ETSI TS 132 612 V9.0.0 (2010-02)

Technical Specification

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Configuration Management (CM);
Bulk CM Integration Reference Point (IRP):
Information Service (IS)
(3GPP TS 32.612 version 9.0.0 Release 9)

Reference
RTS/TSGS-0532612v900

Keywords
GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

32.611: "Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Requirements".

32.612: "Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Information Service (IS)".

32.613: "Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

32.615: "Configuration Management (CM); Bulk CM Integration Reference Point (IRP): eXtensible Markup Language (XML) file format definition".

32.617: "Configuration Management (CM); Bulk CM Integration Reference Point (IRP): SOAP Solution Set (SS).

Configuration Management (CM), in general, provides the operator with the ability to assure correct and effective operation of the 3G network as it evolves. CM actions have the objective to control and monitor the actual configuration on the Network Element (NEs) and Network Resources (NRs), and they may be initiated by the operator or by functions in the Operations Systems (OSs) or NEs.

CM actions may be requested as part of an implementation programme (e.g. additions and deletions), as part of an optimisation programme (e.g. modifications), and to maintain the overall Quality of Service. The CM actions are initiated either as a single actions on single NEs of the 3G network or as part of a complex procedure involving actions on many resources/objects in one or several NEs.

1 Scope

The present document (Bulk Configuration Management IRP: Information Service) defines 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. 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 32.302: "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)".

[12] 3GPP TS 32.742: "Configuration Management (CM); Signalling Transport Network (STN) interface Network Resource Model (NRM)".

[13] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[14] 3GPP TS 32.712: "Telecommunication management; Configuration Management (CM); Transport Network (TN) Network Resource Model (NRM) Integration Reference Point (IRP): Information Service (IS)".

[15] 3GPP TS 32.732: "IP Multimedia Subsystem (IMS) Network Resource Model (NRM) Integration Reference Point (IRP): Information Service (IS)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply. For terms and definitions not found here, please refer to 3GPP TS 32.101 [1], 3GPP TS 32.102 [2] and 3GPP TS 32.600 [8].

Association: In general it is used to model relationships between Managed Objects. Associations can be implemented in several ways, such as:

(1) name bindings,

(2) reference attributes, and

(3) association objects.

This IRP stipulates that containment associations shall be expressed through name bindings, but it does not stipulate the implementation for other types of associations as a general rule. These are specified as separate entities in the object models (UML diagrams). Currently (in R99) however, all (non-containment) associations are modelled. by means of reference attributes of the participating MOs.

Data: is any information or set of information required to give software or equipment or combinations thereof a specific state of functionality.

Element Manager (EM): provides a package of end-user functions for management of a set of closely related types of Network Elements (NEs). These functions can be divided into two main categories:

- *Element Management Functions* for management of NEs on an individual basis. These are basically the same functions as supported by the corresponding local terminals.
- *Sub-Network Management Functions* that are related to a network model for a set of NEs constituting a clearly defined sub-network, which may include relations between the NEs. This model enables additional functions on the sub-network level (typically in the areas of network topology presentation, alarm correlation, service impact analysis and circuit provisioning).

Integration Reference Point (IRP): See 3GPP TS 32.150 [6].

IRP Information Service (IS): See 3GPP TS 32.101 [1].

IRP Network Resource Model (NRM): See 3GPP TS 32.101 [1].

IRP Solution Set (SS): See 3GPP TS 32.101 [1].

Managed Element (ME): An instance of the Managed Object Class G3ManagedElement/ManagedElement.

Managed Object (MO): In the context of the present document, a Managed Object (MO) is a software object that encapsulates the manageable characteristics and behaviour of a particular Network Resource. The MO is instance of a MO class defined in a MIM/NRM. An MO class has attributes that provide information used to characterize the objects that belong to the class (the term "attribute " is taken from TMN and corresponds to a "property " according to CIM). Furthermore, a MO class can have operations that represent the behaviour relevant for that class (the term "operation " is taken from TMN and corresponds to a "method " according to CIM). An MO class may support notifications that provide information about an event occurrence within a network resource.

Managed Object Class (MOC): a description of all the common characteristics for a number of MOs, such as their attributes, operations, notifications and behaviour.

Managed Object Instance (MOI): an instance of a MOC, which is the same as a MO as described above.

Management Information Base (MIB): A MIB is an instance of an NRM and has some values on the defined attributes and associations specific for that instance. In the context of the present document , a MIB consist of (1) a Name space (describing the MO containment hierarchy in the MIB through Distinguished Names), (2) a number of Managed Objects with their attributes and (3) a number of Associations between these MOs. Also note that TMN (X.710 [7]) defines a concept of a Management Information Tree (also known as a Naming Tree) that corresponds to the name space (containment hierarchy) portion of this MIB definition. Figure 3.1 depicts the relationships between a Name space and a number of participating MOs (the shown association is of a non-containment type)

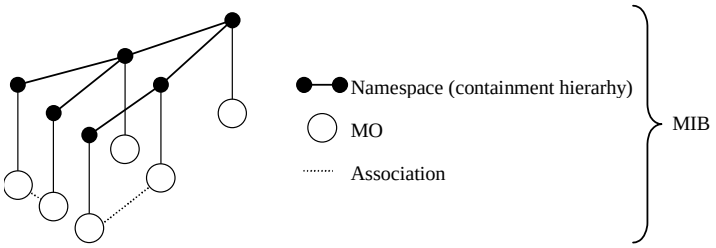


Figure 3.1: Relationships between a Name space and a number of participating MOs

Management Information Model (MIM): Also referred to as NRM – see the definition below. There is a slight difference between the meaning of MIM and NRM – the term MIM is generic and can be used to denote any type of management model, while NRM denotes the model of the actual managed telecommunications Network Resources (NRs).

Name space: A name space is a collection of names. The IRP name convention [7] restricts the name space to a hierarchical containment structure, including its simplest form - the one-level, flat name space. All Managed Objects in a MIB shall be included in the corresponding name space and the MIB/name space shall only support a strict hierarchical containment structure (with one root object). A Managed Object that contains another is said to be the superior (parent); the contained Managed Object is referred to as the subordinate (child). The parent of all MOs in a single name space is called a Local Root. The ultimate parent of all MOs of all managed systems is called the Global Root.

Network Element (NE): is a discrete telecommunications entity, which can be, managed over a specific interface e.g. the RNC.

Network Manager (NM): provides a package of end-user functions with the responsibility for the management of a network, mainly as supported by the EM(s) but it may also involve direct access to the NEs. All communication with the network is based on open and well-standardised interfaces supporting management of multi-vendor and multi-technology NEs.

Network Resource (NR): is a component of a NE, which can be identified as a discrete separate entity and is in an object oriented environment for the purpose of management represented by an abstract entity called Managed Object (MO).

Network Resource Model (NRM): a model representing the actual managed telecommunications Network Resources (NRs) that a System is providing through the subject IRP. An NRM describes Managed Object Classes (MOC), their associations, attributes and operations. The NRM is also referred to as "MIM" (see above) which originates from the ITU-T TMN.

Operator: is either a human being controlling and managing the network; or a company running a network (the 3G network operator).

3.2 Abbreviations

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

CM	Configuration Management
EM	Element Manager
FM	Fault Management
IRP	Integration Reference Point
MIB	Management Information Base
MIM	Management Information Model
MO	Managed Object
MOC	Managed Object Class

MOI	Managed Object Instance
NE	Network Element
NM	Network Manager
NR	Network Resource
NRM	Network Resource Model
PM	Performance Management
TM	Telecom Management
UML	Unified Modelling Language (OMG)
XML	EXtensible Markup Language

4 System Overview

4.1 System Context

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [13] subclause 4.7.

In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below.

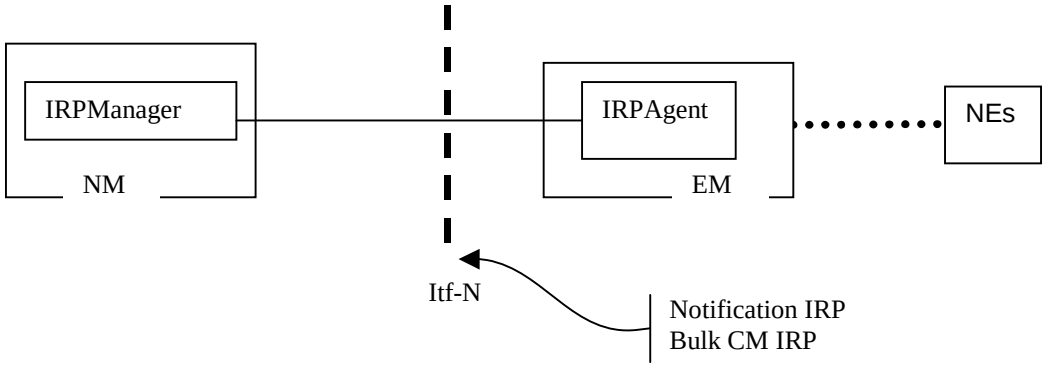


Figure 4.1: System Context A

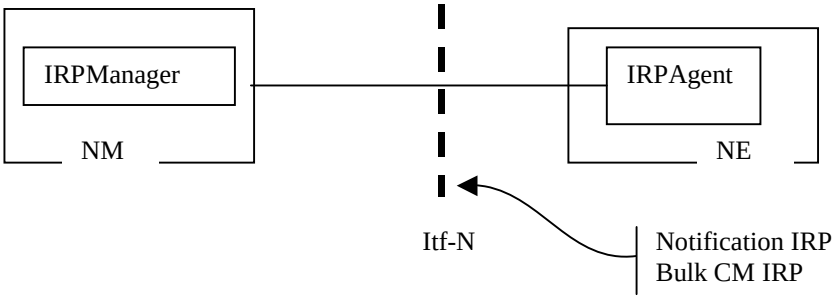


Figure 4.2: System Context B

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 32.150 [13].

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

- 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 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 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 the present document, it is specified how BulkCMIRPsupports the monitoring and provisioning of NEs over Interface-N. It is not within the scope of the present document to specify how BulkCMIRP 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 IRPAgent, is beyond the scope of the present document.

NRMs for Bulk CM IRP are defined in other Network Resource IRP documents of CM. Within the scope of the present 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: "Configuration Management (CM); Inventory Management (IM) network resource Integration Reference Point (IRP); Network Resource Model (NRM)" [11],

32.712 "Configuration Management (CM); Transport Network (TN) interface Network Resource Model (NRM)" [14].

32.732 "IP Multimedia Subsystem (IMS) Network Resource Model (NRM) Integration Reference Point (IRP): Information Service (IS)" [15].

32.742: "Configuration Management (CM); Signalling Transport Network (STN) interface Network Resource Model (NRM)" [12].

The above listed NRM specifications define all the IOCs and attributes that can be configuration managed by the Bulk CM IRP IS.

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, TS 32.632 [10] and in Inventory Management NRM IRP, TS 32.692 [11];
- Manage instances of IOCs specified in Inventory Management NRM IRP TS 32.692 [11].

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

5 Modelling approach

See 3GPP TS 32.102 [2] clause 10.

