3GPP TS 32.606 V16.0.0 (2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Telecommunication management;

Configuration Management (CM);

Basic CM Integration Reference Point (IRP);

Solution Set (SS) definitions

(Release 16)

* *

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, management, CORBA, SOAP, architecture

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

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

Internet

http://www.3gpp.org

***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.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [5](#__RefHeading___Toc335998785)

Introduction [5](#__RefHeading___Toc335998786)

1 Scope [6](#__RefHeading___Toc335998787)

2 References [6](#__RefHeading___Toc335998788)

3 Definitions and abbreviations [7](#__RefHeading___Toc335998789)

3.1 Definitions [7](#__RefHeading___Toc335998790)

3.2 Abbreviations [7](#__RefHeading___Toc335998791)

4 Solution Set Definitions [7](#__RefHeading___Toc335998792)

Annex A (normative): CORBA Solution Set [8](#__RefHeading___Toc335998793)

A.1 Architectural features [8](#__RefHeading___Toc335998794)

A.1.1 Syntax for Distinguished Names and Versions [8](#__RefHeading___Toc335998795)

A.1.2 IRP document version number string [8](#__RefHeading___Toc335998796)

A.1.3 Filter language [8](#__RefHeading___Toc335998797)

A.2 Mapping [9](#__RefHeading___Toc335998798)

A.2.1 General mappings [9](#__RefHeading___Toc335998799)

A.2.2 Operation mapping [9](#__RefHeading___Toc335998800)

A.2.3 Operation parameter mapping [10](#__RefHeading___Toc335998801)

A.3 Solution Set definitions [13](#__RefHeading___Toc335998802)

A.3.1 IDL definition structure [13](#__RefHeading___Toc335998803)

A.3.2 IDL specification "BasicCMIRPConstDefs.idl" [14](#__RefHeading___Toc335998804)

A.3.3 IDL specification "BasicCMIRPSystem.idl" [17](#__RefHeading___Toc335998805)

Annex B (normative): SOAP Solution Set [23](#__RefHeading___Toc335998806)

B.1 Architectural features [23](#__RefHeading___Toc335998807)

B.1.1 Syntax for Distinguished Names and Versions [23](#__RefHeading___Toc335998808)

B.1.2 Supported W3C specifications [23](#__RefHeading___Toc335998809)

B.1.3 Prefixes and namespaces [23](#__RefHeading___Toc335998810)

B.1.4 Filter language [23](#__RefHeading___Toc335998811)

B.2 Mapping [24](#__RefHeading___Toc335998812)

B.2.1 General mappings [24](#__RefHeading___Toc335998813)

B.2.2 Operation mapping [24](#__RefHeading___Toc335998814)

B.2.3 Operation parameter mapping [25](#__RefHeading___Toc335998815)

B.2.3.1 Operation getMoAttributes [25](#__RefHeading___Toc335998816)

B.2.3.1.1 Input parameters [25](#__RefHeading___Toc335998817)

B.2.3.1.2 Output parameters [25](#__RefHeading___Toc335998818)

B.2.3.1.3 Fault definition [26](#__RefHeading___Toc335998819)

B.2.3.2 Operation getContainment [26](#__RefHeading___Toc335998820)

B.2.3.2.1 Input parameters [26](#__RefHeading___Toc335998821)

B.2.3.2.2 Output parameters [26](#__RefHeading___Toc335998822)

B.2.3.2.3 Fault definition [27](#__RefHeading___Toc335998823)

B.2.3.3 Operation createMO [27](#__RefHeading___Toc335998824)

B.2.3.3.1 Input parameters [27](#__RefHeading___Toc335998825)

B.2.3.3.2 Output parameters [27](#__RefHeading___Toc335998826)

B.2.3.3.3 Fault definition [28](#__RefHeading___Toc335998827)

B.2.3.4 Operation deleteMO [28](#__RefHeading___Toc335998828)

B.2.3.4.1 Input parameters [28](#__RefHeading___Toc335998829)

B.2.3.4.2 Output parameters [28](#__RefHeading___Toc335998830)

B.2.3.4.3 Fault definition [29](#__RefHeading___Toc335998831)

B.2.3.5 Operation setMOAttributes [29](#__RefHeading___Toc335998832)

B.2.3.5.1 Input parameters [29](#__RefHeading___Toc335998833)

B.2.3.5.2 Output parameters [30](#__RefHeading___Toc335998834)

B.2.3.5.3 Fault definition [30](#__RefHeading___Toc335998835)

B.3 Solution Set definitions [31](#__RefHeading___Toc335998836)

B.3.1 WSDL definition structure [31](#__RefHeading___Toc335998837)

B.3.2 Graphical Representation [31](#__RefHeading___Toc335998838)

B.3.3 WSDL specification “BasicCMIRPSystem.wsdl” [32](#__RefHeading___Toc335998839)

Annex C (informative): Change history [39](#__RefHeading___Toc335998840)

# Foreword

This Technical Specification 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:

32.601: "Configuration Management (CM); Basic CM Integration Reference Point (IRP); Requirements"

32.602: "Configuration Management (CM); Basic CM Integration Reference Point (IRP); Information Service (IS)"

**32.606: "Configuration Management (CM); Basic CM Integration Reference Point (IRP): Solution Set (SS) definitions"**

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 Elements (NEs) and network resources , 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 (QoS). The CM actions are initiated either as 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 purpose of this document is to define the mapping of the Basic CM IRP: IS (see 3GPP TS 32.602 [8]) to the protocol specific details necessary for implementation of this IRP in a CORBA/IDL environment and in a SOAP/WSDL environment.

This document defines NRM independent data types and methods.

This Solution Set specification is related to 3GPP TS 32.602  [8].

# 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 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[4] 3GPP TS 32.306: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Solution Set (SS) definitions".

[5] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirement".

[6] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[7] 3GPP TS 32.600: "Telecommunication management; Configuration Management (CM); Concept and high-level requirements".

