPConstDefs::MethodList
    get_simple_upload_bulk_CM_IRP_notification_profile
(
    in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version
)
raises (GetSimpleUploadBulkCMIRPNotificationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Uploads a configuration from the subnetwork. The result is put in a
configuration data file in an area specified by the IRPManager.
The MIB of the subnetwork is iterated by means of containment search,
using a SearchControl to control the search and the returned results.
All MOs in the scope constitutes a set that the filter works on.
In case of a concurrent running session the function will
return an exception. If the value of the given baseObject or FilterType
does not exist then this asynchronous error condition will be notified.
*/
void upload (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::FileDestination sink,
    in BulkCmIRPConstDefs::DistinguishedName base_object,
    in BulkCmIRPConstDefs::SearchControl search_control
)
raises (
    BulkCmIRPSystem::UploadException,
    BulkCmIRPSystem::UnknownSessionIdException,
    BulkCmIRPSystem::MaxSessionReachedException,
    BulkCmIRPSystem::NotValidInCurrentStateException,
    BulkCmIRPSystem::ConcurrencyException,
    BulkCmIRPSystem::IllegalDNFormatException,
    BulkCmIRPSystem::IllegalFilterFormatException,
    BulkCmIRPSystem::IllegalScopeTypeException,
    BulkCmIRPSystem::IllegalScopeLevelException,
    BulkCmIRPSystem::IllegalURLFormatException,
    ManagedGenericIRPSystem::InvalidParameter);
};

}; // end of module SimpleUploadBulkCMIRPSystem

module ControlledUploadBulkCMIRPSystem
{
    exception GetControlledUploadBulkCmIRPVersionsException { string reason; };
    exception GetControlledUploadBulkCMIRPOperationProfileException
        { string reason; };
    exception GetControlledUploadBulkCMIRPNotificationProfileException
        { string reason; };

    interface ControlledUploadBulkCMIRP
    {

```

```

/*
Return the list of all supported Bulk CM IRP versions.
*/
ManagedGenericIRPConstDefs::VersionNumberSet
    get_controlled_upload_bulk_CM_IRP_versions (
    )
    raises (GetControlledUploadBulkCmIRPVersionsException);

/*
Return the list of all supported operations and their supported
parameters for a specific BulkCM IRP version.
*/
ManagedGenericIRPConstDefs::MethodList
    get_controlled_upload_bulk_CM_IRP_operation_profile (
        in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version
    )
    raises (GetControlledUploadBulkCMIRPOperationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Return the list of all supported notifications and their supported
parameters for a specific BulkCM IRP version.
*/
ManagedGenericIRPConstDefs::MethodList
    get_controlled_upload_bulk_CM_IRP_notification_profile (
        in ManagedGenericIRPConstDefs::VersionNumber bulk_CM_IRP_version
    )
    raises (GetControlledUploadBulkCMIRPNotificationProfileException,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Uploads a configuration from the subnetwork. The result is put in a
configuration data file in an area specified by the IRPManager.
The MIB of the subnetwork is iterated by means of containment search,
using a SearchControl to control the search and the returned results.
All MOs in the scope constitutes a set that the filter works on.
In case of a concurrent running session the function will
return an exception. If the value of the given baseObject or FilterType
does not exist then this asynchronous error condition will be notified.
*/
void upload (
    in BulkCmIRPConstDefs::SessionId session_id,
    in BulkCmIRPConstDefs::FileDestination sink,
    in BulkCmIRPConstDefs::DistinguishedName base_object,
    in BulkCmIRPConstDefs::SearchControl search_control
)
    raises (
        BulkCmIRPSystem::UploadException,
        BulkCmIRPSystem::UnknownSessionIdException,
        BulkCmIRPSystem::MaxSessionReachedException,
        BulkCmIRPSystem::NotValidInCurrentStateException,
        BulkCmIRPSystem::ConcurrencyException,
        BulkCmIRPSystem::IllegalDNFormatException,
        BulkCmIRPSystem::IllegalFilterFormatException,
        BulkCmIRPSystem::IllegalScopeTypeException,
        BulkCmIRPSystem::IllegalScopeLevelException,
        BulkCmIRPSystem::IllegalURLFormatException,
        ManagedGenericIRPSystem::InvalidParameter);

```

```
/*  
Uploads a log from the subnetwork which is usually used for error  
analysis. The log is put in a logfile in the filesystem which can  
be accessed by the EM. If there are no log entries an empty log file  
is uploaded.
```