6 Information Object Classes

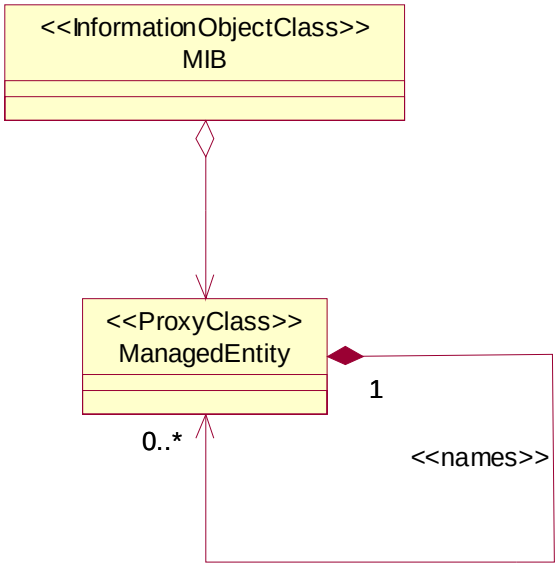
6.1 Information entities imported and local label

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

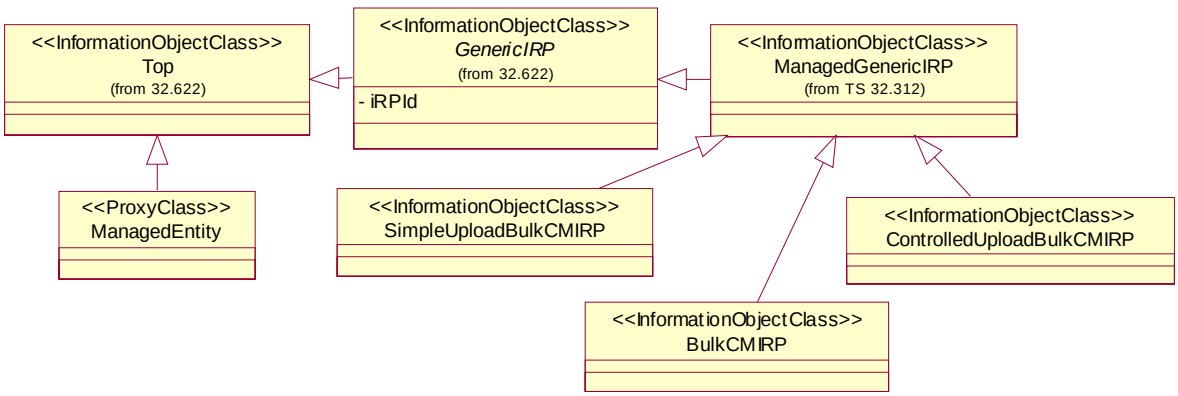
6.2 Class diagram

This clause introduces the set of information object classes (IOCs) that encapsulate capabilities contained by the `IRPAgent`. 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].

6.3.1.2 Attributes

There is no additional attribute defined for this IOC besides those inherited.

6.3.1.3 Notifications

Name	Qualifier	Notes
notifySessionStateChange	M	See 7.6.1.

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].

6.3.2.2 Attributes

There is no additional attribute defined for this IOC besides those inherited.

6.3.2.3 Notifications

Name	Qualifier	Notes
notifySessionStateChange	M	See 7.6.1.
notifyGetSessionLogEnded	M	See 7.7.1.

6.3.3 BulkCMIRP

6.3.3.1 Definition

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

6.3.3.2 Attributes

There is no additional attribute defined for this IOC besides those inherited.

6.3.3.3 Notifications

Name	Qualifier	Notes
notifySessionStateChange	M	See 7.6.1.
notifyGetSessionLogEnded	M	See 7.7.1.

6.4 Void

7 Interface Definition

Clause 7.1 specifies the operations and notifications.

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 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.

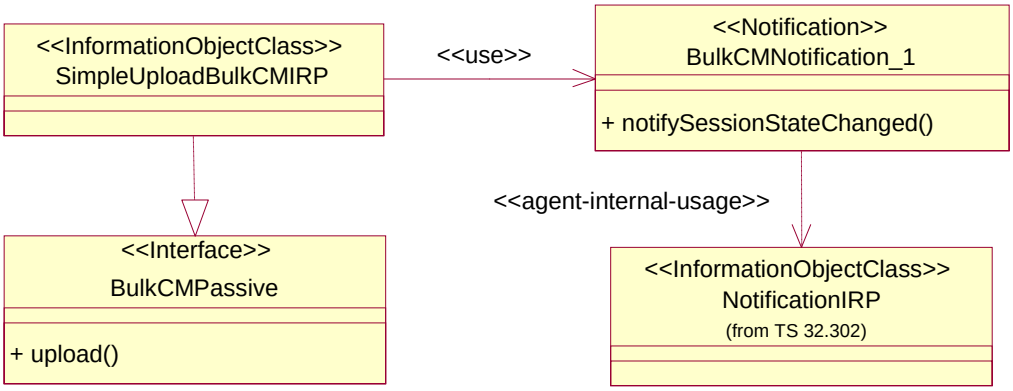
The operations `startSession, endSession, abortSessionOperation, getSessionIds, getSessionStatus` and `getBulkCmIRPVersion,` etc. are performed synchronously in that the result of the operation is returned as a callback to the operation, and the IRPManager will wait until the response is received before 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 network management applications:

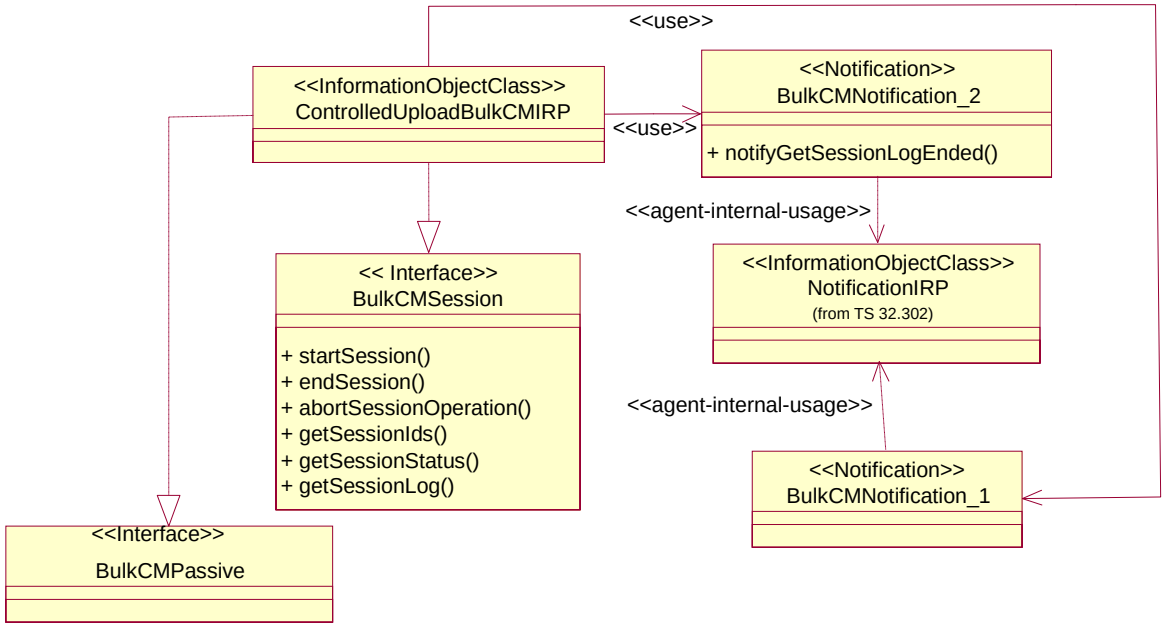
1. BulkCM SimpleUpload, enabling upload of resource information by the IRPManager without explicit session control. This requires the following Interface and Notification specified in clause 7.1.1:
    - BulkCMPassive
    - BulkCMNotification_1
2. BulkCM Controlled Upload, enabling a session controlled upload of resource information by the IRPManager. This requires the following Interfaces and Notification specified in clause 7.1.2:
    - BulkCMPassive
    - BulkCMSession
    - BulkCMNotification_1
    - BulkCMNotification_2
3. BulkCM Controlled Upload & Provisioning, enabling a session controlled upload and provisioning of resource information by the IRPManager. This requires the following Interfaces and Notification specified in clause 7.1.3:
    - BulkCMPassive
    - BulkCMSession
    - BulkCMActive
    - BulkCMNotification_1
    - BulkCMNotification_2

7.1 Class Diagram

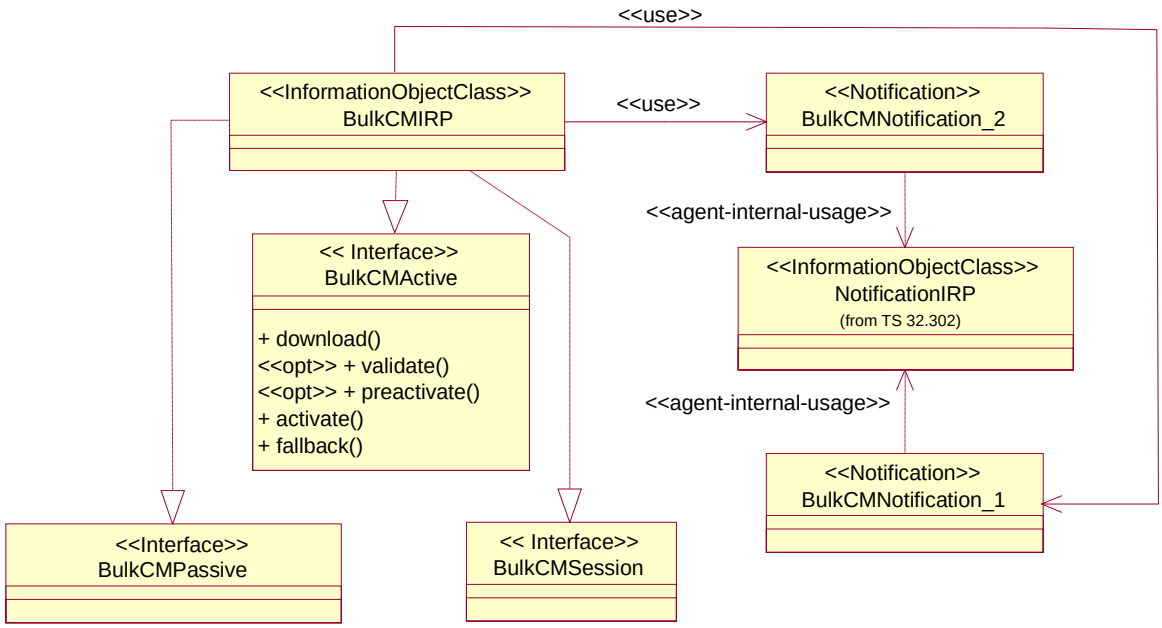
7.1.1 Operations and Notifications for Simple Upload



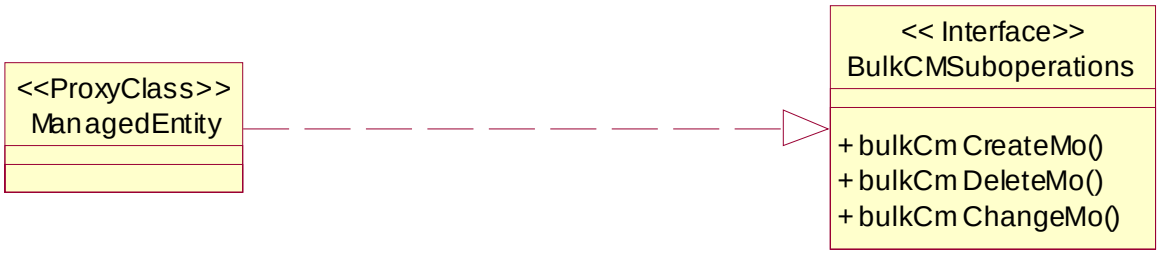
7.1.2 Operations and Notifications for Controlled Upload



7.1.3 Main Operations and Notifications for Controlled Upload & Provisioning



7.1.4 Suboperations for Controlled Upload & Provisioning (of clause 10)



7.2 Generic rules

Rule 1: each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

Rule 2: Each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

Rule 3: each operation shall support a generic exception operation_failed_internal_problem which is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

7.3 Interface BulkCMSession

7.3.1 Operation startSession (M)

7.3.1.1 Definition

The IRPManager invokes this operation to start a session state machine and initialise temporary entities to be related with bulk data configuration sessionId in the IRPAgent.

7.3.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	Identifies the new session and process to be associated with a bulk data operation e.g. upload or download.

7.3.1.3 Output parameters

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

7.3.1.4 Pre-condition

sessionIdNotInUse

Assertion Name	Definition
sessionIdNotInUse	No state, see clause 9. The supplied sessionId is not already open in the BulkCMIRP.

7.3.1.5 Post-condition

sessionStarted

Assertion Name	Definition
sessionStarted	State = IDLE, see clause 9. The BulkCMIRP has successfully opened the session and is ready to handle other operations associated with the session.

7.3.1.6 Exceptions

7.3.1.6.1 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.

7.3.2 Operation endSession (M)

7.3.2.1 Definition

The IRPManager invokes this operation to end a session state machine and delete all temporary entities and their related bulk data configuration for a specified sessionId in the IRPAgent. If a preactivation had been invoked, endSession should release any internal local resources allocated for the preactivation. The deletion will be rejected if the configuration state is in a working state: e.g. uploading (including getting a log), downloading or activating.

7.3.2.2 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

7.3.2.3 Output parameters

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

7.3.2.4 Pre-condition

sessionInStableState

Assertion Name	Definition
sessionInStableState	The supplied sessionId is open in the BulkCMIRP and in not in a transition status as defined in clause 9, table 1.