[8] 3GPP TS 32.602: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP) Information Service (IS)".

[9] 3GPP TS 32.666: "Telecommunication management; Configuration Management (CM); Kernel CM Integration Reference Point (IRP): Solution Set (SS) definitions".

[10] OMG Notification Service, Version 1.0.

[11] W3C SOAP 1.1 specification (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>).

[12] W3C WSDL 1.1 specification (<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>).

[13] W3C XPath 1.0 specification (<http://www.w3.org/TR/1999/REC-xpath-19991116>).

[14] W3C SOAP 1.2 specification (<http://www.w3.org/TR/soap12-part1/>).

[15] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".

[16] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [16], 3GPP TS 32.101 [1], 3GPP TS 32.102 [2] and 3GPP TS 32.600 [7], 3GPP TS 32.602 [8] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [16], 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.600 [7] and 3GPP TS 32.602 [8].

**IRP document version number string:** The IRP document version number (sometimes called "IRPVersion" or "SS version number") string is used to identify this specification.

NOTE1: The string is derived using a rule described in 3GPP TS 32.311 [5]. This string (or sequence of strings, if more than one version is supported) is returned in getBasicCmIRPVersion method.

**Network resource**: See definition in TS 28.622 [15].

**Network Resource Model (NRM)**: See definition in TS 28.622 [15].

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [16] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [16].

DN Distinguished Name

MO Managed Object

MOC Managed Object Class

OMG Object Management Group

SS Solution Set

WSDL Web Service Description Language

# 4 Solution Set Definitions

This specification defines the following 3GPP Basic CM IRP Solution Set Definitions:

Annex A provides the CORBA Solution Set.  
Annex B provides the SOAP Solution Set.

Annex A (normative):  
CORBA Solution Set

This annex specifies the CORBA Solution Set for the IRP whose semantics are specified in 3GPP TS 32.602 [8].

# A.1 Architectural features

The overall architectural feature of Basic Configuration Management IRP is specified in 3GPP TS 32.602 [8].   
This clause specifies features that are specific to the CORBA SS.

## A.1.1 Syntax for Distinguished Names and Versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [3].

The version of this IRP is represented as a string (see also clause 3.1).

## A.1.2 IRP document version number string

The IRP document version number (sometimes called "IRPVersion" or "SS version number") string is used to identify this specification. The string is derived using a rule described in 3GPP TS 32.312: [6].

This string (or sequence of strings, if more than one version is supported) is returned in getBasicCmIRPVersion method.

## A.1.3 Filter language

The filter language used in the SS is the Extended Trader Constraint Language (see OMG Notification Service [10]). IRPAgents may throw a FilterComplexityLimit exception when a given filter is too complex. However, for 3GPP Release 99 an "empty filter" shall be used i.e. a filter that satisfies all MOs of a scoped search (this does not affect the filter for notifications as defined in the Notification IRP – see 3GPP TS 32.306 [4]).

# A.2 Mapping

## A.2.1 General mappings

The IS parameter name managedObjectInstance is mapped into DN.

Attributes modelling associations as defined in the NRM (here also called "reference attributes") are in this SS mapped to attributes. The names of the reference attributes in the NRM are mapped to the corresponding attribute names in the MOC. When the cardinality for an association is 0..1 or 1..1 the datatype for the reference attribute is defined as an MOReference. The value of an MO reference contains the distinguished name of the associated MO. When the cardinality for an association allows more than one referred MO, the reference attribute will be of type MOReferenceSet, which contains a sequence of MO references.

If a reference attribute is changed, an AttributeValueChange notification (see TS 32.666 [9]) is emitted.

## A.2.2 Operation mapping

The Basic CM IRP: IS (see 3GPP TS 32.602 [8]) defines semantics of operation visible across the Basic Configuration Management IRP. Table A.2.2 indicates mapping of these operations to their equivalents defined in this SS.

Table A.2.2: Mapping from IS Operation to SS equivalents

| IS Operation  (3GPP TS 32.602 [8]) | SS Method | Qualifier |
| --- | --- | --- |
| getMoAttributes | BasicCmIrpOperations::find\_managed\_objects  BasicCmInformationIterator::next\_basic\_cm\_informations | M |
| getContainment | BasicCmIrpOperations::find\_managed\_objects  BasicCmInformationIterator::next\_basic\_cm\_informations | O |
| cancelOperation | BasicCmInformationIterator::destroy | O |
| createMo | BasicCmIrpOperations::create\_managed\_object | O |
| deleteMo | BasicCmIrpOperations::delete\_managed\_objects  DeleteResultIterator::next\_basic\_cm\_informations  DeleteResultIterator::next\_delete\_errors | O |
| setMoAttributes | BasicCmIrpOperations::modify\_managed\_objects  ModifyResultIterator::next\_basic\_cm\_informations  ModifyResultIterator::next\_modification\_errors | O |

## A.2.3 Operation parameter mapping

The Basic CM IRP: IS (see 3GPP TS 32.602 [8]) defines semantics of parameters carried in operations across the Basic Configuration Management IRP. Tables A.2.3.1 through A.2.3.6 indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

The SS operation find\_managed\_objects is equivalent to the IS operation getMoAttributes when called with ResultContents set to NAMES\_AND\_ATTRIBUTES. Iterating the BasicCmInformationIterator is used to fetch the result.