```
*/  
void get_session_log (  
    in BulkCmIRPConstDefs::FileDestination sink,  
    in BulkCmIRPConstDefs::SessionId session_id,  
    in boolean only_error_info  
)  
    raises (  
        BulkCmIRPSystem::GetSessionLogException,  
        BulkCmIRPSystem::UnknownSessionIdException,  
        BulkCmIRPSystem::IllegalURLFormatException,  
        ManagedGenericIRPSystem::InvalidParameter);
```

```
/*  
Creates an instance of the configuration session state machine. The  
IDLE_PHASE & COMPLETED is notified
```

```
*/  
void start_session (  
    in BulkCmIRPConstDefs::SessionId session_id  
)  
    raises (  
        BulkCmIRPSystem::StartSessionException,  
        BulkCmIRPSystem::SessionIdInUseException,  
        BulkCmIRPSystem::MaxSessionReachedException,  
        ManagedGenericIRPSystem::InvalidParameter);
```

```
/*  
Returns the state of a configuration session.
```

```
*/  
BulkCmIRPConstDefs::SessionState get_session_status (  
    in BulkCmIRPConstDefs::SessionId session_id,  
    out BulkCmIRPConstDefs::ErrorInformation error_information  
)  
    raises (  
        BulkCmIRPSystem::GetSessionStatusException,  
        BulkCmIRPSystem::UnknownSessionIdException,  
        ManagedGenericIRPSystem::InvalidParameter);
```

```
/*  
The IRPManager invokes this operation to delete all its temporary  
entities and the related sessionId which belong to the scope of  
a configuration session. This includes the related error and log  
informationen too.
```

```
*/  
void end_session (  
    in BulkCmIRPConstDefs::SessionId session_id  
)  
    raises (  
        BulkCmIRPSystem::EndSessionException,  
        BulkCmIRPSystem::UnknownSessionIdException,  
        BulkCmIRPSystem::NotValidInCurrentStateException,  
        ManagedGenericIRPSystem::InvalidParameter);
```

```
/*  
The IRPManager invokes this operation to abort an active operation  
during a configuration session. It is only effecting  
a configuration session in state IN_PROGRESS. In this case the  
current session task is interrupted, e.g. the activating in progress,
```

```

using best effort strategy, and a state change is notified
*/
void abort_session_operation (
    in BulkCmIRPConstDefs::SessionId session_id
)
raises (
    BulkCmIRPSystem::AbortSessionOperationException,
    BulkCmIRPSystem::UnknownSessionIdException,
    BulkCmIRPSystem::NotValidInCurrentStateException,
    ManagedGenericIRPSystem::InvalidParameter);

/*
Returns a list all sessionIds of current running configuration sessions.
*/
BulkCmIRPConstDefs::SessionIdList get_session_ids (
)
raises (
    BulkCmIRPSystem::GetSessionIdsException);

};

}; // end of module ControlledUploadBulkCMIRPSystem

#endif __BULKCMIRPSYSTEM_IDL__

```


A.3 IDL specification (file name “BulkCMIRPNotifications.idl”)

```
// File: BulkCMNotifications.idl

#ifndef _BULKCMIRPNOTIFICATIONS_IDL_
#define _BULKCMIRPNOTIFICATIONS_IDL_

#include <NotificationIRPNotifications.idl>
#include <BulkCmIRPConstDefs.idl>

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

module BulkCMIRPNotifications
{
    interface NotifySessionStateChange: NotificationIRPNotifications::Notify
    {
        // This is the type_name (2nd field) of the fixed header.
        const string EVENT_TYPE =
            BulkCmIRPConstDefs::NotificationType::NOTIFY_SESSION_STATE_CHANGED;

        // -----
        // One of the strings here is the event_name (3rd field) of the
        // fixed header.
        // The first 2 are relevant for IS-defined packages Simple
        // Upload and Controlled Upload.
        // All are relevant for IS-defined package
        // Controlled Upload & Provisioning.