7.3.2.5 Post-condition

sessionEnded

Assertion Name	Definition
sessionIdNotInUse	No state, see clause 9. The session is closed and the sessionId is no longer in use.

7.3.2.6 Exceptions

7.3.2.6.1 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.

7.3.3 Operation abortSessionOperation (M)

7.3.3.1 Definition

An IRPManager invokes this operation to request an IRPAgent to abort a currently activate asynchronous operation. The abort will cause the session state machine to exit the current state and enter a new state, see clause 9.

7.3.3.2 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 operation e.g. upload or download for which the abort is required.

7.3.3.3 Output parameters

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

7.3.3.4 Pre-condition

operationInProgress

Assertion Name	Definition
operationInProgress	The supplied sessionId is open in the BulkCMIRP and an operation is in an ‘in progress’ state as defined in clause 9, table 1.

7.3.3.5 Post-condition

operationAborted

Assertion Name	Definition
operationAborted	State changed from ‘in progress’ to state as a function of the original state as defined in clause 9.

7.3.3.6 Exceptions

7.3.3.6.1 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.

7.3.4 Operation getSessionIds (M)

7.3.4.1 Definition

An IRPManager invokes this operation to request an IRPAgent to return a list of all its currently open sessionIds.

7.3.4.2 Input parameters

None.

7.3.4.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
sessionIdList	M	List of strings identifying sessions	A list of all the sessionIds an IRPAgent currently has open i.e. started with startSession and not ended with endSession operations.
status	M	ENUM(OperationSucceeded, OperationFailed).	indicates (a) operation is successful and (b) operation failed because of specified or unspecified reasons

7.3.4.4 Pre-condition

None.

7.3.4.5 Post-condition

None.

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

7.3.5.4 Pre-condition

knownSessionID

Assertion Name	Definition
knowSessionID	Session has been successfully started (clause 7.3.1) and not ended (clause 7.3.2).

7.3.5.5 Post-condition

None.

7.3.5.6 Exceptions

7.3.5.6.1 operationFailed

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

7.3.6 Operation getSessionLog (M)

7.3.6.1 Definition

An IRPManager invokes this operation to request an IRPAgent to provide a log of the results from activities associated with bulk data configuration file sessionId operations.

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

7.3.6.2 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.
logFileReference	M	String of complete path of file and name.	Specifies the address and file name where the result is to be placed in the IRPManager.
contentType	M	Boolean	Identifies if retrieved file should include (a) complete log including errors, (b) only errors.

7.3.6.3 Output parameters

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

7.3.6.4 Pre-condition

knownSessionID

Assertion Name	Definition
knowSessionID	Session has been successfully started (clause 7.3.1) and not ended (clause 7.3.2).

7.3.6.5 Post-condition

sessionLogWrite

Assertion Name	Definition
sessionLogWrite	The BulkCMIRP will begin to write contents of log to the specified address and file.

7.3.6.6 Exceptions

7.3.6.6.1 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.

7.4 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.

IRPManager can ask for the file to be compressed. IRPManager expresses its wish via the use of `uploadDataFileReference` input parameter.

The syntax of this input parameter, in particular how it can convey the request for a compressed file using a specific compression format, is outside the scope of standard. One suggested syntax would be the use of the file extension or suffix, e.g. 'zip' of file.zip or 'ZIP' of file.ZIP or 'gz' of file.gz to indicate the compressed file format used.

IRPManager and IRPAgent should have an agreement on the choice of file compression format(s) used and the syntax of `uploadDataFileReference` and in particular, how that syntax can indicate if the file is compressed and by which file compression format. How that agreement is reached is outside the scope of standardization

7.4.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	This identifies this specific session and process associated with the requested bulk data upload.
uploadDataFileReference	M	String of complete path of file	This specifies a globally unique file reference to where the specified scope of bulk data is to be uploaded and stored.
baseObjectInstance	M	DistinguishedName	The DN of the MO where the search (see parameter scope below) starts. The DN is in accordance to 3GPP TS 32.300 [7].
scope	M	SEQUENCE < ENUM { BASE _ONLY, BASE_NTH_LEVEL, BASE_SUBTREE, BASE_ALL}, Level> Note: Level contains valid information if NTH_LEVEL_SUBORDINATES or BASE_NTH_LEVEL is used.	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: • BASE_ONLY: level ignored, just return the base object; • BASE_NTH_LEVEL: return all subordinate objects that are on "level" distance from the base object, where 0 is the base object; • BASE_SUBTREE: return the base object and all of its subordinates down to and including the nth level; • BASE_ALL: level ignored, return the base object and all of it's subordinates.
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(OperationSucceded, OperationFailed).	This indicates (a) start of operation is successful and (b) operation failed because of specified (e.g. IRPManager requesting a file compression format not supported by IRPAgent) or unspecified reasons.

7.4.1.4 Pre-condition

sessionIdle

Assertion Name	Definition
sessionIdle	State as defined in clause 9. The BulkCMIRP 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 BulkCMIRP 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	**Condition:** Pre-condition is false or post-condition is false. **Returned information:** The output parameter status. **Exit state:** Entry state.

7.5 Interface BulkCMActive

7.5.1 Operation download (M)

7.5.1.1 Definition

An IRPManager invokes this operation to request an IRPAgent to download and administer a file containing bulk configuration data (clause 10). The IRPAgent obtains the configuration data file from the indicated globally unique data file reference.

IRPManager can first compress the file before invoking this `download` operation. IRPManager uses the `downloadDataFileReference` input parameter to indicate, among other things, if the file is compressed and if it is compressed, by which compression format.

The syntax of this input parameter, and in particular how it can indicate that the file is compressed by a specific compression format, is outside the scope of standard. One suggested syntax would be the use of the file extension or suffix, e.g. 'zip' of file.zip or 'ZIP' of file.ZIP or 'gz' of file.gz, to indicate the compressed file format used.

IRPManager and IRPAgent should have a prior agreement on the syntax of `downloadDataFileReference` and the choice of file compression format used. How that agreement is reached is outside the scope of standardization.

For checks made during download see clause 7.5.6.

7.5.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
sessionId	M	String identifying the session	This identifies this specific session and process associated with the requested bulk data download.
downloadDataFileReference	M		This specifies a globally unique file reference from where the data to be fetched and download from.

7.5.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM(OperationSucceded, OperationFailed).	This indicates (a) start of operation is successful or (b) operation failed because of specified (e.g. IRPManager has compressed the file using a format not supported by IRPAgent) or unspecified reasons.

7.5.1.4 Pre-condition

sessionIdle

Assertion Name	Definition
sessionIdle	State as defined in clause 9. The BulkCMIRP has successfully opened the session and is ready to handle the first operations of the session or repeat this operation.
validDownloadData FileReference	The downloadDataFileReference is valid. An example of invalid downloadDataFileReference is: BulkCMIRP does not support the file compression format indicated in downloadDataFileReference.

7.5.1.5 Post-condition

downLoadInProgress

Assertion Name	Definition
downLoadInProgress	State = UDOWNLOAD_IN_PROGRESS, as defined in clause 9. The BulkCMIRP has successfully started the download of the configuration data changes.

7.5.1.6 Exceptions

7.5.1.6.1 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.

7.5.2 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(OperationSucceded, 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	State as defined in clause 9. The BulkCMIRP 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 BulkCMIRP 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)

7.5.3.1 Definition

An IRPManager invokes this operation to request an IRPAgent to preactivate previously downloaded bulk configuration data (clause 10) that may have optionally been validated (clause7.5.3). The principal, but not mandatory, functions of the preactivate operation is to validate the configuration data changes in the context of current operational data and to pre-process the configuration data changes. Use of this optional operation enables the IRPManager to prepare the activation of the downloaded bulk configuration data at the EM or NE level before requesting its effective activation. The actions shall fall short of executing the bulk configuration data changes (clause 10) in the network and impacting service. (The actions may for example be to validate the configuration data changes in the context of current operational data or to pre-process the configuration data changes). Performing such actions prior to activate may help identify any potential problems prior to executing the changes on a live a network and may minimise activation elapse time. See also clause 7.5.6 for scope of checks during a session and specifically for preactivate.

Specifying an activation mode is optional. There can only be one activation mode for a session. If an activation mode is specified for the preactivation, it shall be when the first preactivate or validate operation is requested. If an activation mode was specified by validate it is not possible to change the activation mode initially specified with any subsequent preactivate or activate operations. If an activation mode was specified for the first preactivate operation, it is not possible to change the activation mode initially specified with any subsequent preactivate retries, activate or activate 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 preactivate, it cannot be subsequently specified with any subsequent preactivate retries, activation or activation retries. (If specification of an activation mode is required, a new session, download, validate, preactivate and activate should be started.) See also clauses 7.5.2 and 7.5.4.

See clause 6.2.4.3 for description of optional verification mode parameter and associated checking.

Selecting a fallback option is optional. There can only be one fallback option for a session.

If the option is selected it shall be initiated when the first preactivation operation is requested. If a fallback option is not requested for the first preactivation, it cannot be subsequently requested for repeated preactivations or activations during the session. If the fallback option was requested, it is not possible to change the fallback option initially selected with any subsequent re- preactivate retries i.e. for a session it is only possible to fallback to the configuration that existed when the first preactivate operation was requested. See also clause 7.5.5. (If a new fallback configuration is required a new session, download, activate and preactivate should be started. The old session can be ended, prior to which fallback can optionally be invoked).

Specifying how preactivate operation retries within a session shall be implemented following a partially successful preactivation (e.g. repeat all preactivation management actions or just the uncompleted delta of management actions that did not previously complete successfully) is beyond the scope of the present document. Only the IRPManager can initiate preactivate retries. (The IRPAgent shall not initiate retries autonomously).

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(OperationSucceded, OperationFailed).	indicates (a) start of operation is successful or (b) operation failed because of specified or unspecified reasons

7.5.3.3 Pre-condition

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

7.5.3.5 Post-condition

preactivationInProgress

Assertion Name	Definition
preactivationInProgress	State = PREACTIVATION_IN_PROGRESS, as defined in clause 9. The BulkCMIRP 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)

7.5.4.1 Definition

An IRPManager invokes this operation to request an IRPAgent to activate previously downloaded bulk configuration data (clause 10) that may have optionally been checked (clause 7.5.2) and/or been preactivated (clause 7.5.3). Activate means that operations specified in a previously downloaded configuration data file, for example create, delete and modify of managed objects are carried out on the live network i.e. mobile subscribers are affected by the downloaded configuration data.

An IRPAgent may support an optional activationMode parameter. This enables the IRPManager to indicate to the IRPAgent the preference for how the activation shall be executed. One of two options may be selected: "least service impact" or "least elapse time". If the "least service impact" option is selected the IRPAgent shall optimise the execution of the activation in a way that minimises disruption to network services. Elapse time to complete the activation is of secondary importance. If the "least elapse time" option is selected the IRPAgent shall optimise the execution of the activation in a way that minimises the elapse time for completing the execution of the activation. During the execution, disruption of network services is of secondary importance.

See clause 7.5.6 for descriptions of checks made during activate execution.

Specifying an activation mode is optional. There can only be one activation mode for a session. If an activation mode is specified for the activation, it shall be when the first activate, validate or preactivate operation is requested. If an activation mode was specified by validate or preactivate operations, it is not possible to change the activation mode initially specified with any subsequent activate operations. If an activation mode was specified for the first activate, it is not possible to change the activation mode initially specified with any subsequent activate 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 activate, it cannot be subsequently specified with any subsequent activate retries. (If specification of an activation mode is required, a new session, download, validate, preactivate and activate should be started.) See also clauses 7.5.2 and 7.5.3.

If a preactivation had been invoked, successful completion of activate should release any internal local resources allocated for the preactivation.

Selecting a fallback option is optional. There can only be one fallback option for a session.

If the fallback option is selected it shall be initiated when the first activation or preactivation operation is requested. If a fallback option is not requested for the first activation or preactivation, it cannot be subsequently requested for repeated activations or an activation following a preactivation during the session. If the fallback option was requested, it is not possible change the fallback option initially selected with any subsequent re-activate retries or an activation following a preactivation i.e. for a session it is only possible to fallback to the configuration that existed when the first activate or preactivate operation was requested. See also clause 7.5.5. (If a new fallback configuration is required a new session, download and activate should be started. The old session can be ended, prior to which fallback can optionally be invoked).

Specifying how activate operation retries within a session shall be implemented following a partially successful activation (e.g. repeat all activation management actions or just the uncompleted delta of management actions that did not previously complete successfully) is beyond the scope of the present document. Only the IRPManager can initiate activate retries. (The IRPAgent shall not initiate retries autonomously).