Table A.2.3.1: Mapping from IS getMoAttributes parameters to SS equivalents

| IS Operation parameter | SS Method parameter | Qualifier |
| --- | --- | --- |
| invokeIdentifier | - (No equivalence) | - |
| invokeIdentifierOut | Return value of type BasicCmInformationIterator | M |
| baseObjectInstance | GenericIRPManagementConstDefs::DN base\_object | M |
| scope | SearchControl search\_control (SearchControl.type and SearchControl.level) | M |
| filter | SearchControl search\_control (SearchControl.filter) | M |
| attributeListIn | AttributeNameSet requested\_attributes | M |
| managedObjectClass managedObjectInstance attributeListOut | Return value of type BasicCmInformationIterator - parameter out ResultSet fetched\_elements of method next\_basic\_cm\_informations | M |
| status | Exceptions:  FindManagedObjects, GenericIRPManagementSystem::InvalidParameter, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit | M |

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

Table A.2.3.2: Mapping from IS getContainment parameters to SS equivalents

| IS Operation parameter | SS Method parameter | Qualifier |
| --- | --- | --- |
| invokeIdentifier | - (No equivalence) | - |
| invokeIdentifierOut | Return value of type BasicCmInformationIterator | M |
| baseObjectInstance | GenericIRPManagementConstDefs::DN base\_object | M |
| scope | SearchControl search\_control (SearchControl.type and SearchControl.level) | O |
| Not specified in IS | SearchControl search\_control (SearchControl.filter) | M |
| containment | Return value of type BasicCmInformationIterator - parameter out ResultSet fetched\_elements of method next\_basic\_cm\_informations | M |
| status | Exceptions:  FindManagedObjects, GenericIRPManagementSystem::OperationNotSupported,  GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit | M |

Table A.2.3.3: Mapping from IS cancelOperation parameters to SS equivalents

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

Table A.2.3.4: Mapping from IS createMo parameters to SS equivalents

| IS Operation parameter | SS Method parameter | Qualifier |
| --- | --- | --- |
| managedObjectClass managedObjectInstance | GenericIRPManagementConstDefs::DN object\_name | M |
| referenceObjectInstance | GenericIRPManagementConstDefs::DN reference\_object | O |
| attributeListIn attributeListOut | GenericIRPManagementConstDefs::MoAttributeSet attributes | M |
| status | AttributeErrorSeq attribute\_errors  Exceptions:  CreateManagedObject, GenericIRPManagementSystem::OperationNotSupported, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, UndefinedMOException, IllegalDNFormatException, DuplicateMO, CreateNotAllowed, ObjectClassMismatch,  NoSuchObjectClass,  ParentObjectDoesNotExist | M |

Table A.2.3.5: Mapping from IS deleteMo parameters to SS equivalents

| IS Operation parameter | SS Method parameter | Qualifier |
| --- | --- | --- |
| baseObjectInstance | GenericIRPManagementConstDefs::DN base\_object | M |
| scope | SearchControl search\_control (SearchControl.type and SearchControl.level) | M |
| filter | SearchControl search\_control (SearchControl.filter) | M |
| deletionList | Return value of type DeleteResultIterator - parameter out ResultSet fetched\_elements of method next\_basic\_cm\_informations | M |
| status | Return value of type DeleteResultIterator - parameter out DeleteErrorSeq fetched\_delete\_errors of method next\_delete\_errors  Exceptions:  DeleteManagedObjects, GenericIRPManagementSystem::OperationNotSupported, GenericIRPManagementSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit | M |

Table A.2.3.6: Mapping from IS setMoAttributes parameters to SS equivalents

| IS Operation parameter | SS Method parameter | Qualifier |
| --- | --- | --- |
| baseObjectInstance | GenericIRPManagementConstDefs::DN base\_object | M |
| scope | SearchControl search\_control (SearchControl.type and SearchControl.level) | M |
| filter | SearchControl search\_control (SearchControl.filter) | M |
| modificationList | AttributeModificationSet modifications | M |
| modificationListOut | Return value of type ModifyResultIterator - parameter out ResultSet fetched\_elements of method next\_basic\_cm\_informations | M |
| status | Return value of type ModifyResultIterator - parameter out ModifyAttributeErrorsSeq fetched\_modify\_errors of method next\_modify\_errors  Exceptions:  ModifyManagedObjects, GenericIRPManagementSystem::OperationNotSupported, GenericIRPManagementSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit | M |

# A.3 Solution Set definitions

## A.3.1 IDL definition structure

Clause A.3.2 defines the constants and types used by the Basic CM IRP.

Clause A.3.3 defines the operations which are performed by the Basic CM IRP agent.

## A.3.2 IDL specification "BasicCMIRPConstDefs.idl"

//File: BasicCMIRPConstDefs.idl

#ifndef \_BASIC\_CM\_IRP\_CONST\_DEFS\_IDL\_

#define \_BASIC\_CM\_IRP\_CONST\_DEFS\_IDL\_

#include <GenericIRPManagementConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: BasicCMIRPConstDefs

This module contains commonly used definitions for BasicCMIRP.

================================================================

\*/

module BasicCMIRPConstDefs

{

/\*\*

\* Defines the name of a Managed Object Class

\*/

typedef string MOClass;

/\*\*

\*

\* In this version the only allowed filter value is "TRUE" i.e. a filter that

\* matches everything.

\*/

typedef string Filter;

/\*\*

\* ResultContents is used to tell how much information to get back

\* from the find\_managed\_objects operation.

\*

\* NAMES: Used to get only Distinguished Name

\* for MOs.

\* The name contains both the MO class

\* and the names of all superior objects in the naming

\* tree.

\*

\* NAMES\_AND\_ATTRIBUTES: Used to get both NAMES plus

\* MO attributes (all or selected).

\*/

enum ResultContents

{

NAMES,

NAMES\_AND\_ATTRIBUTES

};

/\*\*

\* ScopeType defines the kind of scope to use in a search

\* together with SearchControl.level, in a SearchControl value.

\*

\* SearchControl.level is always >= 0. If a level is bigger than the

\* depth of the tree there will be no exceptions thrown.

\*/

enum ScopeType

{

BASE\_ONLY,

BASE\_NTH\_LEVEL,

BASE\_SUBTREE,

BASE\_ALL

};

/\*\*

\* SearchControl controls the find\_managed\_object search,

\* and contains:

\* the type of scope ("type" field),

\* the level of scope ("level" field), level 0 means the "baseObject",

\* level 1 means baseobject including its sub-ordinates etc..

\* the filter ("filter" field),

\* the result type ("contents" field).

