

ETSI TS 132 612 V4.2.0 (2002-06)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
Telecommunication management;
Configuration management;
3G configuration management:
Bulk configuration management
Integration Reference Point (IRP): Information service
(3GPP TS 32.612 version 4.2.0 Release 4)**



Reference

RTS/TSGS-0532612v420

Keywords

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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.fr

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2002.
All rights reserved.

DECTTM, **PLUGTESTS**TM and **UMTS**TM are Trade Marks of ETSI registered for the benefit of its Members.
TIPHONTM and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

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 www.etsi.org/key.

Contents

Intellectual Property Rights	2
Foreword.....	2
Foreword.....	5
Introduction	5
1 Scope	6
2 References	6
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	8
4 System Overview	9
4.1 System Context	9
4.2 Compliance rules.....	9
4.3 Scope of Bulk CM Management Specification	10
5 Modelling approach.....	10
6 IRP Information Service.....	10
6.1 Introduction	10
6.2 IRP Information Service	10
6.2.1 Interfaces.....	11
6.2.2 Bulk CM Operations	11
6.2.2.1 startSession (M)	12
6.2.2.2 endSession (M)	12
6.2.2.3 upload (M).....	12
6.2.2.4 download (M).....	13
6.2.2.5 activate (M)	13
6.2.2.6 fallback (M).....	14
6.2.2.7 abortSessionOperation (M)	14
6.2.2.8 getSessionIds (M)	14
6.2.2.9 getSessionStatus (M).....	15
6.2.2.10 getSessionLog (M).....	15
6.2.2.11 getBulkCmIRPVersion (M)	15
6.2.3 Configuration File Notifications	16
6.2.3.1 General	16
6.2.3.2 notifySessionStateChanged (M).....	16
6.2.3.3 NotifyGetSessionLogEnded (M).....	17
6.3 Network Resource Model (NRM).....	17
7 State Machine	17
7.1 State Machine Overview	17
7.2 State Machine Description	18
7.2.1 Upload Phase	19
7.2.2 Download Phase	20
7.2.3 Activation Phase	20
7.2.4 Fallback Phase	21
7.3 State Machine Pre and Post Conditions Tables	22
8 Bulk Configuration Data File	23
8.1 Bulk Configuration Data Management Actions – Sub-operations	24
8.1.1 bulkCmCreateMo (Create MO Sub-operation) (M)	24
8.1.2 bulkCmDeleteMo (Delete MO Sub-operation) (M)	24
8.1.3 bulkCmChangeMo (Change MO Sub-operation) (M).....	24
8.2 Rules for ordering Management Actions (Sub-operations) in Configuration Data Files	25
8.2.1 Download files.....	25

8.2.2 Upload files.....25

Annex A (informative): Scenarios26

Annex B (informative): Change history28

History29

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

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 an Integration Reference Point (IRP) through which an 'IRPAgent' (typically an Element Manager or Network Element) can communicate bulk Configuration Management related information to one or several 'IRPManagers' (typically Network Managers).

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

- [1] 3GPP TS 32.101: "3G Telecom Management principles and high level requirements".
 - [2] 3GPP TS 32.102: "3G Telecom Management architecture".
 - [3] 3GPP TS 32.302: "Telecommunication Management; Notification Management; Part 2: Notification Integration Reference Point; Information Service".
 - [4] 3GPP TS 32.622: "3G Configuration Management: Generic Network Resources IRP: NRM".
 - [5] 3GPP TS 32.642: "3G Configuration Management: UTRAN Network Resources IRP: NRM".
 - [6] 3GPP TS 32.652: "3G Configuration Management: GERAN Network Resources IRP: NRM".
 - [7] 3GPP TS 32.300: "Name Convention for Managed Objects".
 - [8] 3GPP TS 32.600: "3G Configuration Management: Concepts and requirements".
-

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).

IRP: See 3GPP TS 32.101 [1].

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 consists 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. The following figure depicts the relationships between a Name space and a number of participating MOs (the shown association is of a non-containment type)

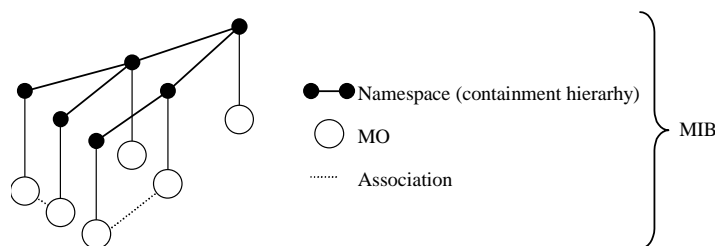


Figure 1: Relationships between a Name space and a number of participating MOs

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
CMIP	Common Management Information Protocol
CORBA	Common Object Request Broker Architecture
EM	Element Manager
FM	Fault Management
IRP	Integration Reference Point
ITU-T	International Telecommunication Union, Telecommunication Standardisation Sector
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
SS	Solution Set
SW	Software
TM	Telecom Management
UML	Unified Modelling Language (OMG)
UMTS	Universal Mobile Telecommunications System
XML	EXtensible Markup Language

4 System Overview

4.1 System Context

Figure 2 and-Figure 3 identify system contexts of the subject IRP in terms of its implementation called IRPAgent and the user of the IRPAgent, called IRPManager. For a definition of IRPManager and IRPAgent, see 3GPP TS 32.102 [2].

The IRP Agent implements and supports the Bulk CM IRP. The IRP Agent shall be an Element Manager (EM) or a mediator that interfaces to several NE (see Figure 2) or it can be a Network Element (NE) (see Figure 3). In the former case, the interfaces (represented by the a thick dotted line) between the EM and the NEs are not subject of this IRP.