7.5.4.2 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.
activationMode	O		Identifies whether a specific activation mode is required. See also clauses 7.5.2 and 7.5.3. It may be set to indicate "least service impact" or "least elapse time" types of activation are required.
fallbackEnabled	M		Indicates whether or not it is required to initialise and enable fallback option prior to the activation. This option is only open for the first activate operation of a session. For any subsequent activate operation retries within a session the fallbackEnabled parameter must be set to indicate it is not required to initialise fallback otherwise the activate operation retry shall fail.

7.5.4.3 Output parameters

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

7.5.4.4 Pre-condition

downLoaded

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

7.5.4.5 Post-condition

activationInProgress

Assertion Name	Definition
activationInProgress	State = ACTIVATE_IN_PROGRESS, as defined in clause 9. The BulkCMIRP has successfully started the activation of the downloaded configuration data.

7.5.4.6 Exceptions

7.5.4.6.1 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.

7.5.5 Operation fallback (M)

7.5.5.1 Definition

An IRPManager may invoke this operation to request an IRPAgent to recover (best effort) after a previously executed activation or preactivation operation.

If a fallback is requested after a preactivation but before an activation the IRPAgent should as necessary return any internal local resources impacted by the preactivation back to the same state they were in prior to the preactivation being invoked. There is no impact to the operational network resources as the activate operation has not been invoked.

If fallback is requested after an activation the IRPAgent shall instigate activating the fallback area to restore the operational network resources impacted by the configuration changes for the session back to the configuration they were in when the fallback option was selected during the session. If a preactivation was also performed, as necessary the IRPAgent should return any internal local resources impacted by the preactivation back to the same state they were in prior to the preactivation being invoked.

Specifying how fallback operation retries within a session shall be implemented after a fallback fails (e.g. repeat all fallback functions or just the delta of fallback functions that did not previously complete successfully) is beyond the scope of the present document. Only the IRPManager can initiate the fallback operation. The IRPAgent shall not initiate fallback or fallback retries autonomously. Within a session the fallback operation shall only be accepted if an initial activate or preactivate operations was performed with fallback option enabled. For further discussion of enabling or not the fallback option see clause 7.5.4.

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
status	M	ENUM(OperationSucceded, 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 BulkCMIRP 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 BulkCMIRP 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

7.5.6.1 Download Checks

During download the IRPAgent should check the consistency of imported configuration data against the data schema to ensure there are no errors. The IRPAgent is not required to check the semantic of the downloaded bulk configuration data during the download.

7.5.6.2 Validate Checks

During validation the IRPAgent should check the syntax and semantic of previously downloaded bulk configuration data.

7.5.6.3 Preactivation Checks

During preactivation the IRPAgent should check the semantic of previously downloaded bulk configuration data, and must also check the syntax if a validate operation has not previously been successfully performed.

An Element Manager should, if technically feasible, send the configuration data changes to all Network Elements (NE) for the NE to verify, to the extent possible, that the activate will successfully execute the configuration data changes. If any elements of configuration change data that will not successfully execute are identified, diagnostic data identifying the NEs and failing configuration data elements will be made available to the Manager.

An IRPAgent may support an optional verification mode parameter, see clause 7.5.2. When the IRPManager does not require extensive checking, this parameter may be used to constrain the scope of validation to avoid performing checks that potentially may require extensive real time to execute, for example checks actively involving entities outside the IRPAgent such as NE's. The validation mode parameter has two values: "full checking" and "limited checking". In the "full checking" mode, the checking should be as complete as possible with the intent of achieving the greatest assurance that the subsequent activation operation will be successful. In the "limited checking" mode, checking that can be performed by the IRPAgent rapidly is still performed, but further checking that may cause significant delays to execute should be omitted.

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 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.

7.6.1.2 Input Parameters

Parameter Name	Qualifiers	Matching Information	Comment
objectClass	M, Y	ManagedEntity.objectClass	Notification header - see [3].
objectInstance	M, 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	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

Parameter Name	Qualifiers	Matching Information	Comment
objectClass	M, Y	ManagedEntity.objectClass	Notification header - see [3].
objectInstance	M, 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	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 Void

9 State Machine

9.1 State Machine Overview

The Bulk CM IRPAgent state machine satisfies the following general requirements and characteristics for Bulk CM IRP:

1) Each configuration session is associated with one state machine. The session is identified by the sessionId. If a session is started (startSession operation) an instance of the state machine is created. If the session is ended (endSession operation) the instance of the state machine is deleted.

2) Under normal operation without errors the IRPManager is able to supervise a configuration session by just monitoring the state change notifications (notifySessionStateChanged) triggered by the IRPAgent

3) Under abnormal conditions where the IRPManager is not notified of a change, the getSessionStatus operation can be invoked to determine current state of the session. The IRPManager does not need to maintain a history of the state machine.

4) On the IRPAgent there is only one download configuration data file (clause 10) associated with a session at a time.

5) Multi configuration session must be supported by the IRPAgent. E.g. it must be possible to invoke an upload session in parallel with an active activate session.

6) The IRPAgent resolves concurrency problems on a "first come - first serve" basis. E.g. an upload and an activation requested on the same configuration data cannot be performed at the same time and in this case the first will be progress to completions and the second request rejected.

7) It must be possible to abort a configuration session within a transition state.

8) The operator/IRPManager decides on whether or not enabling the fallback option is required before requesting an activation or preactivation Enabling the fallback option will maintain the disposition of the configuration before the activation or preactivation . The fallback configuration information is established at point before the first activation or preactivation is started. If there is multiple activation or preactivation attempts during a session only one (first) fallback configuration is maintained.

9) The session log file can be requested in any state. The uploaded log file contains information which is specific to the configuration session.

10) Clause 7.3 defines the valid state machine pre and post conditions for each operation.

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 is not received.

The session state notification events are a 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 is 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.

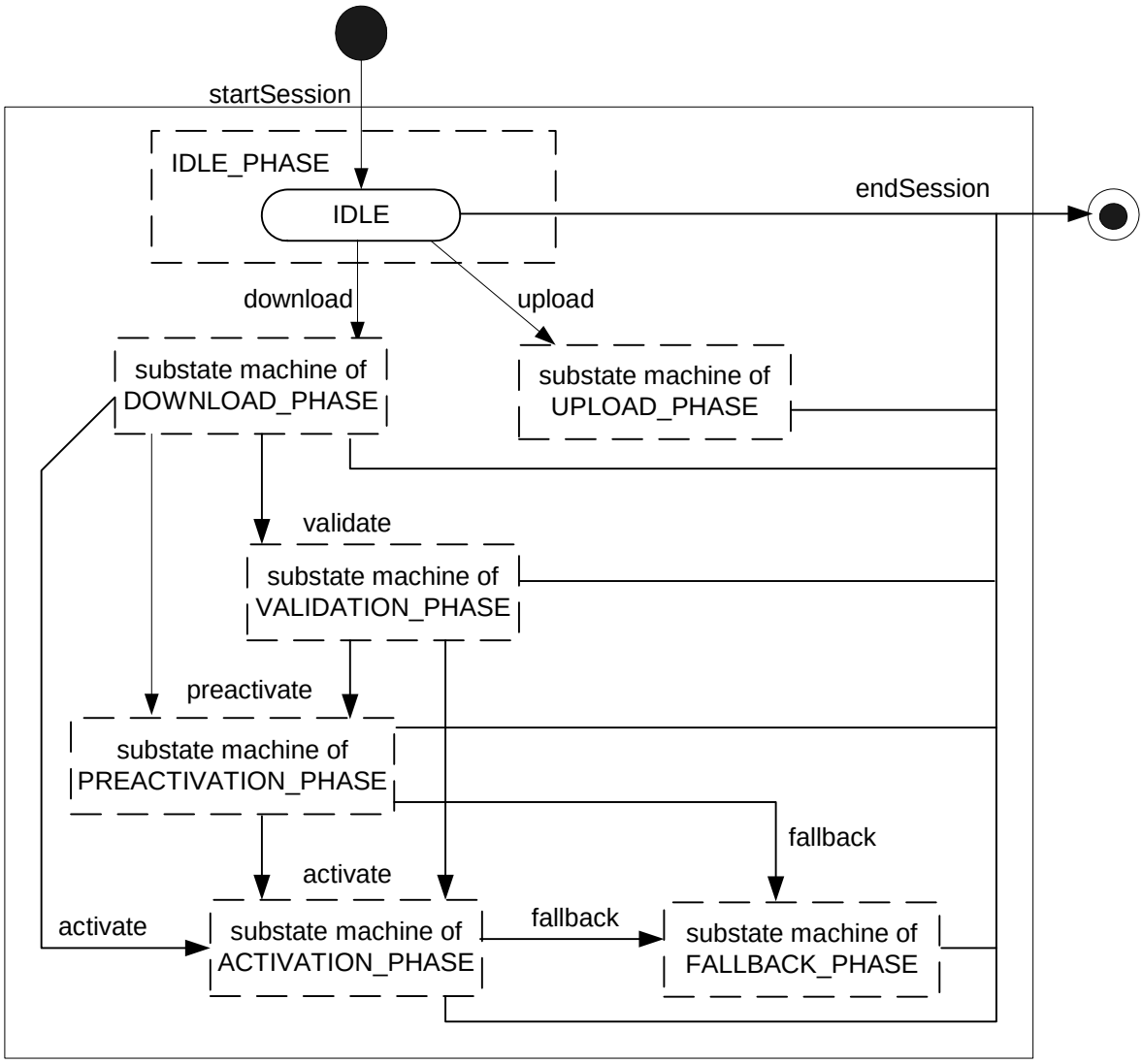


Figure 1: State Machine for Controlled Upload and Controlled Upload & Provisioning

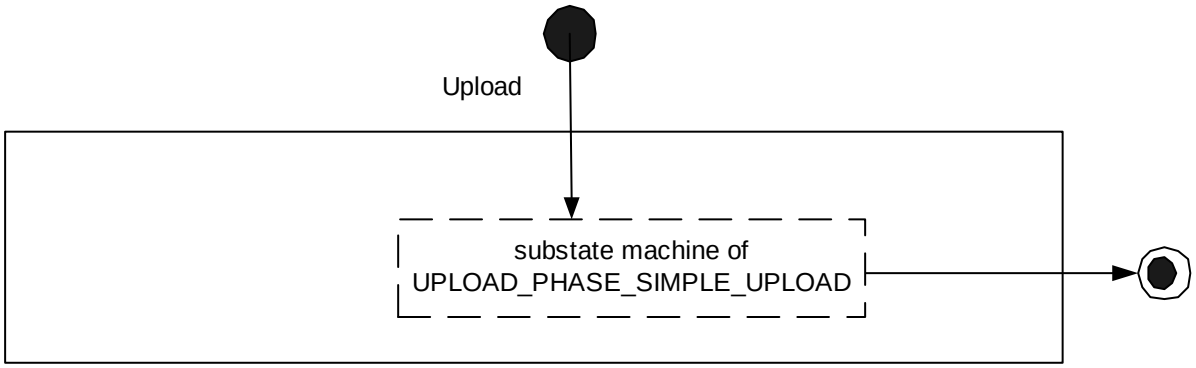


Figure 1a: State Machine for Simple Upload

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 IRPAgent 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.

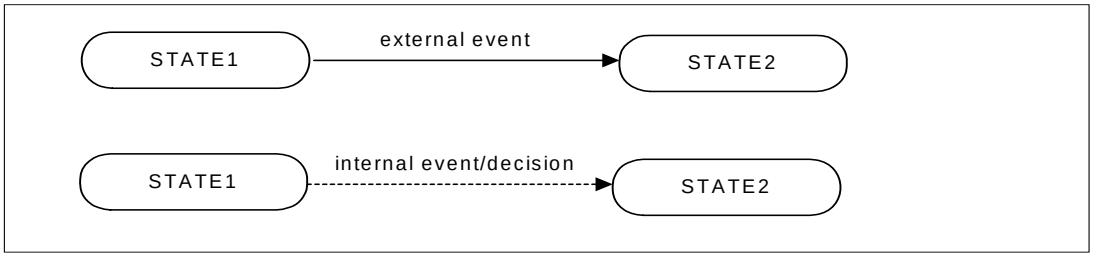


Figure 2: Depicting State Transition Lines for Internal and External Events and Decision

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.