\* The type, level and contents fields are all mandatory.

\* The filter field contains the filter expression.

\* The string "TRUE" indicates "no filter",

\* i.e. a filter that matches everything.

\*/

struct SearchControl

{

ScopeType type;

unsigned long level;

Filter filter\_;

ResultContents contents;

};

struct Result

{

GenericIRPManagementConstDefs::DN mo;

GenericIRPManagementConstDefs::MOAttributeSet attributes;

};

typedef sequence <Result> ResultSet;

/\*\*

\* AttributeErrorCategory defines the categories of errors, related to

\* attributes, that can occur during creation or modification of MOs.

\*

\* NO\_SUCH\_ATTRIBUTE: The specified attribute does not exist.

\* INVALID\_ATTRIBUTE\_VALUE: The specified attribute value is not valid.

\* MISSING\_ATTRIBUTE\_VALUE: An attribute value is required but none was

\* provided and no default value is defined for the attribute.

\* INVALID\_MODIFY\_OPERATOR: The specified modify operator is not valid

\* (e.g. operator ADD\_VALUES applied to a non multi-valued attribute

\* or operator SET\_TO\_DEFAULT applied where no default value is defined).

\* MODIFY\_NOT\_ALLOWED: The modification of the attribute is not allowed.

\* MODIFY\_FAILED: The modification failed because of an unspecified reason.

\*/

enum AttributeErrorCategory

{

NO\_SUCH\_ATTRIBUTE,

INVALID\_ATTRIBUTE\_VALUE,

MISSING\_ATTRIBUTE\_VALUE,

INVALID\_MODIFY\_OPERATOR,

MODIFY\_NOT\_ALLOWED,

MODIFY\_FAILED

};

/\*\*

\* DeleteErrorCategory defines the categories of errors that can occur

\* during deletion of MOs.

\*

\* SUBORDINATE\_OBJECT: The MO cannot be deleted due to subordinate MOs.

\* DELETE\_NOT\_ALLOWED: The deletion of the MO is not allowed.

\* DELETE\_FAILED: The deletion failed because of an unspecified reason.

\*/

enum DeleteErrorCategory

{

SUBORDINATE\_OBJECT,

DELETE\_NOT\_ALLOWED,

DELETE\_FAILED

};

/\*\*

\* AttributeError represents an error, related to an attribute, that occured

\* during creation or modification of MOs.

\* It contains:

\* - the name of the indicted attribute ("name" field),

\* - the category of the error ("error" field),

\* - optionally, the indicted attribute value ("value" field),

\* - optionally, additional details on the error ("reason" field).

\*/

struct AttributeError

{

GenericIRPManagementConstDefs::MOAttributeName name;

AttributeErrorCategory error;

GenericIRPManagementConstDefs::MOAttributeValue value;

string reason;

};

typedef sequence <AttributeError> AttributeErrorSeq;

/\*\*

\* DeleteError represents an error that occured during deletion of MOs.

\* It contains:

\* - the distinguished name of the indicted MO ("object\_name" field),

\* - the category of the error ("error" field),

\* - optionally, additional details on the error ("reason" field).

\*/

struct DeleteError

{

GenericIRPManagementConstDefs::DN object\_name;

DeleteErrorCategory error;

string reason;

};

typedef sequence <DeleteError> DeleteErrorSeq;

/\*\*

\* ModifyAttributeErrors represents errors that occured during

\* modification of attributes of a MO.

\* It contains:

\* - the distinguished name of the indicted MO ("object\_name" field),

\* - a sequence containing the attribute errors ("errors" field).

\*/

struct ModifyAttributeErrors

{

GenericIRPManagementConstDefs::DN object\_name;

AttributeErrorSeq errors;

};

typedef sequence <ModifyAttributeErrors> ModifyAttributeErrorsSeq;

typedef sequence < GenericIRPManagementConstDefs::MOAttributeName> AttributeNameSet;

/\*\*

\* ModifyOperator defines the way in which an attribute value is to be

\* applied to an attribute in a modification of MO attributes.

\*

\* REPLACE: replace the current value with the provided value

\* ADD\_VALUES: for a multi-valued attribute, add the provided values to the

\* current list of values

\* REMOVE\_VALUES: for a multi-valued attribute, remove the provided values

\* from the current list of values

\* SET\_TO\_DEFAULT: set the attribute to its default value

\*/

enum ModifyOperator

{

REPLACE,

ADD\_VALUES,

REMOVE\_VALUES,

SET\_TO\_DEFAULT

};

/\*\*

\* AttributeModification defines an attribute value and the way it is to

\* be applied to an attribute in a modification of MO attributes.

\* It contains:

\* - the name of the attribute to modify ("name" field),

\* - the value to apply to this attribute ("value" field),

\* - the way the attribute value is to be applied to the attribute

\* ("operator" field).

\*/

struct AttributeModification

{

GenericIRPManagementConstDefs::MOAttributeName name;

GenericIRPManagementConstDefs::MOAttributeValue value;

ModifyOperator operator;

};

typedef sequence <AttributeModification> AttributeModificationSet;

};

#endif // \_BASIC\_CM\_IRP\_CONST\_DEFS\_IDL\_

## A.3.3 IDL specification "BasicCMIRPSystem.idl"

//File: BasicCMIRPSystem.idl

#ifndef \_BASIC\_CM\_IRP\_SYSTEM\_IDL\_

#define \_BASIC\_CM\_IRP\_SYSTEM\_IDL\_

#include <GenericIRPManagementSystem.idl>

#include <GenericIRPManagementConstDefs.idl>

#include <BasicCMIRPConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

module BasicCmIRPSystem

{

exception IllegalFilterFormatException {

string reason;

};

exception IllegalDNFormatException {

string reason;

};

exception IllegalScopeTypeException {

string reason;

};

exception IllegalScopeLevelException {

string reason;

};

exception UndefinedMOException {

string reason;

};

exception UndefinedScopeException {

string reason;

};

exception FilterComplexityLimit {

string reason;

};

exception DuplicateMO {};

exception CreateNotAllowed {};

exception ObjectClassMismatch {};

exception NoSuchObjectClass {

BasicCMIRPConstDefs::MOClass objectClass;

};

exception ParentObjectDoesNotExist {};

/\*\*

\* System otherwise fails to complete the operation. System can provide

\* reason to qualify the exception. The semantics carried in reason

\* is outside the scope of this IRP.