An IRPManager using this IRP shall choose one of the two System Contexts defined here, for each NE. For instance, if an EM is responsible for managing a number of NEs, the NM shall access this IRP through the EM and not directly to those NEs. For another IRP though, the System Context may be different. For Bulk CM IRP its judged System A in most application is most appropriate, but this does not preclude use of System B when the need is appropriate.

For another IRP the System Context may be different.

As indicated in Figure 2 and Figure 3, the subject IRP needs to be complemented with the Notification IRP 3GPP TS 32.302 [3]. (This is to allow the IRP Manager to subscribe and unsubscribe to notifications issued by the IRP Agent).

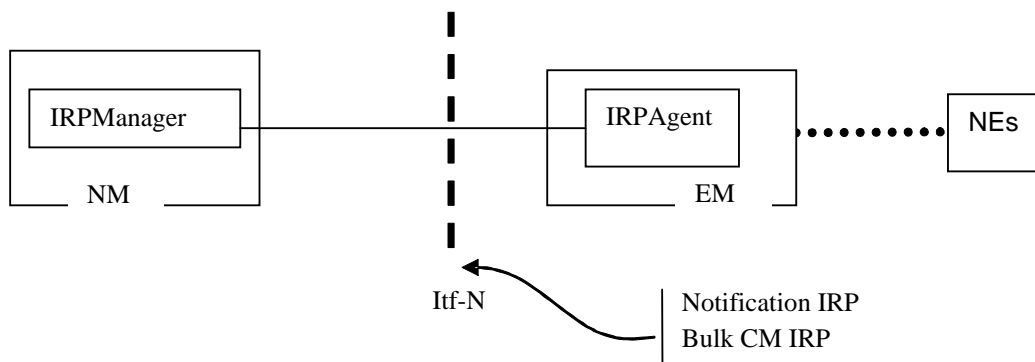


Figure 2: System Context A

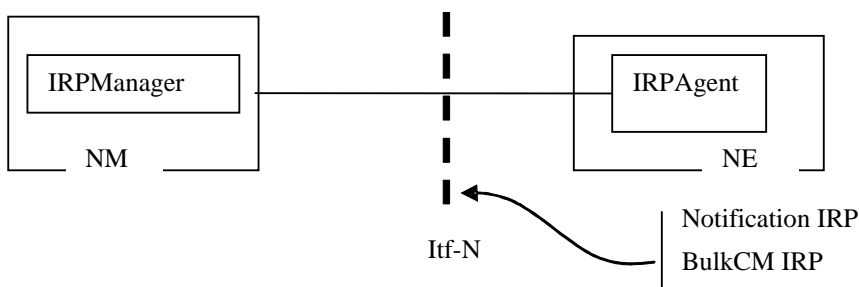


Figure 3: System Context B

4.2 Compliance rules

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

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

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

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

Given that

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

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

4.3 Scope of Bulk CM Management Specification

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

5 Modelling approach

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

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

6 IRP Information Service

6.1 Introduction

As already introduced in the previous clause, the present clause defines the Bulk CM IRP Information Service in the form of the IRP Information Service.

The corresponding Solution Set and Data Format documents provide protocol dependent object model solutions. They provide the actual realization of the operations and notifications defined in this subclause in each protocol environment. One may find that the operation names and operation parameters defined in this protocol-neutral model differ from those defined in the Solution Sets.

6.2 IRP Information Service

This subclause specifies the *operations* and *notifications* that are visible over this IRP. These operations are generic in the sense that they do not specify the MOs that are retrieved/manipulated over the interface.

6.2.1 Interfaces

Figure 5 illustrates the operations and notifications defined as interfaces implemented and used by IRP Agent and IRPManager, described using UML notation (Interface in IRP Information Model is identical to concepts conveyed by stereotype <<interface>> of UML). Parameters and return status are not indicated.

Two interfaces are defined. One is called BulkCmIRPOperations. This interface defines operations implemented by IRP Agent and used (or called) by IRPManager. The other is called BulkCmIRPNotifications. This interface defines notifications implemented by IRPManager and used by IRP Agent.

The interfaces support multiple IRPManagers connected to an IRP Agent.

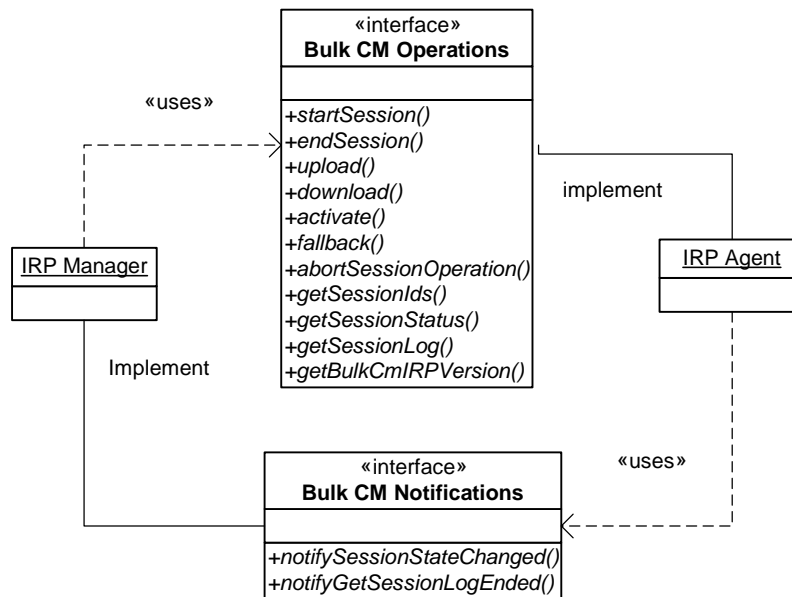


Figure 4: UML Interface Class Diagram

6.2.2 Bulk CM Operations

Configuration files defined in clause 8 define bulk configuration management changes. The following configuration file handling operations exist in the Itf-N.