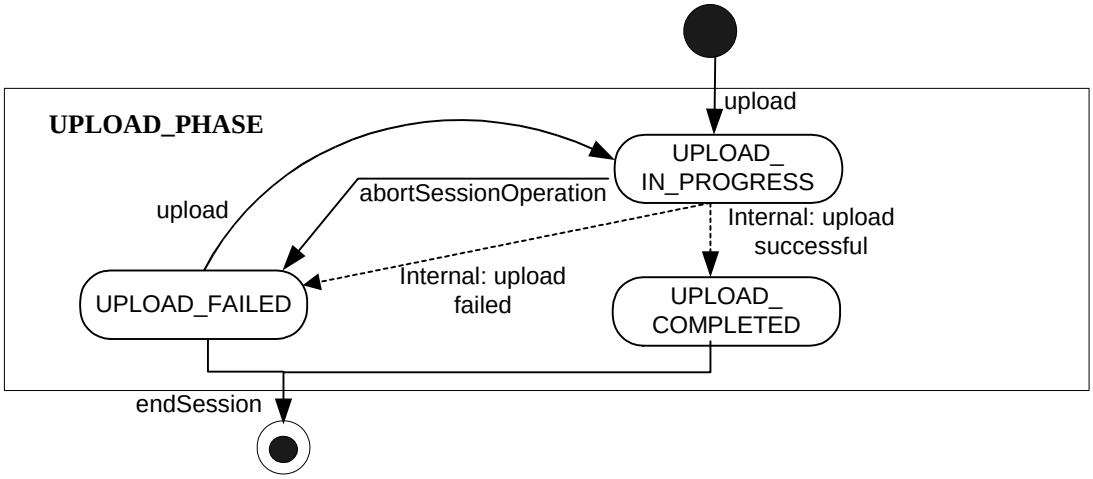


Figure 9.2.1: Substate Machine - UPLOAD_PHASE

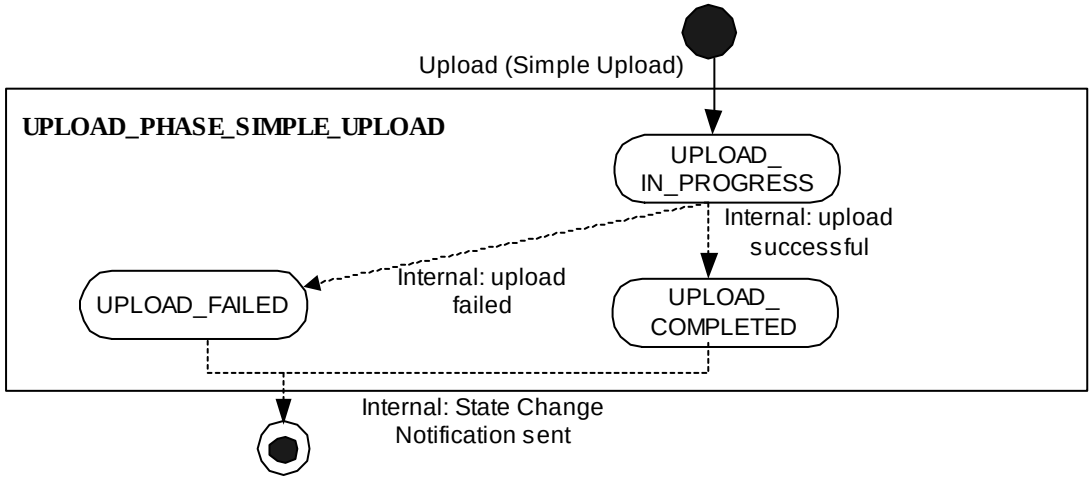


Figure 9.2.1a: Substate Machine - UPLOAD_PHASE_SIMPLE_UPLOAD

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.

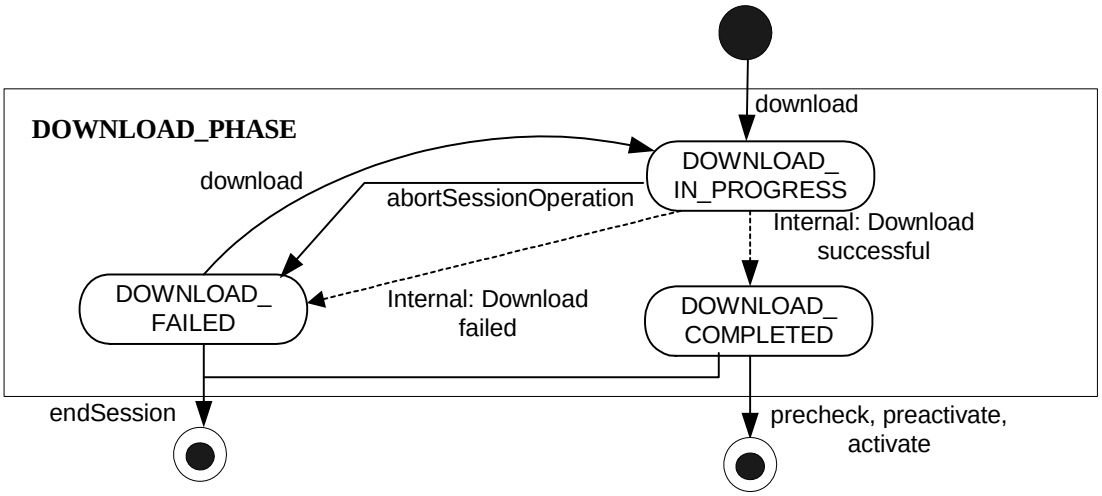


Figure 9.2.2: Substate Machine - DOWNLOAD_PHASE

9.2.3 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 IRPAgent. (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.

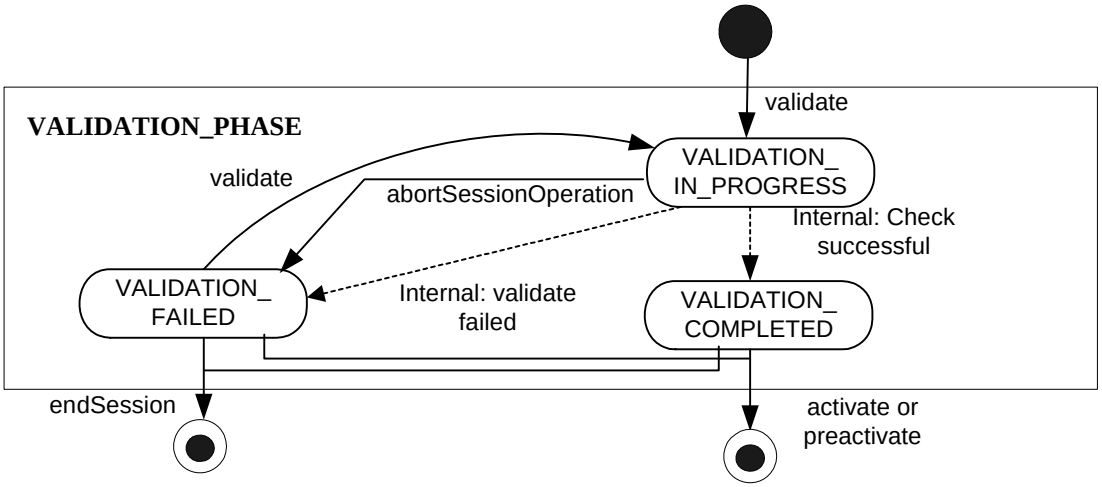


Figure 9.2.3: Substate Machine - VALIDATION_PHASE

9.2.4 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.

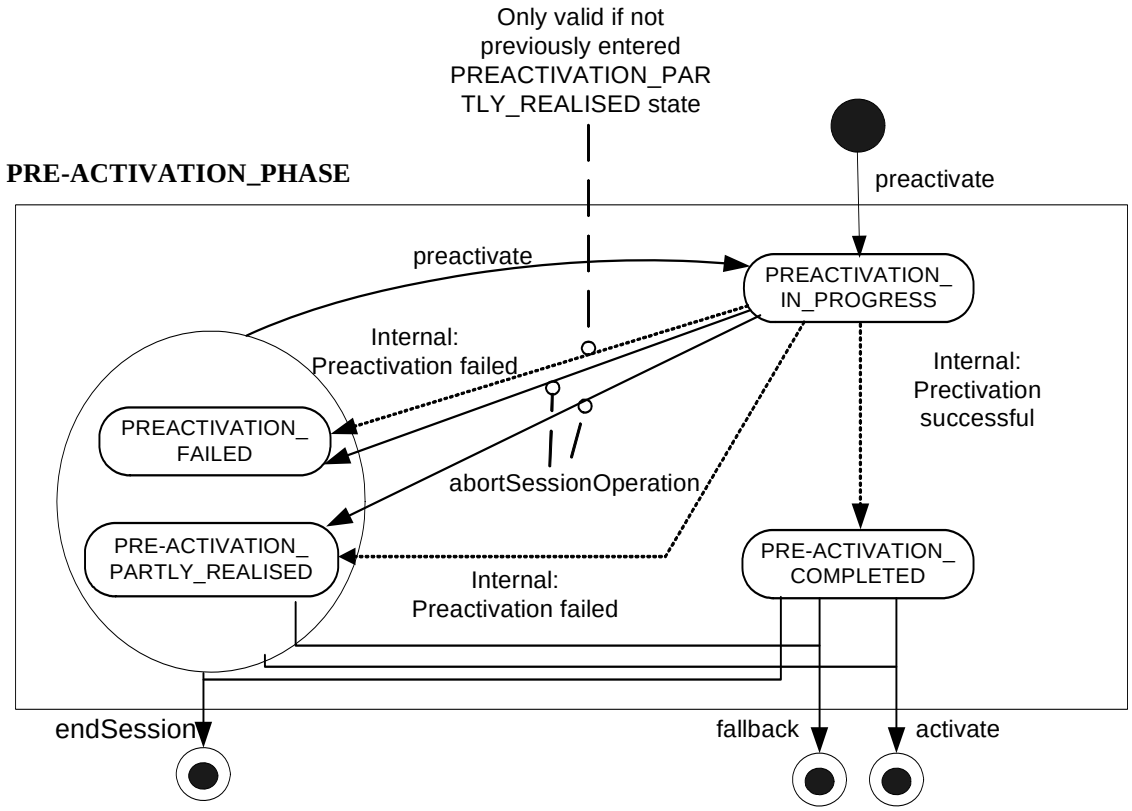


Figure 9.2.4: Substate Machine - PREACTIVATION_PHASE

9.2.5 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.

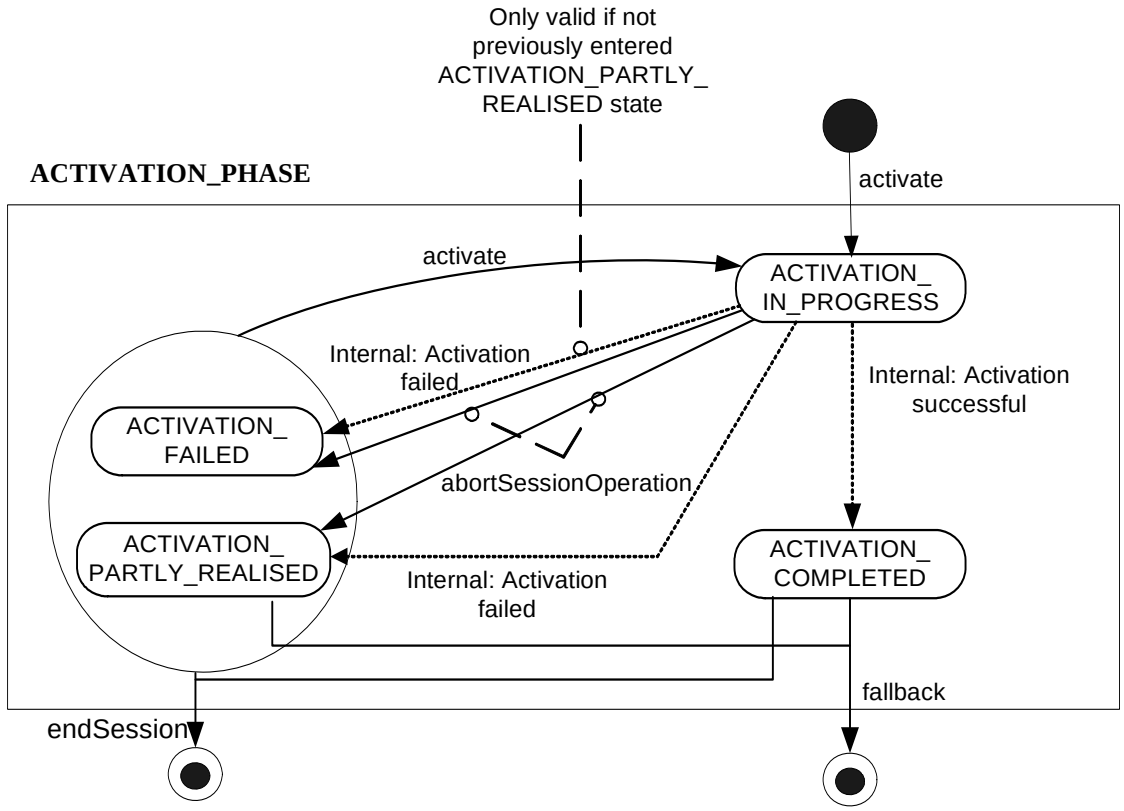


Figure 9.2.5: Substate Machine - ACTIVATION_PHASE

9.2.6 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 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.