        const string UPLOAD_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::UPLOAD_FAILED;
        const string UPLOAD_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::UPLOAD_COMPLETED;
        const string DOWNLOAD_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::DOWNLOAD_FAILED;
        const string DOWNLOAD_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::DOWNLOAD_COMPLETED;
        const string ACTIVATION_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::ACTIVATION_FAILED;
        const string ACTIVATION_PARTLY_REALISED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::ACTIVATION_PARTLY_REALISED;
        const string ACTIVATION_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::ACTIVATION_COMPLETED;
        const string FALLBACK_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::FALLBACK_FAILED;
        const string FALLBACK_PARTLY_REALISED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::FALLBACK_PARTLY_REALISED;
        const string FALLBACK_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::FALLBACK_COMPLETED;
        const string VALIDATION_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::VALIDATION_FAILED;
        const string VALIDATION_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::VALIDATION_COMPLETED;
        const string PREAMTIVATION_FAILED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::PREAMTIVATION_FAILED;
        const string PREAMTIVATION_PARTLY_REALISED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::PREAMTIVATION_PARTLY_REALISED;
        const string PREAMTIVATION_COMPLETED = BulkCmIRPConstDefs::
            SessionStateChangeNotification::PREAMTIVATION_COMPLETED;
    }
}
```

```

// -----
const string SESSION_ID =
    BulkCmIRPConstDefs::AttributeNameValue::SESSION_ID;

const string SOURCE_INDICATOR =
    BulkCmIRPConstDefs::AttributeNameValue::SOURCE_INDICATOR;
};

interface NotifyGetSessionLogEnded: NotificationIRPNotifications::Notify
{
    // This is the type_name (2nd field) of the fixed header.
    const string EVENT_TYPE =
        BulkCmIRPConstDefs::NotificationType::NOTIFY_GET_SESSION_LOG_ENDED;

// -----
// One of the 2 strings here is the event_name (3rd field) of the
// fixed header.
const string GET_SESSION_LOG_COMPLETED_SUCCESSFULLY =
    BulkCmIRPConstDefs::LogStateNotification::
        GET_SESSION_LOG_COMPLETED_SUCCESSFULLY;
const string GET_SESSION_LOG_COMPLETED_UNSUCCESSFULLY =
    BulkCmIRPConstDefs::LogStateNotification::
        GET_SESSION_LOG_COMPLETED_UNSUCCESSFULLY;
// -----

const string SESSION_ID =
    BulkCmIRPConstDefs::AttributeNameValue::SESSION_ID;

const string SOURCE_INDICATOR =
    BulkCmIRPConstDefs::AttributeNameValue::SOURCE_INDICATOR;
};
};