- startSession
- endSession
- upload
- download
- activate
- fallback
- abortSessionOperation
- getSessionIds
- getSessionStatus
- getSessionLog

- getBulkCmIRPVersion

Notification IRP [3] related operations are also associated with Bulk CM IRP (e.g. Subscribe an Unsubscribe), but these operations are described in 32.302: "Telecommunication Management; Notification Management: Part 2: Notification IRP; Information Service " [3].).

The operations, upload, download, activate, fallback and getSessionLog are performed asynchronously in that when the operations are initiated, 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 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 subclause 4.3 for system conditions that need to be potentially managed, but are outside the scope of this document.

6.2.2.1 startSession (M)

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.

Table 1: startSession parameters

Name	Qualifier	Description
SessionId	Input, M	Identifies the new session and process to be associated with a bulk data operation e.g. upload or download.
Status	Output, M	indicates (a) operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.2 endSession (M)

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. The deletion will be rejected if the configuration state is in a working state: e.g. uploading (including getting a log), downloading or activating.

Table 2: endSession parameters

Name	Qualifier	Description
sessionId	Input, M	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download
status	Output, M	indicates (a) operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.3 upload (M)

An IRPManager invokes this operation to request the IRPAgent to create a file containing bulk configuration data (clause 8) and transfer the file to the indicated globally unique data file reference.

Table 3 : upload parameters

Name	Qualifier	Description
sessionId	Input, M	Identifies this specific session and process associated with the requested bulk data upload.
uploadDataFileReference	Input, M	This specifies a globally unique file reference to where the specified scope of bulk data is to be uploaded and stored .
BaseObjectInstance	Input, M	The MO where the search starts. This is a full Distinguished Name according to 3GPP TS 32.300 [7].
scope	Input, M	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: the base object alone (default); the n-th level subordinates of the base object; the base object and all of its subordinates down to and including the n-th level; the base object and all of its subordinates.
filter	Input, M	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.
status	Output, M	indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.4 download (M)

An IRPManager invokes this operation to request an IRPAgent to download and administer a file containing bulk configuration data (clause 8). The IRPAgent obtains the configuration file data from the indicated globally unique data file reference.

Table 4: download parameters

Name	Qualifier	Description
sessionId	Input, M	Identifies this specific session and process associated with the requested bulk data download.
downloadDataFileReference	Input, M	This specifies a globally unique file reference from where the data to be fetched and download from.
status	Output, M	indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.5 activate (M)

An IRPManager invokes this operation to request an IRPAgent to activate previously downloaded bulk configuration data (clause 8). Activate means that operations specified in a previously downloaded configuration 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.

Enabling the fallback option is optional. There can only be one choice for a session: enabled or not enabled. If enabling the fallback option is selected it shall be initiated when the first activation operation is requested. If enabling the fallback option is not requested for the first activation, it cannot be subsequently requested for repeated activations during the session. If enabling the fallback option was requested, it is not possible change this choice with any subsequent re-activate retries i.e. for a session it is only possible to fallback to the configuration that existed when the first activate operation was requested. See also subclause 6.2.2.6. (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 this document. Only the IRPManager can initiate activate retries. (The IRPAgent shall not initiate retries autonomously).

Table 5: activate parameters

Name	Qualifier	Description
sessionId	Input, M	Identifies this specific session and process associated with an earlier bulk data download that is required to be activated.
fallbackEnabled	Input, M	Indicates whether or not the fallback option is required to be enabled. If required to be enabled, the first activate operation of a session that gets properly initiated in the IRPAgent shall initialise and enable fallback option prior to the activation. Enabling or not the fallback option is only open for the first activate operation of a session. For any subsequent activate operation retries within a session the fallbackEnabled parameter shall have the same value as for the first activate operation otherwise the re-activate operation shall be rejected.
status	Output, M	indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.6 fallback (M)

An IRPManager invokes this operation to request an IRPAgent to activate a fallback area if a previously ordered activation has failed.

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 this 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 operations was performed with fallback option enabled. For further discussion of enabling or not the fallback option see subclause 6.2.2.5.

Table 6 : fallback parameters

Name	Qualifier	Description
sessionId	Input, M	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.
status	Output, M	indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.7 abortSessionOperation (M)

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 7.

Table 7: abortSessionOperation parameters

Name	Qualifier	Description
sessionId	Input, M	Identifies this specific session and process associated with an earlier bulk data operation e.g. upload or download for which the abort is required.
status	Output, M	indicates (a) start of abort operation is successful and (b) abort operation failed because of specified or unspecified reasons

6.2.2.8 getSessionIds (M)

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

Table 8: getSessionIds parameters

Name	Qualifier	Description
sessionIdList	Output, M	A list of all the sessionIds an IRPAgent currently has open i.e. started with startSession and not ended with endSession operations.
status	Output, M	indicates (a) operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.9 getSessionStatus (M)

The IRPManager invokes this operation to request the IRPAgent to send the current state of the bulk data configuration file operation. The IRPAgent returns the current state. See clause 7.

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

Table 9: getSessionStatus parameters

Name	Qualifier	Description
sessionId	Input, M	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.
sessionState	Output, M	Indicates current state of the configuration file operation. See clause 7, i.e. will be one of: Upload In Progress, Upload Failed, Upload Completed, Down Load In Progress, Download Failed, Download Completed, Activation In Progress, Activation Failed, Activation Partly Realised, Activation Completed, Fallback In Progress, Fallback Failed, Fallback Partly Realised, Fallback Completed,
status	Output, M	Indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.10 getSessionLog (M)

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.