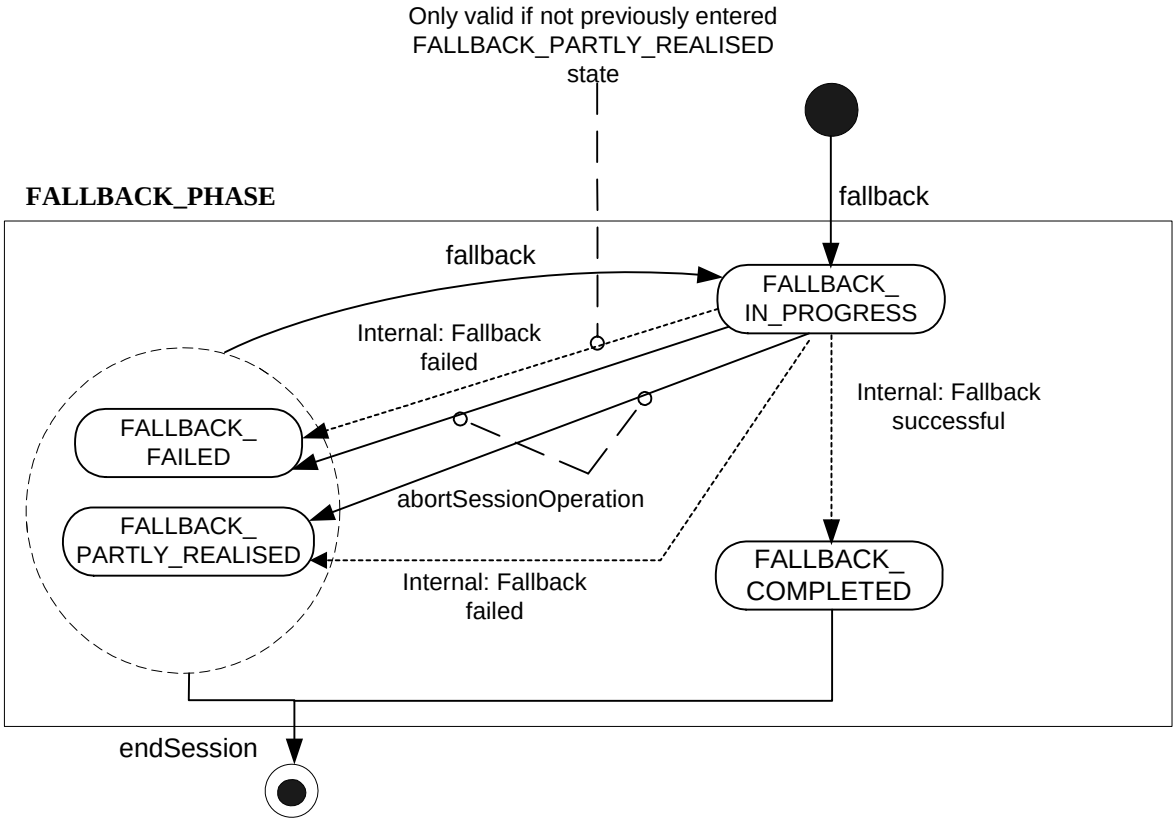


Figure9.2.6: Substate Machine – FALLBACK_PHASE

9.3 State Machine Pre and Post Conditions Tables

For each operation the following tables identify the state machine pre and post conditions.

Table 9.3: 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	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 9.3a: State Machine Pre and Post Conditions (Simple Upload)

Operation	Pre-condition	Post Condition
upload	No state – input sessionId provided by an IRPManager is not already in use in the IRPAgent by this or any other IRPManager	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

10 Bulk Configuration Data File

The overall management of Bulk CM is controlled by the operations in clause 7. Unitary management information is aggregated into a configuration data file for bulk CM operations. The file can be used for active and passive CM.

Bulk configuration data files consist of one or more blocks. Each block contains one or more object containment trees defined by a standardised language, for example XML. The basic building block (node) of this tree is a specifically typed MO. This MO is identified by an ID attribute (the Naming attribute used in the RDN), and contains (1) data associated with the MO, and (2) zero or more children nodes. The structure and content of the MO data is constrained by the possible types of contained objects for the CM NRM that is being managed by Bulk CM IRP IS.

The file structure is the same for both upload and download bulk CM operations, apart that for active bulk CM operations, as well as containing MO data the blocks also specify the management actions (sub-operations) associated with each MOs item in the file. The following management actions (sub-operations) on MOs are supported for active bulk CM:

- Create MO. (clause 10.1.1)
- Delete MO. (clause 10.1.2)
- Change one or more existing MO attribute values. (clause 10.1.3)

The rules for ordering management actions in the configuration data file are defined in clause 10.2.

10.1 Bulk Configuration Data Management Actions – Sub-operations

By the nature of active Bulk CM IRP, in the download bulk configuration file all sub-operation parameters identified in the following clauses 10.1.1 – 10.1.3 are "input " only. Bulk CM IRP:IS will not generate any explicit notifications or responses for each sub-operation. The resulting session log and output(s) from the associated Bulk CM operations will record and convey the overall result of the sub-operations in the bulk configuration data file. The IRPAgent can record the outcome of relevant sub-operations in the session log. The IRPManager can subsequently get the session log (clause 7.3.6) if it is required to make a detailed analysis.

It should be noted other IRPs can generate notifications as a result of Bulk CM: IS sub-operations if an IRPAgent implements Basic CM IRP. The rules and definitions for these notifications are beyond the scope of the present document. The NRMs identified in clause 6.4 and references [4], [5] and [6] give further details of which MOCs may generate Basic CM IRP notifications as a consequence of the sub-operations defined here.

10.1.1 bulkCmCreateMo (Create MO Sub-operation) (M)

The IRPManager associates this sub-operation with an MOI in the configuration data file to request the IRPAgent to create the MOI.

Table 10.1.1: bulkCmCreateMo parameters

Name	Qualifier	Description
objectClass	Input, M	Identifies the NRM MOC within the scope of clause 6.4 that is to be created.
objectInstance	Input, M	Identifies the NRM MOC instance that is to be created.
attributeList	Input, O	Empty, or one or more attribute name and value pairs valid for the MOC. See clause 6.4. If the list is not empty the indicated attributes will be set to their indicated values when the object is created.

10.1.2 bulkCmDeleteMo (Delete MO Sub-operation) (M)

The IRPManager associates this sub-operation with an MOI in the configuration data file to request the IRPAgent to delete the MOI.

Table 10.1.2: bulkCmDeleteMo parameters

Name	Qualifier	Description
objectClass	Input, M	Identifies the NRM MOC within the scope of clause 6.3 that is to be deleted.
objectInstance	Input, M	Identifies the NRM MOC instance that is to be deleted.

10.1.3 bulkCmChangeMo (Change MO Sub-operation) (M)

The IRPManager associates this sub-operation with an MOI in the configuration data file to request the IRPAgent to change/set one or more attributes of the MOI.

Table 10.1.3: bulkCmChangeMo parameters

Name	Qualifier	Description
objectClass	Input, M	Identifies the NRM MOC within the scope of clause 6.4 that the attributes are to be changed.
objectInstance	Input, M	Identifies the NRM MOC instance for which the attributes are to be changed.
attributeList	Input, M	One or more attribute name and value pairs valid for the MOC. See clause 6.3. The indicated attributes of the MOC instance will be changed / set to their indicated values.

10.2 Rules for ordering Management Actions (Sub-operations) in Configuration Data Files

10.2.1 Download files

1. The IRP Manager shall enter the management actions into the configuration data file in the order they are to be interpreted and actioned by the IRPAgent following its sequentially step-by-step single pass operation. The IRPManager has overall responsibility for ensuring the correct order of action is given according to the rules in this clause.

2. The IRPAgent shall interpret the management actions in the configuration data file sequentially step-by-step in a single pass operation. The IRPManager has overall responsibility for ensuring the correct order of action is given.

3. The permitted order shall follow NRM hierarchy subtree(s) of the Managed Object instances pertaining to the configuration data file.

4. All delete MOs actions shall precede any Create MOs actions.

5. The present document does not specify any limitations on the ordering of change MO attribute actions other than the impacted if the impacted MO does not already exist it needs to be created by a prior create action. The choice of standardised language may recommend or specify some additional constraints e.g. for reasons of efficiency or for compliance with language syntax. Such recommendation and constraints are beyond the scope of the present document

6. All necessary MO changes supported by Bulk CM IRP interface-N need to be fully specified in a configuration data file to maintain consistency within the NRM MIB subtree being operated on. (e.g. if an object is to be deleted, all relations and associations shall be removed).

7. All relations to an MO instance shall be removed prior to deleting an MO instance.

8. When part or whole NRM subtree is to be deleted, in the configuration data file the IRPManager shall first action delete of all associated child instances contained in the NRM subtree before actioning delete of MO parents instances i.e. delete actions on MO instances shall be specified in a recursive manner following the NRM

hierarchy subtree from the lowest MO instances to the highest MO instances the IRPManager requires to be deleted. (The IRPAgent will not support autonomous deletion of all MO instance contained in a NRM subtree identified by a single delete action of the highest MO instance of the subtree).

9. When part or a whole NRM subtree is to be created, in the configuration data file the IRPManager shall first action the create action of parents MO instances before actioning the create of any child MO instances contained in the NRM subtree i.e. create actions on MO instances shall be specified in recursive manner following the NRM hierarchy subtree from the highest MO instances to the lowest MO instances the IRPManager requires to be created.

10.2.2 Upload files

1. No rules are identified i.e. it is not necessary that they be part of the scope of the present document. They may be implementation specific and specified in other document as part of a specific solution.

Annex A (informative): Scenarios

Supporting background informational only.