\*/

exception NextBasicCmInformations { string reason; };

exception NextDeleteErrors { string reason; };

exception NextModifyErrors { string reason; };

exception DestroyException { string reason; };

exception GetBasicCmIRPVersion { string reason; };

exception GetBasicCmIRPOperationProfile { string reason; };

exception GetBasicCmIRPNotificationProfile { string reason; };

exception FindManagedObjects { string reason; };

exception CreateManagedObject { string reason; };

exception DeleteManagedObjects { string reason; };

exception ModifyManagedObjects { string reason; };

/\*\*

The BasicCmInformationIterator is used to iterate through a snapshot of

Managed Object Information when IRPManager invokes find\_managed\_objects.

IRPManager uses it to pace the return of Managed Object Information.

IRPAgent controls the life-cycle of the iterator. However, a destroy

operation is provided to handle the case where IRPManager wants to stop

the iteration procedure before reaching the last iteration.

\*/

interface BasicCmInformationIterator

{

/\*\*

This method returns between 1 and "how\_many" Managed Object information.

The IRPAgent may return less than "how\_many" items even if there are

more items to return. "how\_many" must be non-zero. Return TRUE if there

may be more Managed Object information to return. Return FALSE if there

are no more Managed Object information to be returned.

If FALSE is returned, the IRPAgent will automatically destroy the

iterator.

@parm how\_many how many elements to return in the "fetched\_elements" out

parameter.

@parm fetched\_elements the elements.

@returns A boolean indicating if any elements are returned.

"fetched\_elements" is empty when the BasicCmInformationIterator is

empty.

\*/

boolean next\_basic\_cm\_informations (

in unsigned short how\_many,

out BasicCMIRPConstDefs::ResultSet fetched\_elements

)

raises (

NextBasicCmInformations,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::OperationNotSupported);

/\*\*

This method destroys the iterator.

\*/

void destroy ()

raises (

DestroyException,

GenericIRPManagementSystem::OperationNotSupported);

}; // end of BasicCmInformationIterator

/\*\*

The DeleteResultIterator is used to iterate through the list of deleted MOs

when IRPManager invokes method "delete\_managed\_objects".

IRPManager uses it to pace the return of Managed Object Information.

IRPAgent controls the life-cycle of the iterator. However, a destroy

operation is provided to handle the case where IRPManager wants to stop

the iteration procedure before reaching the last iteration.

\*/

interface DeleteResultIterator : BasicCmInformationIterator

{

/\*\*

Inherited method "next\_basic\_cm\_informations" has the same behaviour as

for interface BasicCmInformationIterator, except that:

- The Managed Object information returned in parameter

"fetched\_elements" contains only the DNs of the deleted MOs

(no attributes are returned).

- If FALSE is returned, the IRPAgent will not automatically destroy the

iterator.

\*/

/\*\*

This method returns between 0 and "how\_many" deletion errors. The

IRPAgent may return less than "how\_many" items even if there are more

items to return. "how\_many" must be non-zero. Return TRUE if there are

more deletion errors to return. Return FALSE if there are no more

deletion errors to be returned.

If FALSE is returned and last call to inherited method

"next\_basic\_cm\_informations" also returned FALSE (i.e. no more Managed

Object information to be returned), the IRPAgent will automatically

destroy the iterator.

@parm how\_many: how many deletion errors to return in the

"fetched\_delete\_errors" out parameter.

@parm fetched\_delete\_errors: the deletion errors.

@returns: a boolean indicating if any deletion errors are returned.

\*/

boolean next\_delete\_errors (

in unsigned short how\_many,

out BasicCMIRPConstDefs::DeleteErrorSeq fetched\_delete\_errors

)

raises (

NextDeleteErrors,

GenericIRPManagementSystem::InvalidParameter);

}; // end of DeleteResultIterator

/\*\*

The ModifyResultIterator is used to iterate through the list of modified

MOs when IRPManager invokes method "modify\_managed\_objects".

IRPManager uses it to pace the return of Managed Object Information.

IRPAgent controls the life-cycle of the iterator. However, a destroy

operation is provided to handle the case where IRPManager wants to stop

the iteration procedure before reaching the last iteration.

\*/

interface ModifyResultIterator : BasicCmInformationIterator

{

/\*\*

Inherited method "next\_basic\_cm\_informations" has the same behaviour as

for interface BasicCmInformationIterator, except that:

- The Managed Object information returned in parameter

"fetched\_elements" contains DNs and attributes of the modified MOs.

- If FALSE is returned, the IRPAgent will not automatically destroy the

iterator.

\*/

/\*\*

This method returns between 0 and "how\_many" modification errors. The

IRPAgent may return less than "how\_many" items even if there are more

items to return. "how\_many" must be non-zero. Return TRUE if there are

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

modification errors to be returned.

If FALSE is returned and last call to inherited method

"next\_basic\_cm\_informations" also returned FALSE (i.e. no more Managed

Object information to be returned), the IRPAgent will automatically

destroy the iterator.

@parm how\_many: how many modification errors to return in the

"fetched\_modify\_errors" out parameter.

@parm fetched\_modify\_errors: the modification errors.

@returns: a boolean indicating if any modification errors are returned.

\*/

boolean next\_modification\_errors (

in unsigned short how\_many,

out BasicCMIRPConstDefs::ModifyAttributeErrorsSeq

fetched\_modify\_errors

)

raises (

NextModifyErrors,

GenericIRPManagementSystem::InvalidParameter);

}; // end of ModifyResultIterator

/\*\*

\* The BasicCmIrpOperations interface.