Table 10: getSessionLog parameters

Name	Qualifier	Description
sessionId	Input, M	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	Input, M	Specifies the address and file name where the result is to be placed in the IRPManager.
contentType	Input, M	Identifies if retrieved file should include (a) complete log including errors, (b) only errors.
status	Output, M	Indicates (a) start of operation is successful and (b) operation failed because of specified or unspecified reasons

6.2.2.11 getBulkCmIRPVersion (M)

IRPManager invokes this operation when it wishes to find out the Bulk CM IRP SS versions supported by IRPAgent. IRPAgent shall respond with a list of supported Bulk CM IRP SS versions.

Table 11: Parameters of getBulkCmIRPVersion

Name	Qualifier	Description
VersionNumberList	Output, M	It indicates one or more SS version numbers supported by the IRPAgent.
status	Output, M	Operation succeeded in that versionNumberList contains valid result. (b) Operation failed. Output parameter versionNumberList may contain invalid result.

6.2.3 Configuration File Notifications

The following configuration file Notifications exist in the Itf-N.

- notifySessionStateChanged
- notifyGetSessionLogEnded

(Subscribe and Unsubscribe are also associated with the Bulk CM IRP, but these operations are part of the 32.302: "TM; Notification Management; Part 2: Notification IRP; IS " [3]).

6.2.3.1 General

Operations that IRPManager uses to manage subscription to receive notifications are specified in 3GPP TS 32.302 [3]. 3GPP TS 32.302 [3] also specifies a generic parameter information that is commonly found in notifications defined by IRPs. The commonly carried parameter-attributes are collectively called `notificationHeader` in the present document and . The parameter-attribute names and their qualifiers are listed in Table 12 .

Table 12: Notification Header

Parameter-Attributes defined in 3GPP TS 32.302 [3]	Qualifier for use in this IS	Comment
<code>managedObjectClass/ (objectClass([3])</code>	O	See [3]
<code>ManagedObjectInstance/(objectInstance [3])</code>	O	See [3]
<code>NotificationId</code>	O	See [3]
<code>EventTime</code>	M	See [3]
<code>systemDN</code>	O	See [3]
<code>NotificationType</code>	M	Indicates the type of notification. The type used for each Bulk CM Notification are specified in Tables 13 and 14

The following clauses define specific notifications relevant for Bulk CM IRP by extending notify in 32.302 Notification IRP IS [3].

6.2.3.2 notifySessionStateChanged (M)

The IRPAgent notifies the IRPManager that a state change has occurred on a bulk data configuration 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 7 for a further description of states.

Table 13: notifySessionStateChange parameters

Name	Qualifier	Description
<code>notificationHeader</code>	Input, M	See Table 12 Notification Header.
<code>NotificationType of notificationhHeader</code>	Input, M	See Table 12 Notification Header. For this notification it indicates notification type is Notify Session State Changed.
<code>sessionId</code>	Input, M	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.
<code>sourceIndicator</code>	Input, O	This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: <ul style="list-style-type: none"> • 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.
<code>sessionState</code>	Input, M	Indicates the state transition that caused the Notification. See clause 7. i.e. Upload Failed, Upload Completed, Download Failed, Download Completed, Activation Failed, Activation Partly Realised, Activation Completed, Fallback Failed, Fallback Partly Realised, Fallback Completed. (Note: as per sub-clause 7.2 "in-progress " transition states are not notified)

6.2.3.3 NotifyGetSessionLogEnded (M)

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.

Table 14 : notifyGetSessionLogEnded parameters

Name	Qualifier	Description
NotificationHeader	Input, M	See Table 12: Notification Header.
NotificationType of notificationHeader	Input, M	See Table 12 Notification Header. For this notification it indicates notification type is Notify Bulk CM Log State.
SessionId	Input, M	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	Input, O	This parameter, when present, indicates the source of the operation that led to the generation of this notification. It can have one of the following values: <ul style="list-style-type: none"> 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	Input, M	Indicates event that caused the Notification i.e. GetSessionLog completed successfully, GetSessionLog completed unsuccessfully.

6.3 Network Resource Model (NRM)

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

32.622: "3G Configuration Management: Generic Network Resources IRP: NRM " [4],

32.642: "3G Configuration Management: UTRAN Network Resources IRP: NRM " [5],

32.652: "3G Configuration Management: GERAN Network Resources IRP: NRM " [6].

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

7 State Machine

7.1 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 file (clause 8) 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 can not 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. Enabling the fallback option will maintain the disposition of the configuration before the activation. The fallback configuration information is established at point before the first activation is started. If there are multiple activation 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.

7.2 State Machine Description

The IRPAgent progresses Bulk CM operations and associated configuration data changes (clause 8) 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, ACTIVATION_IN_PROGRESS) are not notified to the IRPManager as they are not required.

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

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

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

The session state notification events are 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 are named unique. This kind of notion is not to be interpreted as providing implementation directions. Within the description of the substate machines it is becoming clear that they have the following state symmetries.

the state of the UPLOAD_PHASE and the DOWNLOAD_PHASE are the same

the state of the ACTIVATION_PHASE and the FALLBACK_PHASE are the same

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.