#endif _BULKCMIRPNOTIFICATIONS_IDL_

```

End of Changes

CHANGE REQUEST

⌘ **32.613 CR 012** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ Add Signalling Transport Network (STN) NRM IRP in BulkCM IRP CORBA SS		
Source:	⌘ SA5 (llrui@bupt.edu.cn ; liyewen@chinamobile.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ B	Release:	⌘ Rel-6
	<p>Use <u>one</u> of the following categories:</p> <p>F (correction)</p> <p>A (corresponds to a correction in an earlier release)</p> <p>B (addition of feature),</p> <p>C (functional modification of feature)</p> <p>D (editorial modification)</p> <p>Detailed explanations of the above categories can be found in 3GPP TR 21.900.</p>		<p>Use <u>one</u> of the following releases:</p> <p>2 (GSM Phase 2)</p> <p>R96 (Release 1996)</p> <p>R97 (Release 1997)</p> <p>R98 (Release 1998)</p> <p>R99 (Release 1999)</p> <p>Rel-4 (Release 4)</p> <p>Rel-5 (Release 5)</p> <p>Rel-6 (Release 6)</p>

Reason for change:	⌘ Since Signalling Transport Network (STN) NRM IRP is introduced in R6, the scope of NRM IRP which can use BulkCMIRP needs to be extended.
Summary of change:	⌘ BulkCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP.
Consequences if not approved:	⌘

Clauses affected:	⌘ 1, 2, 3.1										
Other specs affected:	<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> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
Other comments:	⌘ Parent Rel-6 CRs 32.612 in S5-049020										

Change in Clause 1

1 Scope

The purpose of this *Bulk CM IRP: CORBA Solution Set* is to define the mapping of the IRP information service (see 3GPP TS 32.612 [3]) to the protocol specific details necessary for implementation of this IRP in a CORBA/IDL environment.

The present document does not describe any Network Resource Model (NRM) – they are described in Generic Network Resources IRP: NRM 3GPP TS 32.622 [4], [CN Network Resources IRP: NRM 3GPP TS 32.632 \[14\]](#), UTRAN Network Resources IRP: NRM 3GPP TS 32.642 [11], GERAN Network Resources IRP: NRM 3GPP TS 32.652 [12], [STN Network Resources IRP: NRM 3GPP TS 32.742 \[15\]](#).

This Solution Set specification is related to 3GPP TS 32.612 V6.01.X.

End of Change in Clause 1

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 32.612: "Telecommunication management; Configuration Management (CM); Bulk CM Integration Reference Point (IRP); Information Service (IS)".
- [4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP); Network Resource Model (NRM)".
- [5] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [6] OMG Notification Service, Version 1.0.
- [7] OMG CORBA services: Common Object Services Specification, Update: November 22, 1996.
- [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 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [12] 3GPP TS 32.652: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [13] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".
- [14] [3GPP TS 32.632: "Telecommunication management; Configuration Management \(CM\); CN network resources Integration Reference Point \(IRP\): Network Resource Model \(NRM\)".](#)
- [15] [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 3.1

3.1 Definitions

For terms and definitions please refer to 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.612 [3], 3GPP TS 32.622 [4], [3GPP TS 32.632 \[14\]](#), 3GPP TS 32.642 [11], ~~and~~ 3GPP TS 32.652 [12] [and 3GPP TS 32.742 \[15\]](#).

- IRP document version number string (or "IRPVersion"): See 3GPP TS 32.312 [13].

End of Change in Clause 3.1 End of the Document

CHANGE REQUEST

⌘ **32.615 CR 018** ⌘ rev - ⌘ Current version: **6.0.0** ⌘

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

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

Title:	⌘ Add Signalling Transport Network (STN) NRM IRP in BulkCM IRP XML FF		
Source:	⌘ SA5 (llrui@bupt.edu.cn ; liyewen@chinamobile.com)		
Work item code:	⌘ OAM-NIM	Date:	⌘ 19/11/2004
Category:	⌘ B	Release:	⌘ Rel-6
	<p>Use <u>one</u> of the following categories:</p> <p>F (correction)</p> <p>A (corresponds to a correction in an earlier release)</p> <p>B (addition of feature),</p> <p>C (functional modification of feature)</p> <p>D (editorial modification)</p> <p>Detailed explanations of the above categories can be found in 3GPP TR 21.900.</p>		<p>Use <u>one</u> of the following releases:</p> <p>2 (GSM Phase 2)</p> <p>R96 (Release 1996)</p> <p>R97 (Release 1997)</p> <p>R98 (Release 1998)</p> <p>R99 (Release 1999)</p> <p>Rel-4 (Release 4)</p> <p>Rel-5 (Release 5)</p> <p>Rel-6 (Release 6)</p>

Reason for change:	⌘ Since Signalling Transport Network (STN) NRM IRP is introduced in R6, the scope of NRM IRP which can use BulkCMIRP needs to be extended.
Summary of change:	⌘ BulkCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP.
Consequences if not approved:	⌘

Clauses affected:	⌘ 1, 2, 4.3A.1, Annex A										
Other specs affected:	<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> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
Other comments:	⌘ Parent Rel-6 CRs 32.612 in S5-049020										

Change in Clause 1

1 Scope

The present document provides the main part of the XML file format definition for the Bulk Configuration Management IRP IS [1].

The other parts of this XML file format definition are NRM-specific parts.

Those NRM-specific parts are provided by 3GPP TS 32.625 [11], 3GPP TS 32.635 [12], 3GPP TS 32.645 [13], ~~and 3GPP TS 32.655 [14]~~ [and 3GPP TS 32.745 \[15\]](#).

Bulk CM XML file formats are based on XML [2], XML Schema [3] [4] [5] and XML Namespace [6] standards.

This File Format Definition specification is related to 3GPP TS 32.612 V6.~~0~~1.X.

End of Change in Clause 1