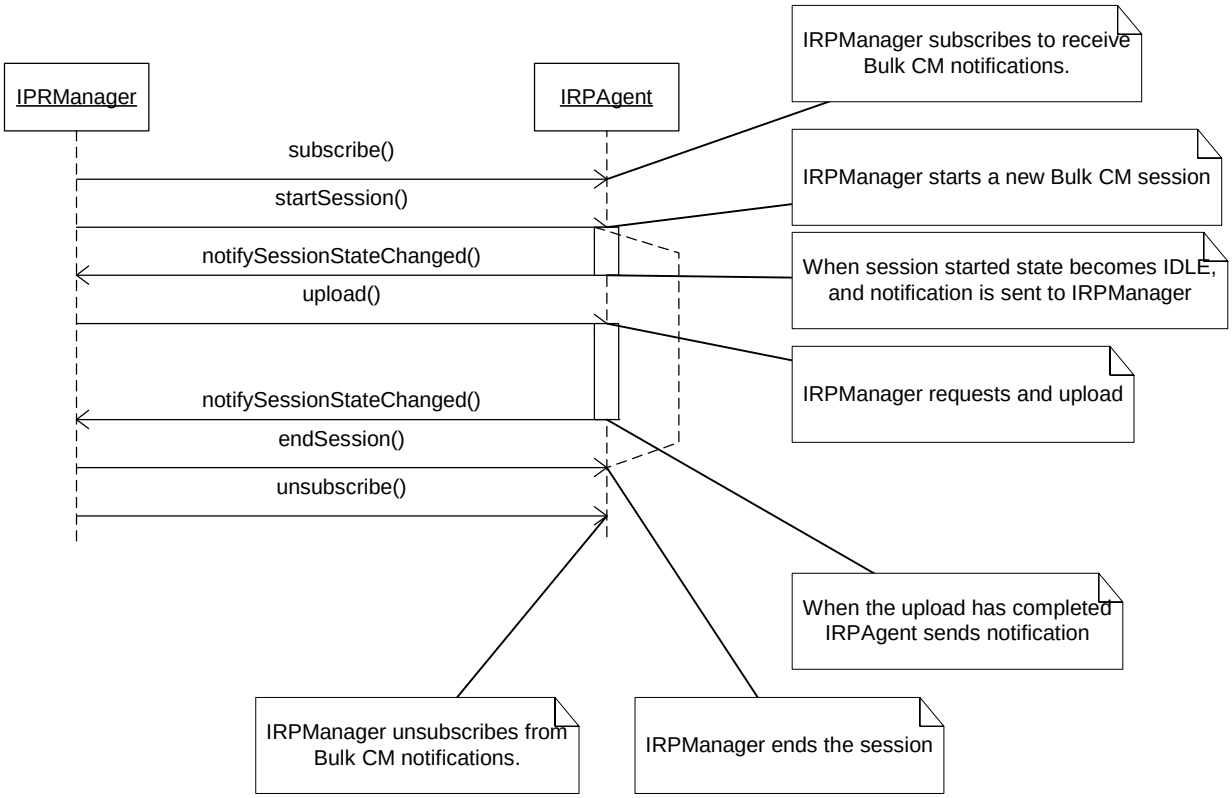


Figure A.1: Example 1: Successful Upload Session

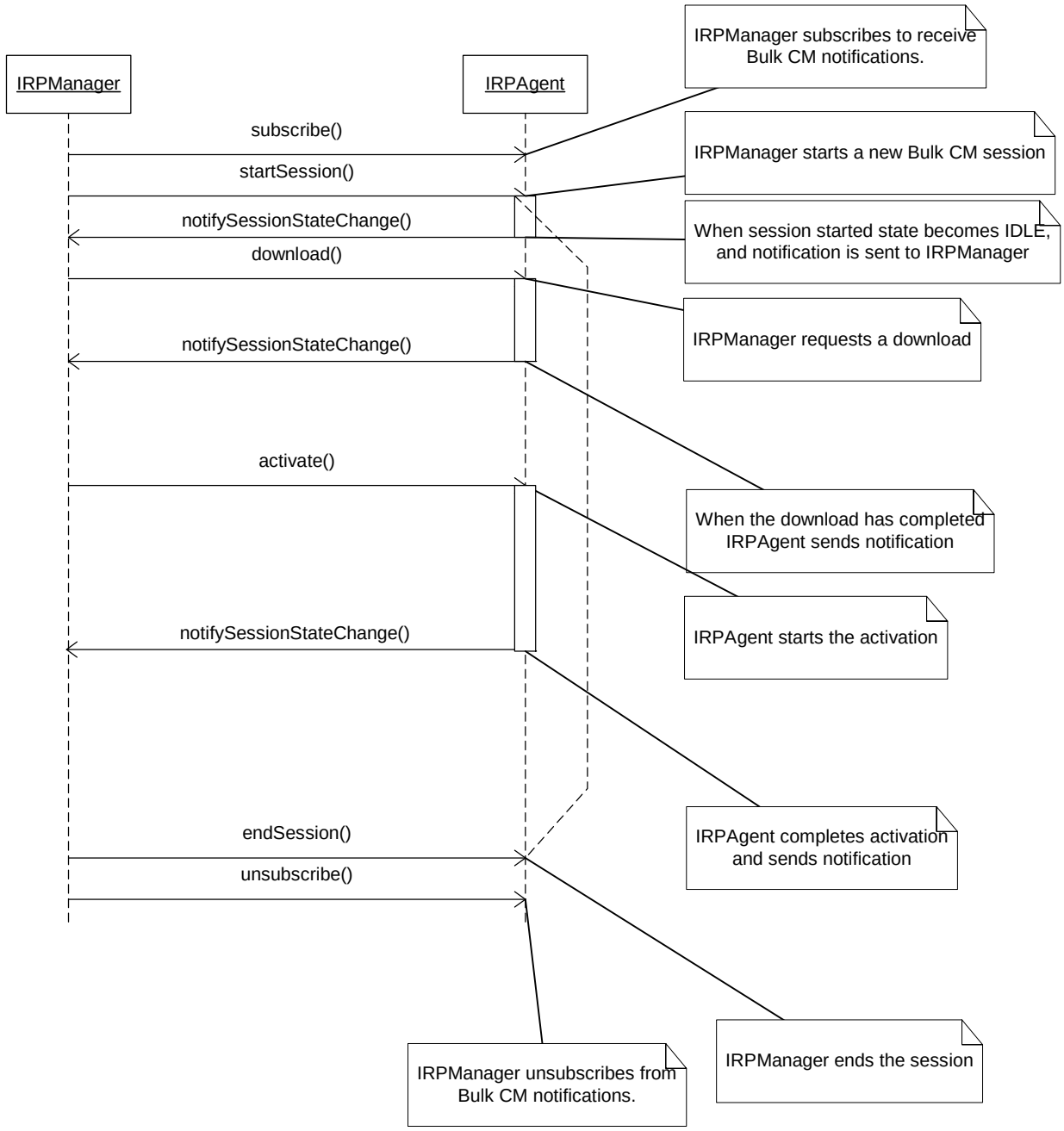


Figure A.2: Example 2: Successful Download and Activation without validation and preactivation

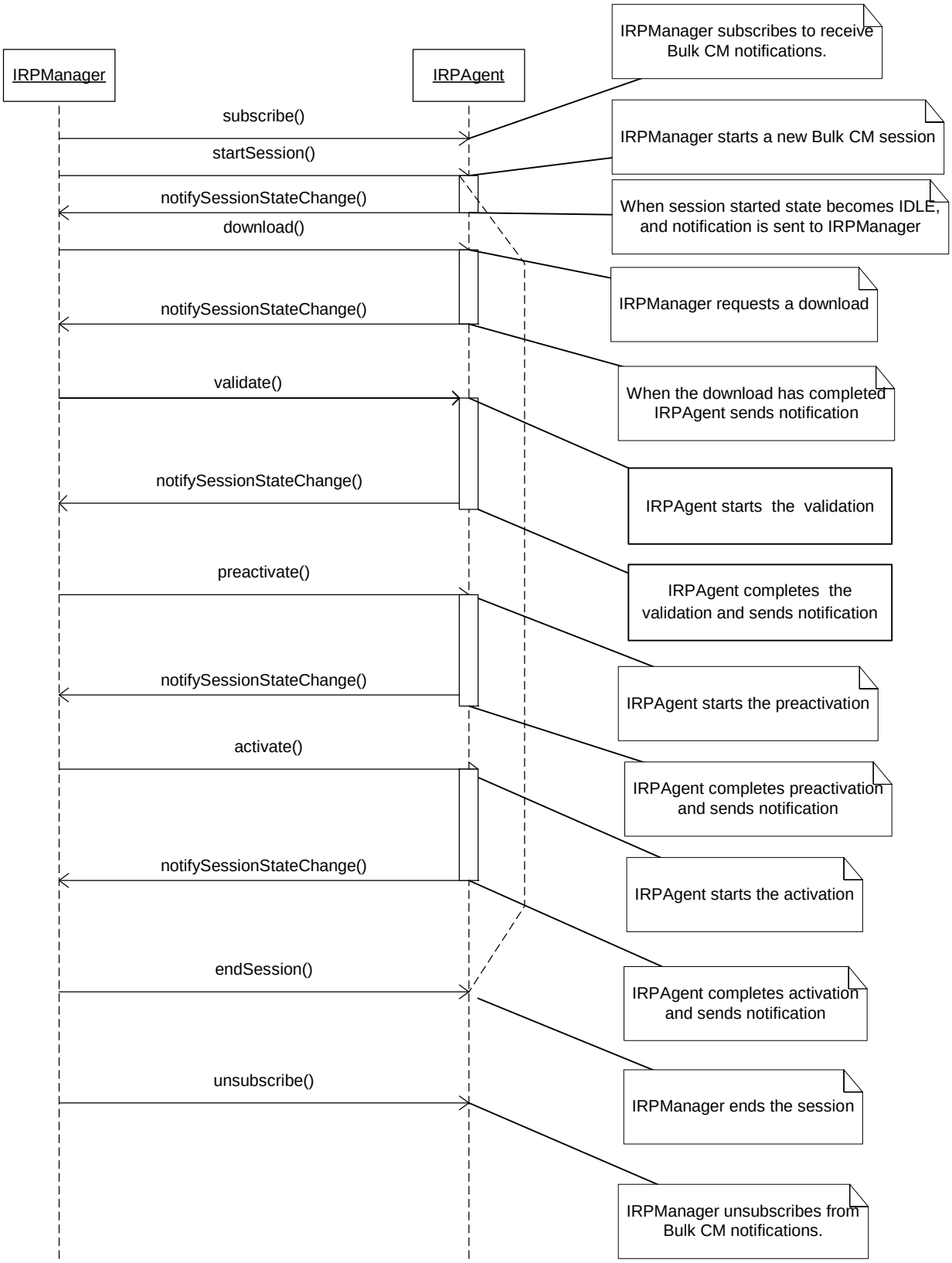


Figure A.3: Example 3: Successful Download and Activation with validation and preactivation