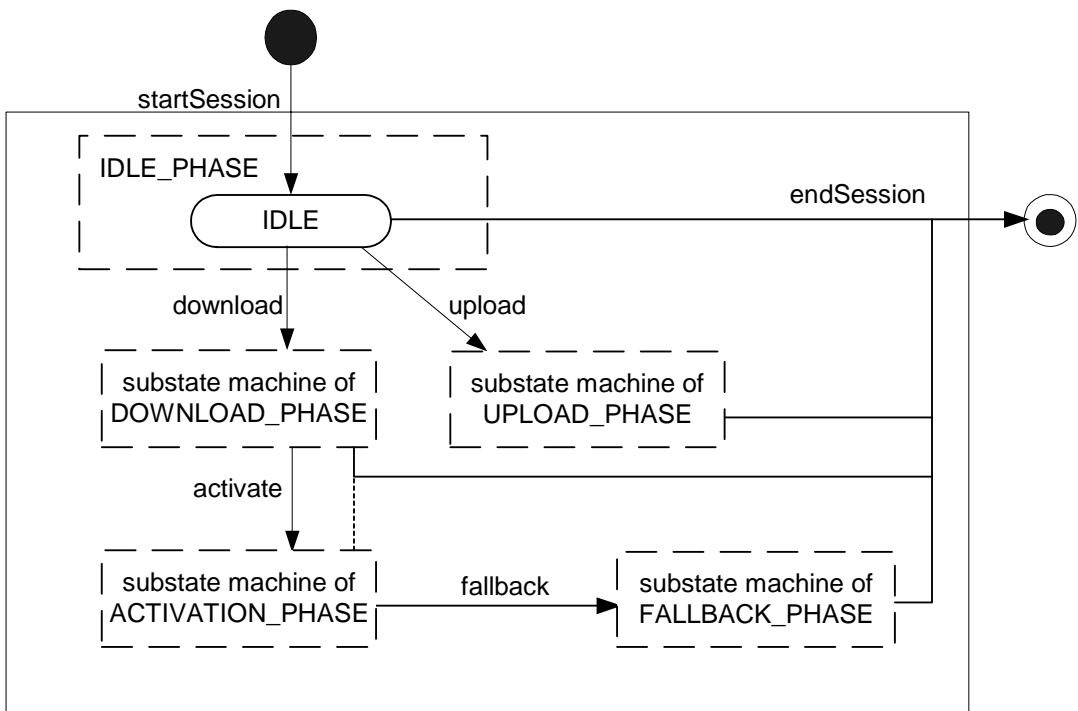


Figure 5: State Machine

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

In these figures solid transition lines indicate the transition is caused by an external event and dashed transition lines indicate the transition is caused by an internal event or decision as depicted in figure 6.

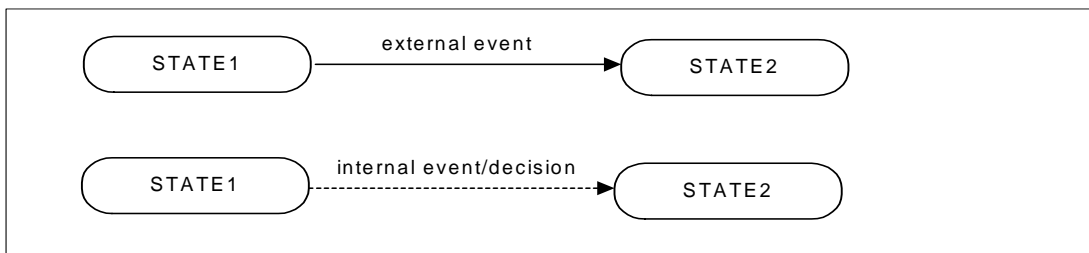


Figure 6: Depicting State Transition Lines for Internal and External Events and Decision

7.2.1 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 and activate phases can be triggered for the session.

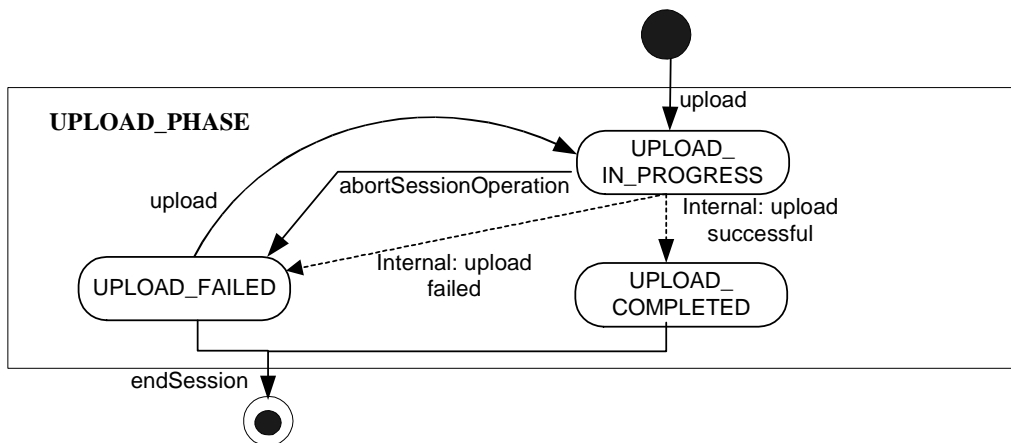


Figure 7: Substate Machine – UPLOAD_PHASE

7.2.2 Download Phase

When the download is triggered the IRP Agent copies the configuration data file (clause 0) 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 configuration is specialised to download/activation behaviour then an upload phase can not be triggered within this session.