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 32.612: "Telecommunication management; Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Information Service (IS)".
- [2] W3C REC-xml-20001006: "Extensible Markup Language (XML) 1.0 (Second Edition)".
- [3] W3C REC-xmlschema-0-20010502: "XML Schema Part 0: Primer".
- [4] W3C REC-xmlschema-1-20010502: "XML Schema Part 1: Structures".
- [5] W3C REC-xmlschema-2-20010502: "XML Schema Part 2: Datatypes".
- [6] W3C REC-xml-names-19990114: "Namespaces in XML".
- [7] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".
- [8] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".
- [9] ~~Void. 3GPP TS 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".~~

- [10] ~~[Void 3GPP TS 32.652: "Telecommunication management; Configuration Management \(CM\); GERAN network resources Integration Reference Point \(IRP\): Network Resource Model \(NRM\)".](#)~~
- [11] 3GPP TS 32.625: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Bulk CM eXtensible Markup Language (XML) file format definition".
- [12] 3GPP TS 32.635: "Telecommunication management; Configuration Management (CM); Core network resources Integration Reference Point (IRP): Bulk CM eXtensible Markup Language (XML) file format definition".
- [13] 3GPP TS 32.645: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Bulk CM eXtensible Markup Language (XML) file format definition".
- [14] 3GPP TS 32.655: "Telecommunication management; Configuration Management (CM); GERAN network resources Integration Reference Point (IRP): Bulk CM eXtensible Markup Language (XML) file format definition".
- [15] [3GPP TS 32.745: "Telecommunication management; Configuration Management \(CM\); Signalling Transport Network \(STN\) Interface Network Resource Model \(NRM\) Integration Reference Point \(IRP\): Bulk CM eXtensible Markup Language \(XML\) file format definition".](#)

End of Change in Clause 2

Change in Clause 4.3A.1

4.3A.1 NRM-specific XML schemas

NRM-specific XML schemas are defined in the NRM-specific parts (see clause 1) of the XML file format definition for the Bulk Configuration Management IRP IS [1].

NRM-specific XML schemas with definition of corresponding XML namespace prefixes (see subclause 4.1) are listed by the following table:

Table 2: NRM-specific XML schemas, corresponding 3GPP TSs and XML namespace prefixes

NRM	XML schema	3GPP TS no.	XML namespace prefix
Generic Network Resources	<code>genericNrm.xsd</code>	32.625 [11]	<code>xn</code>
Core Network Resources	<code>coreNrm.xsd</code>	32.635 [12]	<code>cn</code>
UTRAN Network Resources	<code>utranNrm.xsd</code>	32.645 [13]	<code>un</code>
GERAN Network Resources	<code>geranNrm.xsd</code>	32.655 [14]	<code>gn</code>
STN Network Resources	stnNrm.xsd	32.745 [15]	stn

Each NRM-specific XML schema explicitly declares NRM-specific XML element types for the related NRM.

Additionally, XML schema `genericNrm.xsd` (see [11]) also provides global XML declarations and definitions for the support of:

- NRM-specific XML element type declaration;
- vendor-specific XML element type declaration (see subclause 4.5).

End of Change in Clause 4.3A.1

Annex A (normative): Configuration data file base XML schema (file name "configData.xsd")

The following XML schema configData.xsd is the base schema for configuration data XML files:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  3GPP TS 32.615 Bulk CM IRP
  Configuration data file base XML schema
  configData.xsd
-->

<schema
  targetNamespace=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.615#configData"

  elementFormDefault="qualified"

  xmlns="http://www.w3.org/2001/XMLSchema"

  xmlns:xn=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.625#genericNrm"

  xmlns:cn=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.635#coreNrm"

  xmlns:un=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.645#utranNrm"

  xmlns:gn=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.655#geranNrm"
  xmlns:stn=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.745#stnNrm"
>

  <import
    namespace=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.625#genericNrm"
    />
  <import
    namespace=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.635#coreNrm"
    />
  <import
    namespace=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.645#utranNrm"
    />
  <import
    namespace=
"http://www.3gpp.org/ftp/specs/archive/32_series/32.655#geranNrm"
    />
  <import
namespace=
```

["http://www.3gpp.org/ftp/specs/archive/32_series/32.745#stnNrm"](http://www.3gpp.org/ftp/specs/archive/32_series/32.745#stnNrm)
/>

```
<!-- Configuration data file root XML element -->

<element name="bulkCmConfigDataFile">
  <complexType>
    <sequence>
      <element name="fileHeader">
        <complexType>
          <attribute name="fileFormatVersion" type="string" use="required"/>
          <attribute name="senderName" type="string" use="optional"/>
          <attribute name="vendorName" type="string" use="optional"/>
        </complexType>
      </element>
      <element name="configData" maxOccurs="unbounded">
        <complexType>
          <choice>
            <element ref="xn:SubNetwork"/>
            <element ref="xn:MeContext"/>
            <element ref="xn:ManagedElement"/>
          </choice>
          <attribute name="dnPrefix" type="string" use="optional"/>
        </complexType>
      </element>
      <element name="fileFooter">
        <complexType>
          <attribute name="dateTime" type="dateTime" use="required"/>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>

</schema>
```

End of Change in Annex A
End of the Document