\* Supports a number of Resource Model versions.

\*/

interface BasicCmIrpOperations : GenericIRPManagementSystem::

GenericIRPManagement

{

/\*\*

\* Performs a containment search, using a SearchControl to

\* control the search and the returned results.

\*

\* All MOs in the scope constitute a set that the filter works on.

\* The result BasicCmInformationIterator contains all matched MOs,

\* with the amount of detail specified in the SearchControl.

\* For the special case when no managed objects are matched in

\* find\_managed\_objects, the BasicCmInformationIterator will be returned.

\* Executing the next\_basicCmInformations in the

\* BasicCmInformationIterator will return FALSE for

\* completion.

\*

\* @parm base\_object The start MO in the containment tree.

\* @parm search\_control the SearchControl to use.

\* @parm requested\_attributes defines which attributes to get.

\* If this parameter is empty (""), all attributes shall

\* be returned. In this version this is the only supported semantics.

\* Note that this argument is only

\* relevant if ResultContents in the search control is

\* specifed to NAMES\_AND\_ATTRIBUTES.

\*

\* @raises GenericIRPManagementSystem::ValueNotSupported if a valid but

\* unsupported parameter value is passed. E.g. the contents

\* field in the searchcontrol parameter contains the value NAMES and

\* the optional getContainment IS operation is not supported.

\* @raises UndefinedMOException The MO does not exist.

\* @raises IllegalDNFormatException The dn syntax string is

\* malformed.

\* @raises IllegalScopeTypeException The ScopeType in scope contains

\* an illegal value.

\* @raises IllegalScopeLevelException The scope level is negative

\* (<0).

\* @raises IllegalFilterFormatException The filter string is

\* malformed.

\* @raises FilterComplexityLimit if the filter syntax is correct,

\* but the filter is too complex to be processed by the IRPAgent.

\* @see SearchControl

\* @see BasicCmInformationIterator

\*/

BasicCmInformationIterator find\_managed\_objects(

in GenericIRPManagementConstDefs::DN base\_object,

in BasicCMIRPConstDefs::SearchControl search\_control,

in BasicCMIRPConstDefs::AttributeNameSet requested\_attributes)

raises (

FindManagedObjects,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

GenericIRPManagementSystem::ValueNotSupported,

GenericIRPManagementSystem::OperationNotSupported,

UndefinedMOException,

IllegalDNFormatException,

UndefinedScopeException,

IllegalScopeTypeException,

IllegalScopeLevelException,

IllegalFilterFormatException,

FilterComplexityLimit);

/\*\*

\* Performs the creation of a MO instance in the MIB maintained

\* by the IRPAgent.

\*

\* @parm object\_name: the distinguished name of the MO to create.

\* @parm reference\_object: the distinguished name of a reference MO.

\* @parm attributes: in input, initial attribute values for the MO to

\* create; in output, actual attribute values of the created MO.

\* @parm attribute\_errors: errors, related to attributes, that caused the

\* creation of the MO to fail.

\*

\* @raises GenericIRPManagementSystem::OperationNotSupported: The operation

\* is not supported.

\* @raises GenericIRPManagementSystem::ParameterNotSupported: An optional

\* parameter is not supported.

\* @raises GenericIRPManagementSystem::InvalidParameter: An invalid

\* parameter value has been provided.

\* @raises UndefinedMOException: The MO does not exist.

\* @raises IllegalDNFormatException: The DN syntax string is malformed.

\* @raises DuplicateMO: A MO already exist with the same DN as the one

\* to create.

\* @raises CreateNotAllowed: The creation of the MO is not allowed.

\* @raises ObjectClassMismatch: The object class of the MO to create does

\* not match with the object class of the provided reference MO.

\* @raises NoSuchObjectClass: The class of the object to create is not

\* recognized.

\* @raises ParentObjectDoesNotExist: The parent MO instance of the

\* ManagedEntity specified to be created does not exist.

\*/

void create\_managed\_object (

in GenericIRPManagementConstDefs::DN object\_name,

in GenericIRPManagementConstDefs::DN reference\_object,

inout GenericIRPManagementConstDefs::MOAttributeSet attributes,

out BasicCMIRPConstDefs::AttributeErrorSeq attribute\_errors

)

raises (

CreateManagedObject,

GenericIRPManagementSystem::OperationNotSupported,

GenericIRPManagementSystem::ParameterNotSupported,

GenericIRPManagementSystem::InvalidParameter,

UndefinedMOException,

IllegalDNFormatException,

DuplicateMO,

CreateNotAllowed,

ObjectClassMismatch,

NoSuchObjectClass,

ParentObjectDoesNotExist);

/\*\*

\* Performs the deletion of one or more MO instances from the MIB

\* maintained by the IRPAgent, using a SearchControl to control the

\* instances to be deleted.

\*

\* All MOs in the scope constitute a set that the filter works on.

\* All matched MOs will be deleted by this operation.

\* The returned DeleteResultIterator is used to retrieve the DNs of the

\* MOs deleted and the errors that may have occurred preventing deletion

\* of some MOs.

\* For the special case when no managed objects are matched in

\* delete\_managed\_objects, the DeleteResultIterator will be returned.

\* Executing the next\_basicCmInformations in the DeleteResultIterator

\* will return FALSE for completion.

\*

\* @parm base\_object: the start MO in the containment tree.

\* @parm search\_control: the SearchControl to use; field "contents" has no

\* meaning here and shall be ignored.

\* @returns: a DeleteResultIterator (see above).

\*

\* @raises GenericIRPManagementSystem::OperationNotSupported: The operation

\* is not supported.

\* @raises GenericIRPManagementSystem::InvalidParameter: An invalid

\* parameter value has been provided.

\* @raises UndefinedMOException: The MO does not exist.

\* @raises IllegalDNFormatException: The DN syntax string is malformed.

\* @raises IllegalScopeTypeException: The ScopeType in scope contains

\* an illegal value.