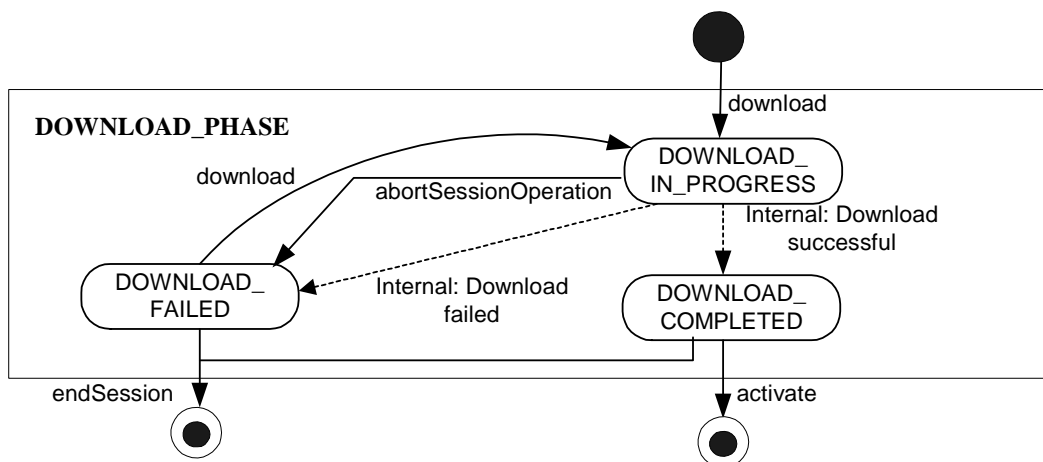


Figure 8: Substate Machine – DOWNLOAD_PHASE

7.2.3 Activation Phase

After a download had been completed the configuration can be activated into the real subnetwork of an IRP Agent. If the process fully succeeds the activation is completed.

For activation a best effort strategy shall be employed.

If the IRP Agent is unable to successfully complete all MIB changes and corresponding changes in the network elements that were actioned in the configuration data file (clause 8) the state ACTIVATION_PARTLY_REALISED is indicated. This state is not an error condition because the activation of configuration data changes follows a best effort strategy. If the activate fails completely i.e. there are no MIB changes or corresponding changes in the network elements, the state ACTIVATION_FAILED is indicated. A retry of the activate can be performed in states ACTIVATION_PARTLY_REALISED and ACTIVATION_FAILED. The ACTIVATION_FAILED state cannot be

entered if previously during the session the state had become ACTIVATION_PARTLY_REALISED. The ACTIVATION_PARTLY_REALISED state should be re-entered instead. A retry of the activate is allowed so that it is possible to recover after transient condition that caused an activate to fail or partly realise are no longer present.

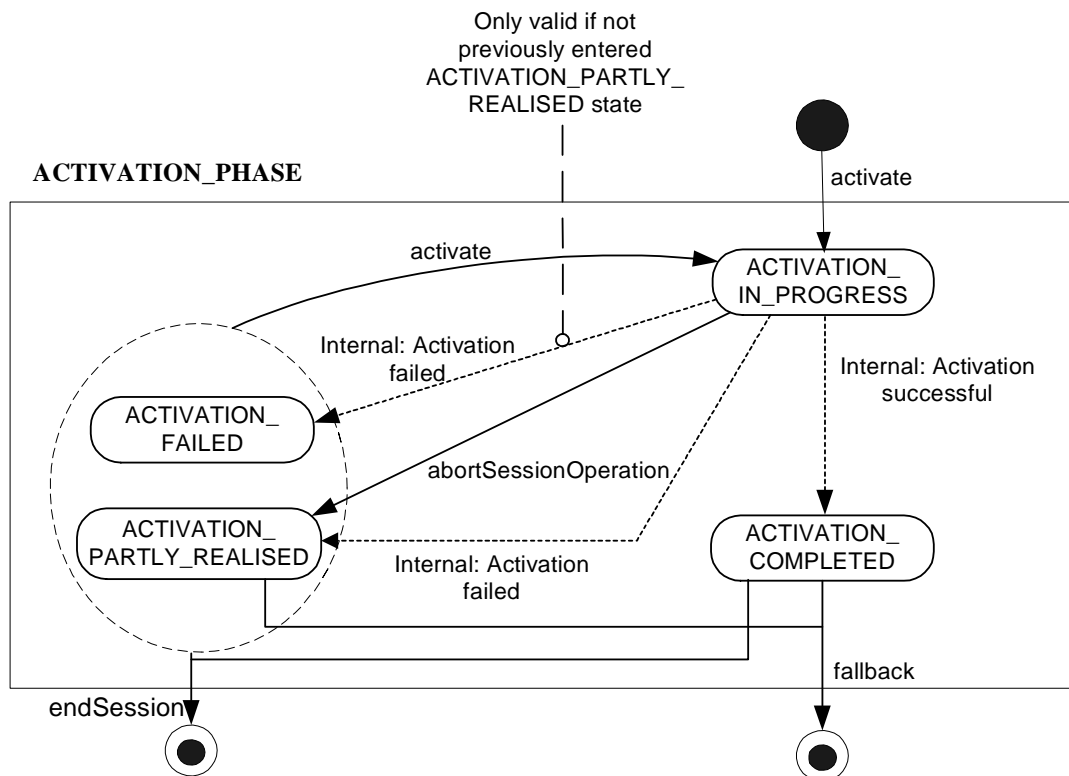


Figure 9: Substate Machine – ACTIVATION_PHASE

7.2.4 Fallback Phase

If an activate 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 activation of a session.

For fallback a best effort strategy shall be employed.