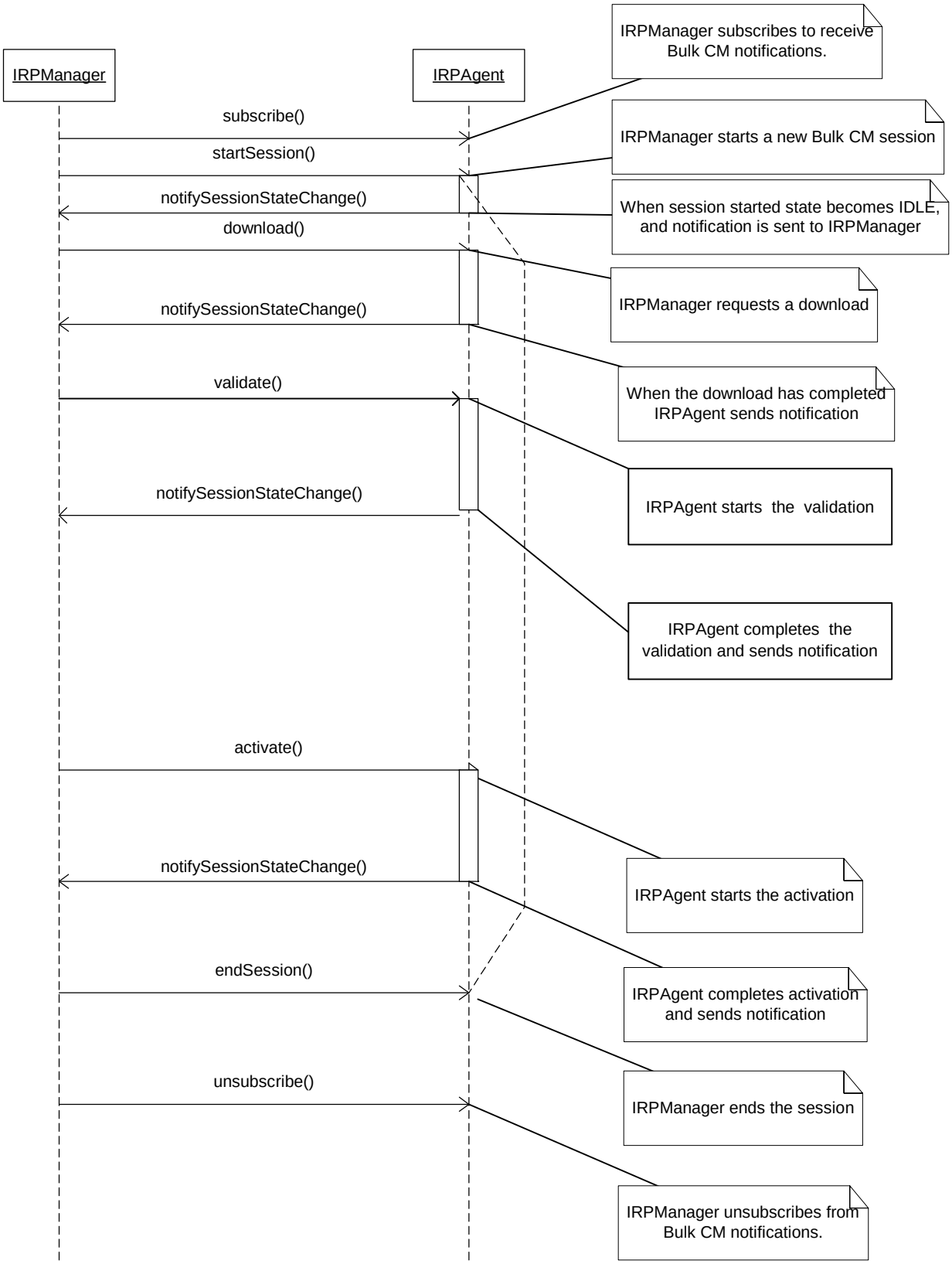


Figure A.4: Example 4: Successful Download and Activation with Validation

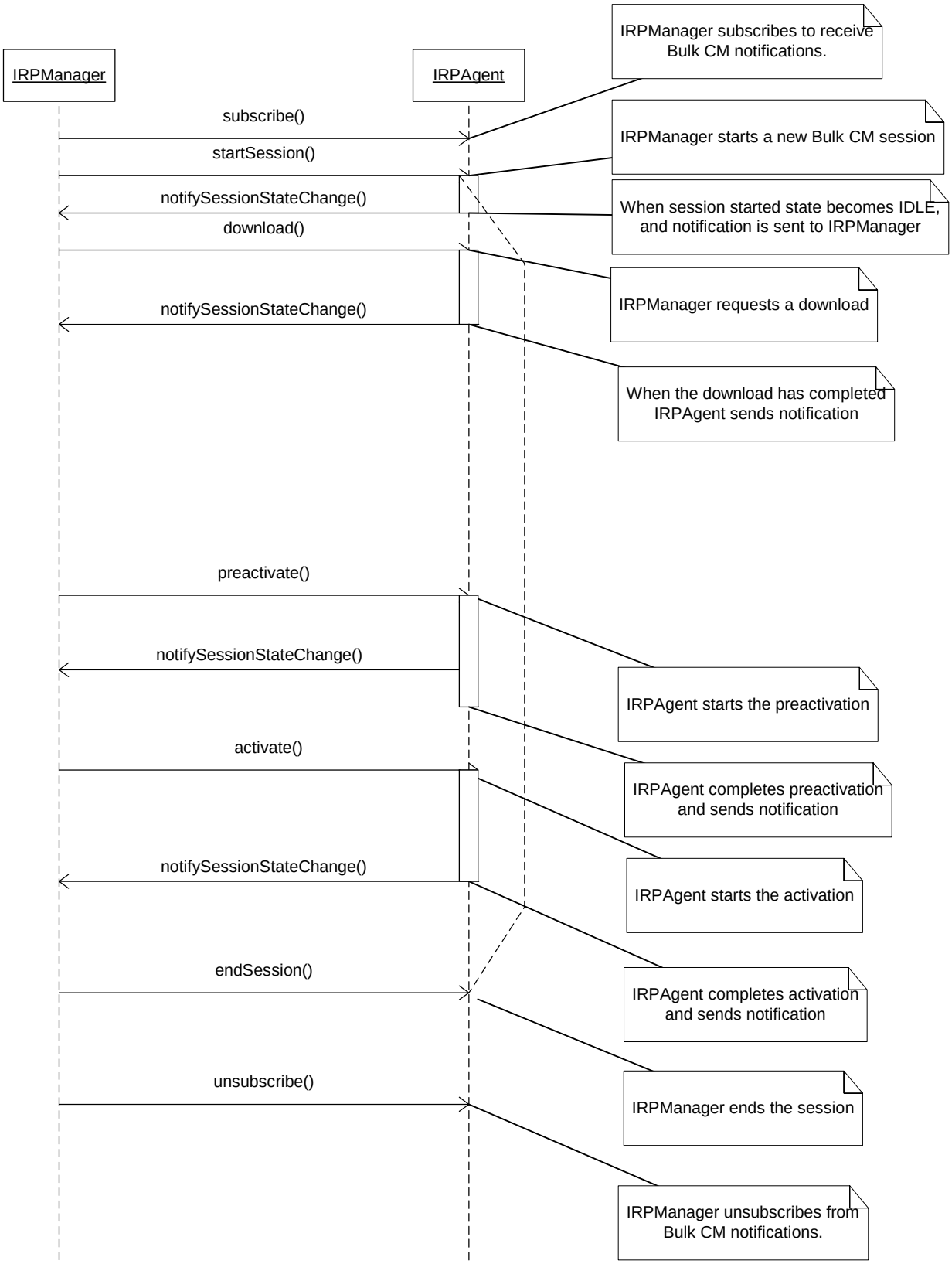


Figure A.5: Example 5: Successful Download and Activation with Preactivation

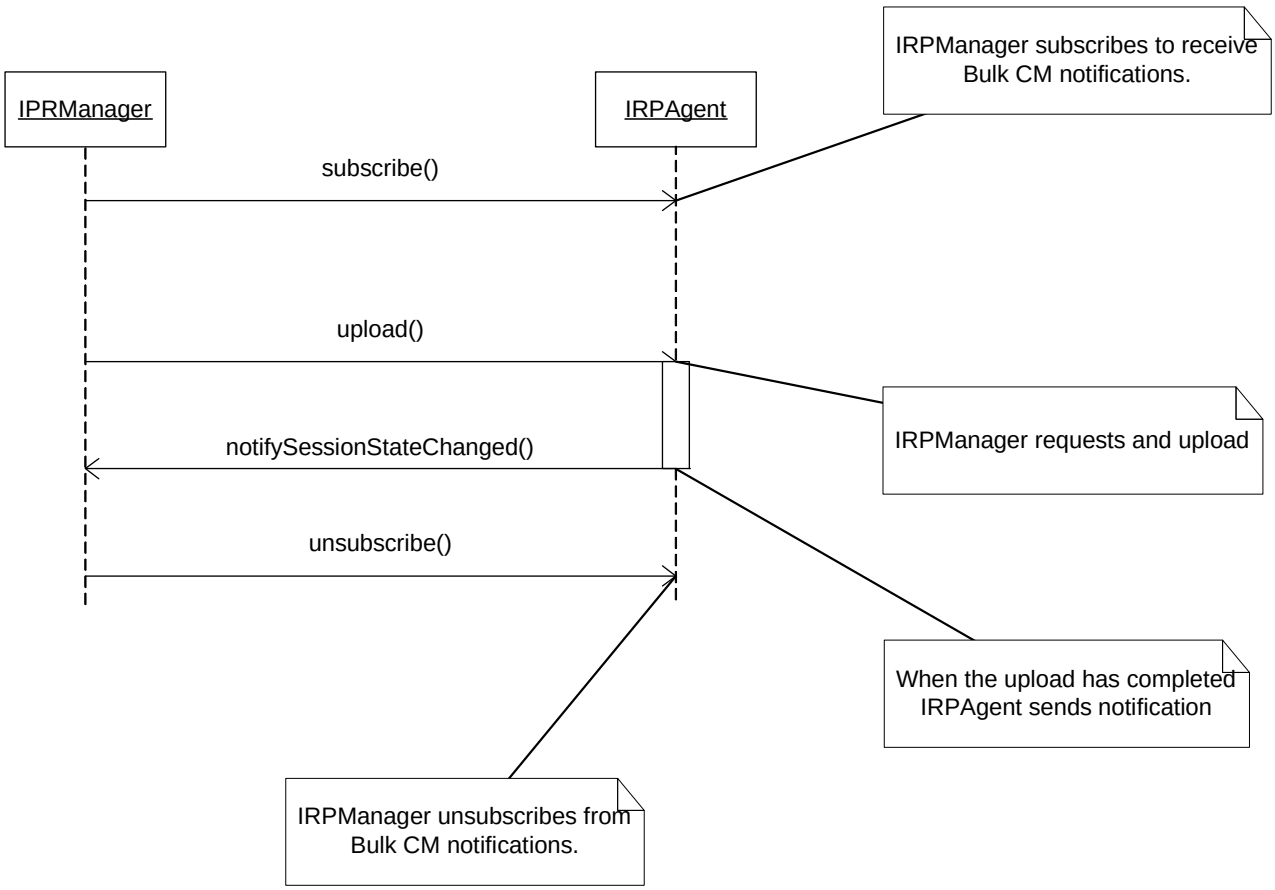


Figure A.6: Example 6: Successful Upload Session – Simple Upload

Annex B (informative): Bulk CM Application and Operation Principles

B.1 Key characteristics

1. Bulk CM operations are not transaction based.
2. The state machine does not allow looping. Can only progress forward through main states.
3. If any errors are found in the configuration data, it should not be possible to revise the configuration data during a session e.g. to try to resolve any problems found during a session. A new session should be started with a new corrected version of the configuration data being applied.
4. Non-transitional interface;
5. Sessions may be run in parallel. There should not be any exclusion of specified changes between parallel sessions.

Annex C (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Jun 2001	SA_12	SP-010283	--	--	Approved at TSG SA #12 and placed under Change Control	--	2.0.0	4.0.0
Sep 2001	SA_13	SP-010479	0001	--	Correction of State Machine Pre and Post Conditions	F	4.0.0	4.1.0
Jun 2002	SA_16	SP-020296	0002	--	Correction of behaviour for IS parameter "saveFallback" of IS operation "activate"	F	4.1.0	4.2.0
Sep 2002	SA_17	SP-020484	0003	--	Correction of pre- and post-conditions for the operations getSessionStatus and getSessionLog	F	4.2.0	4.3.0
Sep 2002	SA_17	SP-020486	0003	--	Add Bulk CM IRP IS Enhancements for Rel-5	C	4.3.0	5.0.0
Dec 2002	SA_18	SP-020744	0006	--	Incomplete getSessionStatus	A	5.0.0	5.1.0
Mar 2003	--	--	--	--	Editorial (Clause heading missing: 8 Bulk Configuration Data File)	--	5.1.0	5.1.1
Dec 2003	SA_22	SP-030630	0008	--	Correction of System Context	A	5.1.1	5.2.0
Mar 2004	SA_23	SP-040119	0010	--	Correction of System Context	A	5.2.0	5.3.0
Mar 2004	SA_23	SP-040105	--	--	Automatic upgrade to Rel-6 (no CR)	--	5.3.0	6.0.0
Dec 2004	SA_26	SP-040807	0011	--	Partition Bulk CM IRP capabilities into packages	F	6.0.0	6.1.0
Dec 2004	SA_26	SP-040807	0012	--	BulkCMIRP should be extended to be applicable to new NRM model, such as Signalling Transport Network (STN) NRM IRP	B	6.0.0	6.1.0
Mar 2005	SA_27	SP-050045	0013	--	Apply Generic System Context	F	6.1.0	6.2.0
Mar 2005	SA_27	SP-050045	0014	--	Add missing reference to TS 32.712 Transport Network NRM	F	6.1.0	6.2.0
Mar 2005	SA_27	SP-050045	0015	--	Correct Annex B text style to comply with drafting rules	F	6.1.0	6.2.0
Jun 2005	SA_28	SP-050295	0017	--	Correction of ambiguous precondition statement related to fallback operation	A	6.2.0	6.3.0
Sep 2005	SA_29	SP-050450	0019	--	Correct state machine diagrams and pre-condition box to reflect the correction of fallback pre-condition	A	6.3.0	6.4.0
Sep 2006	SA_33	SP-060534	0020	--	Correct the class diagrams and the qualifiers of systemDNs in the "Bulk Configuration Management (CM) Integration Reference Point (IRP)"	F	6.4.0	6.5.0
Sep 2006	SA_33	SP-060534	0022	--	Add missing Notification subclause in Bulk CM IRP IS	F	6.4.0	6.5.0
Sep 2006	SA_33	SP-060553	0021	--	Correct the qualifiers of objectClasses and objectInstances	F	6.5.0	7.0.0
Dec 2006	SA_34	SP-060730	0023	--	Include IMS NRM IRP in the scope for Bulk CM IRP	C	7.0.0	7.1.0
Mar 2007	SA_35	SP-070046	0024	--	Add missing information to support file compression	F	7.1.0	7.2.0
Mar 2009	SA_43	SP-090207	0025	--	Include reference to SOAP Solution Set specification	D	7.2.0	8.0.0
Dec 2009	SA-46	SP-090718	0026	--	Clarify ScopeType Definition for BulkCM IRP	F	8.0.0	8.1.0
Dec 2009	-	-	-	-	Update to Rel-9 version	-	8.1.0	9.0.0

History

Document history		
V9.0.0	February 2010	Publication