\* @raises IllegalScopeLevelException: The scope level is negative (<0).

\* @raises IllegalFilterFormatException: The filter string is malformed.

\* @raises FilterComplexityLimit: The filter syntax is correct,

\* but the filter is too complex to be processed by the IRPAgent.

\*/

DeleteResultIterator delete\_managed\_objects (

in GenericIRPManagementConstDefs::DN base\_object,

in BasicCMIRPConstDefs::SearchControl search\_control

)

raises (

DeleteManagedObjects,

GenericIRPManagementSystem::OperationNotSupported,

GenericIRPManagementSystem::InvalidParameter,

UndefinedMOException,

IllegalDNFormatException,

UndefinedScopeException,

IllegalScopeTypeException,

IllegalScopeLevelException,

IllegalFilterFormatException,

FilterComplexityLimit);

/\*\*

\* Performs the modification of MO attributes. One or more MOs attributes

\* may be modified according to a SearchControl.

\*

\* All MOs in the scope constitute a set that the filter works on.

\* All matched MOs will have their attributes modified by this operation.

\* The returned ModifyResultIterator is used to retrieve the DNs of the

\* modified MOs together with the values of the modified attributes, and

\* the errors that may have occurred preventing modification of some

\* attributes.

\* For the special case when no managed objects are matched in

\* modify\_managed\_objects, the ModifyResultIterator will be returned.

\* Executing the next\_basicCmInformations in the ModifyResultIterator

\* will return FALSE for completion.

\*

\* @parm base\_object: the start MO in the containment tree.

\* @parm search\_control: the SearchControl to use; field "contents" has no

\* meaning here and shall be ignored.

\* @parm modifications: the values for the attributes to modify and

\* the way those values are to be applied to the attributes.

\* @returns: a ModifyResultIterator (see above).

\*

\* @raises GenericIRPManagementSystem::OperationNotSupported: The operation

\* is not supported

\* @raises GenericIRPManagementSystem::InvalidParameter: An invalid

\* parameter value has been provided

\* @raises UndefinedMOException: The MO does not exist.

\* @raises IllegalDNFormatException: The DN syntax string is malformed.

\* @raises IllegalScopeTypeException: The ScopeType in scope contains

\* an illegal value.

\* @raises IllegalScopeLevelException: The scope level is negative (<0).

\* @raises IllegalFilterFormatException: The filter string is malformed.

\* @raises FilterComplexityLimit: The filter syntax is correct,

\* but the filter is too complex to be processed by the IRPAgent.

\*/

ModifyResultIterator modify\_managed\_objects (

in GenericIRPManagementConstDefs::DN base\_object,

in BasicCMIRPConstDefs::SearchControl search\_control,

in BasicCMIRPConstDefs::AttributeModificationSet modifications

)

raises (

ModifyManagedObjects,

GenericIRPManagementSystem::OperationNotSupported,

GenericIRPManagementSystem::InvalidParameter,

UndefinedMOException,

IllegalDNFormatException,

UndefinedScopeException,

IllegalScopeTypeException,

IllegalScopeLevelException,

IllegalFilterFormatException,

FilterComplexityLimit);

};

};

#endif // \_BASIC\_CM\_IRP\_SYSTEM\_IDL\_

Annex B (normative):  
SOAP Solution Set

This annex specifies the SOAP Solution Set for the IRP whose semantics are specified in 3GPP TS 32.602 [8].

# B.1 Architectural features

The overall architectural feature of Basic Configuration Management IRP is specified in 3GPP TS 32.602 [8].   
This clause specifies features that are specific to the SOAP SS.

## B.1.1 Syntax for Distinguished Names and Versions

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [3].

The version of this IRP is represented as a string (see also clause 3.1).

## B.1.2 Supported W3C specifications

The SOAP 1.1 specification [11] and WSDL 1.1 specification [12] are supported.

The SOAP 1.2 specification [14] is supported optionally.

This specification uses "document" style in the WSDL description.

This specification uses "literal" encoding style in the WSDL description.

## B.1.3 Prefixes and namespaces

This specification uses a number of namespace prefixes throughout that are listed in Table B.1.3.

Table B.1.3: Prefixes and Namespaces used in this specification

|  |  |
| --- | --- |
| **Prefix** | **Namespace** |
| http | http://schemas.xmlsoap.org/wsdl/http/ |
| soap | http://schemas.xmlsoap.org/wsdl/soap/ |
| SOAP-ENV | http://schemas.xmlsoap.org/soap/envelope/ |
| SOAP-ENC or soapenc | http://schemas.xmlsoap.org/soap/encoding/ |
| xs or xsd | http://www.w3.org/2001/XMLSchema |
| xsi | http://www.w3.org/2001/XMLSchema-instance |
| basicCMIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32606#BasicCMIRPSystem |
| basicCMIRPData | http://www.3gpp.org/ftp/specs/archive/32\_series/32606#BasicCMIRPData |
| genericIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32316#GenericIRPSystem |

## B.1.4 Filter language

The filter language used in the SS is the XPath Language (see W3C XPath 1.0 specification [13]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

# B.2 Mapping

## B.2.1 General mappings

The IS parameter name managedObjectInstance is mapped into DN.

Attributes modelling associations as defined in the NRM (here also called "reference attributes") are in this SS mapped to attributes. The names of the reference attributes in the NRM are mapped to the corresponding attribute names in the MOC. When the cardinality for an association is 0..1 or 1..1 the datatype for the reference attribute is defined as a MOReference. The value of an MO reference contains the distinguished name of the associated MO. When the cardinality for an association allows more than one referred MO, the reference attribute will be of type MOReferenceSet, which contains a sequence of MO references.

If a reference attribute is changed, an AttributeValueChange notification (see 3GPP TS 32.666 [9]) is emitted.

## B.2.2 Operation mapping

The Basic CM IRP: IS (see 3GPP TS 32.602 [8]) defines semantics of operation visible across the Basic Configuration Management IRP. Table B.2.2 indicates mapping of these operations to their equivalents defined in this SS.