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

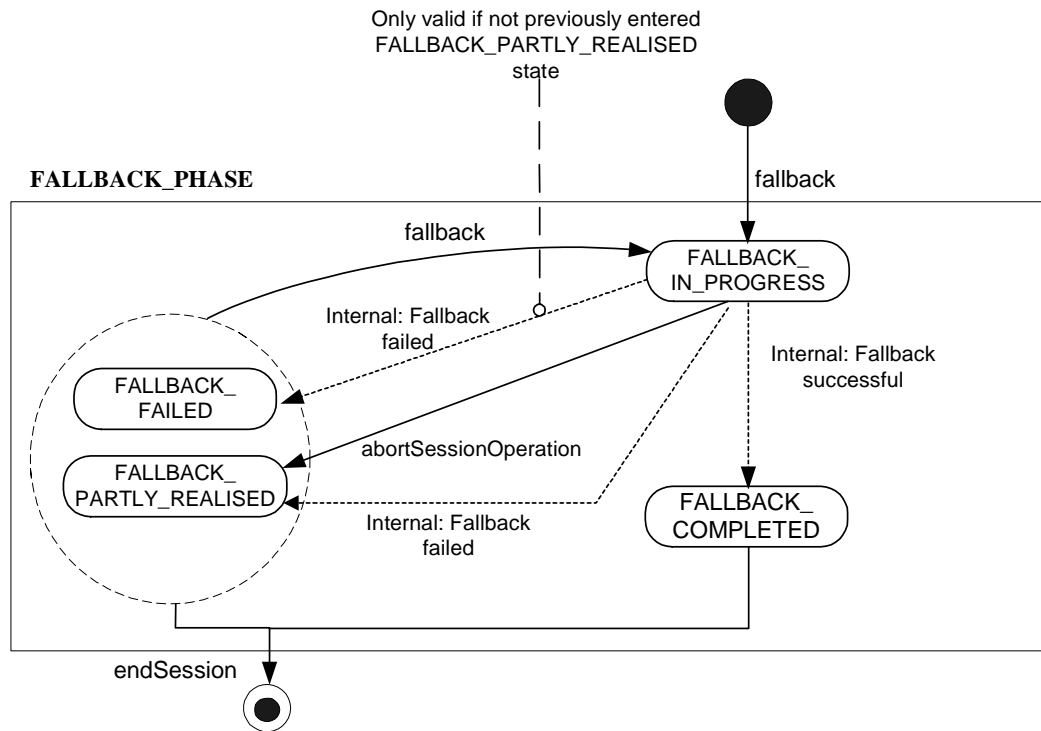


Figure 10: Substate Machine – FALLBACK_PHASE

7.3 State Machine Pre and Post Conditions Tables

For each operation Table 15 identifies the state machine pre and post conditions.

Table 15: State Machine Pre and Post Conditions

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
activate	State = DOWNLOAD_COMPLETED or ACTIVATION_PARTLY_REALISED or ACTIVATION_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 = 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 ACTIVATION_IN_PROGRESS or FALLBACK_IN_PROGRESS	State = UPLOAD_FAILED or DOWNLOAD_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

8 Bulk Configuration Data File

The overall management of Bulk CM is controlled by the operations in subclause 6.2.2. 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. (sub-clause 8.1.1)
- Delete MO. (sub-clause 8.1.2)
- Change one or more existing MO attribute values. (sub-clause 8.1.3)

The rules for ordering management actions in the configuration data file are defined in sub-clause 8.2.

8.1 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 sub-clauses 8.1.1 – 8.1.3 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 (sub-clause 6.2.2.10) 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 this document. The NRMs identified in sub-clause 6.3 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.

8.1.1 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 16 bulkCmCreateMo parameters

Name	Qualifier	Description
objectClass	Input, M	Identifies the NRM MOC within the scope of sub-clause 6.3 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 sub-clause 6.3. If the list is not empty the indicated attributes will be set to their indicated values when the object is created.

8.1.2 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 17 bulkCmDeleteMo parameters

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

8.1.3 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 18: bulkCmChangeMo parameters

Name	Qualifier	Description
objectClass	Input, M	Identifies the NRM MOC within the scope of sub-clause 6.3 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 sub-clause 6.3. The indicated attributes of the MOC instance will be changed/set to their indicated values.

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

8.2.1 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 IRP Agent following its sequentially step-by-step single pass operation. The IRP Manager has overall responsibility for ensuring the correct order of action is given according to the rules in this sub-clause.
2. The IRP Agent shall interpret the management actions in the configuration data file sequentially step-by-step in a single pass operation. The IRP Manager 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. This 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 this 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 IRP Manager 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 IRP Manager requires to be deleted. (The IRP Agent 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 IRP Manager 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 IRP Manager requires to be created.

8.2.2 Upload files

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

Annex A (informative): Scenarios

Draft supporting background informational only.

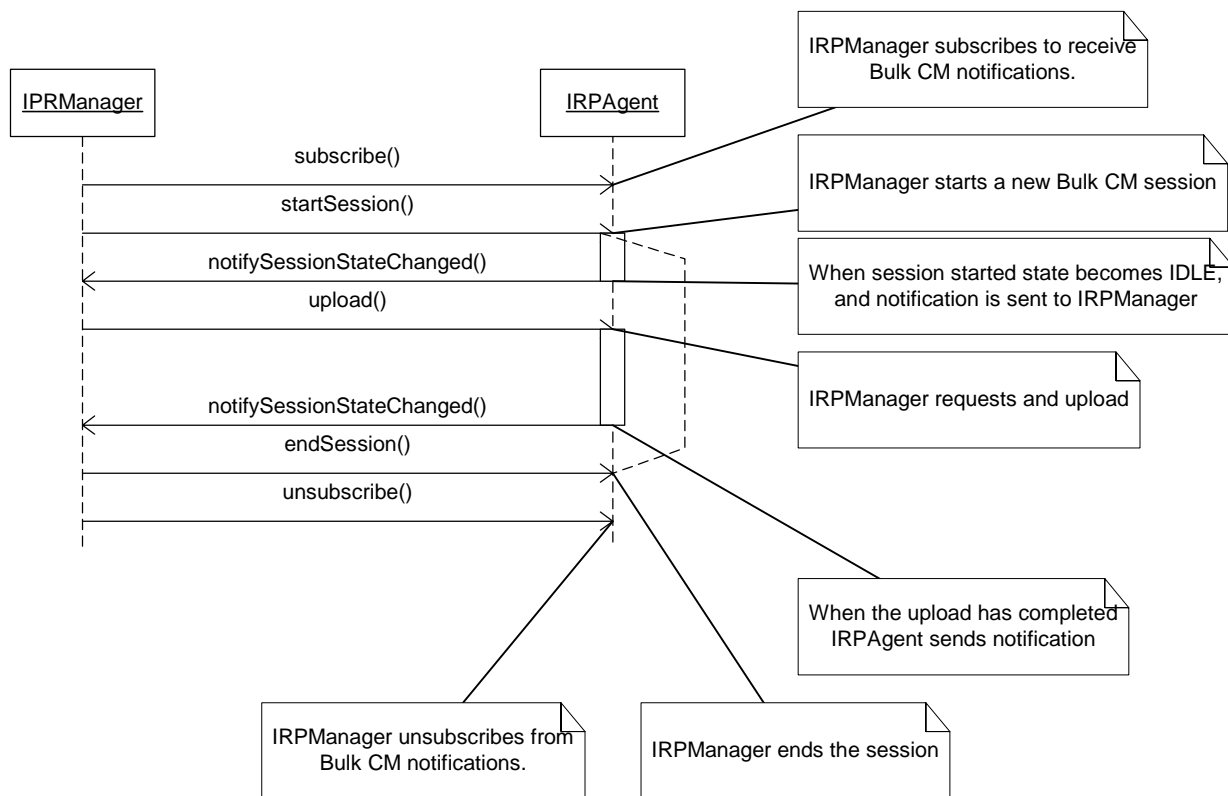


Figure A.1: Example 1: Successful Upload Session

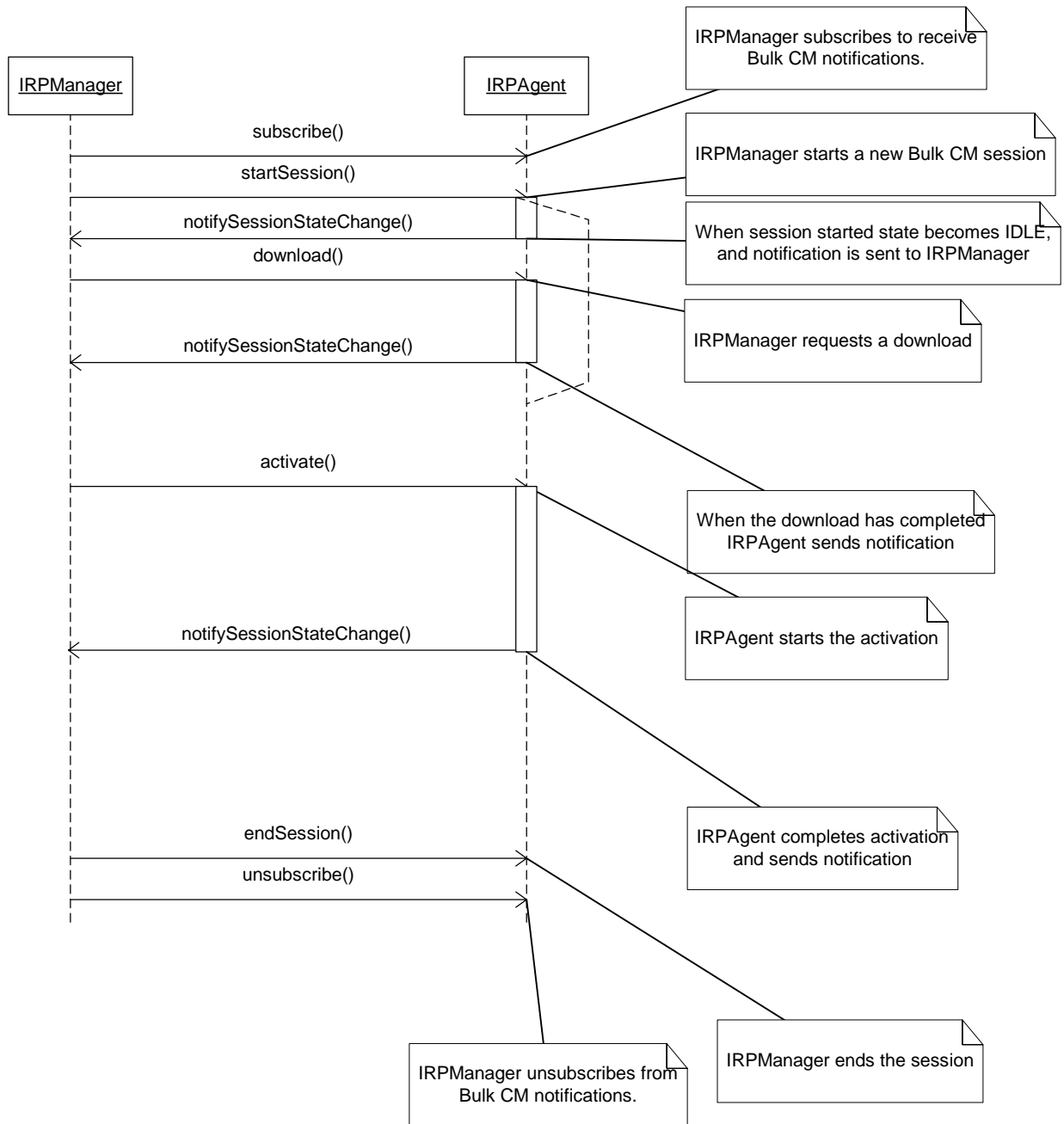


Figure A.2: Example 2: Successful Download and Activate

Annex B (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Jun 2001	S_12	SP-010283	--	--	Approved at TSG SA #12 and placed under Change Control	2.0.0	4.0.0
Sep 2001	S_13	SP-010479	001	--	Correction of State Machine Pre and Post Conditions	4.0.0	4.1.0
Jun 2002	S_16	SP-020296	002	--	Correction of behaviour for IS parameter "saveFallback" of IS operation "activate"	4.1.0	4.2.0

History

Document history		
V4.0.0	June 2001	Publication
V4.1.0	September 2001	Publication
V4.2.0	June 2002	Publication