Table B.2.2: Mapping from IS Operation to SS equivalents

| IS Operation  (3GPP TS 32.602 [8]) | SS Operation | Qualifier |
| --- | --- | --- |
| getMoAttributes | getMOAttributes | M |
| getContainment | getContainment | O |
| cancelOperation (see note 1) | N/A | N/A |
| createMO | createMO | O |
| deleteMO | deleteMO | O |
| setMOAttributes | setMOAttributes | O |
| getIRPVersion (see note 2) | getIRPVersion | M |
| getOperationProfile (see note 2) | getOperationProfile | O |
| getNotificationProfile (see note 2) | getNotificationProfile | O |
| NOTE 1: This operation is NOT mapped because it’s useful for one-request-and-multiple-responses operations, which are not used in this Solution Set.  NOTE 2: This operation is of IOC ManagedGenericIRP specified in [6]. The IOC BasicCmIRP of [8] inherits from it. | | |

## B.2.3 Operation parameter mapping

The Basic CM IRP: IS (see 3GPP TS 32.602 [8]) defines semantics of parameters carried in operations across the Basic Configuration Management IRP. The following tables show the mapping of these parameters, as per operation, to their equivalents defined in the present document.

### B.2.3.1 Operation getMoAttributes

#### B.2.3.1.1 Input parameters

Mapping from IS getMoAttributes input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| invokeIdentifierIn | string invokeIdentifierIn | M |
| baseObjectInstance,  scope,  filter,  attributeListIn | string queryXpathExp | M |

Here is the XML schema fragment of the getMOAttributes request:

<!-- getMOAttributes Request -->

<element name="getMOAttributes">

<complexType>

<sequence>

<element name="invokeIdentifierIn" type="string"/>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

#### B.2.3.1.2 Output parameters

Mapping from IS getMoAttributes output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| invokeIdentifierOut | string invokeIdentifierOut | M |
| managedObjectClass,  managedObjectInstance,  attributeListOut | basicCMIRPData:MOSequenceType moiListOut | M |
| status | basicCMIRPData:getMOAttributesFault | M |

The specific “attributeListOut” definition depends on the corresponding NRM XML definition.

Here is the XML schema fragment of the getMOAttributes response:

<!-- getMoAttributes Response -->

<element name="getMoAttributesResponse">

<complexType>

<sequence>

<element name="invokeIdentifierOut" type="string"/>

<<element name="moiListOut" type="basicCMIRPData:MOSequenceType"/>/sequence>

</complexType>

</element>

#### B.2.3.1.3 Fault definition

<!-- getMoAttributes Fault -->

<element name="getMOAttributesFault">

<complexType>

<choice>

<element name="getMOAttributesFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="operationCancelledFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

### B.2.3.2 Operation getContainment

#### B.2.3.2.1 Input parameters

Mapping from IS getContainment input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| invokeIdentifierIn | string invokeIdentifierIn | M |
| baseObjectInstance,  scope | string queryXpathExp | M |

Here is the XML schema fragment of the getContainment request:

<!-- getContainment Request -->

<element name="getContainment">

<complexType>

<sequence>

<element name="invokeIdentifierIn" type="string"/>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

#### B.2.3.2.2 Output parameters

Mapping from IS getContainment output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| invokeIdentifierOut | string invokeIdentifierOut | M |
| containment | string topContainerLoc | M |
| status | basicCMIRPData:getContainmentFault | M |

Here is the XML schema fragment of the getContainment response:

<!-- getContainment Response -->

<element name="getContainmentResponse">

<complexType>

<sequence>

<element name="invokeIdentifierOut" type="string"/>

<element name="topContainerLoc" type="string"/>

<!--each element contains only id attribute and other MO it contains -->

<any minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

</element>

#### B.2.3.2.3 Fault definition

<!-- getContainment Fault -->

<element name="getContainmentFault">

<complexType>

<choice>

<element name="getContainmentFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="operationCancelledFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

### B.2.3.3 Operation createMO

#### B.2.3.3.1 Input parameters

Mapping from IS createMO input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| managedObjectClass,  managedObjectInstance | string mOIElementLoc | M |
| referenceObjectInstance | string referenceObjectInstance | O |
| attributeListIn | basicCMIRPData:AnyMOType mO | M |

The specific “attributeListIn” definition depends on the corresponding NRM XML definition.

Here is the XML schema fragment of the createMO request:

<!-- createMO Request -->

<element name="createMO">

<complexType>

<sequence>

<element name="mOIElementLoc" type="string"/>

<element name="referenceObjectInstance" type="string" minOccurs="0"/>

<element name="mO" type="basicCMIRPData:AnyMOType"/>

</sequence>

</complexType>

</element>

#### B.2.3.3.2 Output parameters

Mapping from IS createMO output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| attributeListOut | basicCMIRPData:AnyMOType mO | M |
| status | basicCMIRPData:createMOFault | M |

The specific “attributeListOut” definition depends on the corresponding NRM XML definition.

Here is the XML schema fragment of the createMO response:

<!-- createMO Response -->

<element name="createMOResponse">

<complexType>

<sequence>

<element name="mO" type="basicCMIRPData:AnyMOType"/>

</sequence>

</complexType>

</element>

#### B.2.3.3.3 Fault definition

<!-- createMO Fault -->

<element name="createMOFault">

<complexType>

<choice>

<element name="createMOFault" type="string"/>

<element name="objectClassSpecificationMissmatchedFault" type="string"/>

<element name="InvalidObjectInstanceFault" type="string"/>

<element name="createNotAllowedFault" type="string"/>

<element name="noSuch­ObjectClassFault" type="string"/>

<element name="classInstanceConflictFault" type="string"/>

<element name="noSuchAttributeFault" type="string"/>

<element name="invalidAttributeValueFault" type="string"/>

<element name="missingAttributeValueFault" type="string"/>

<element name="parentObjectDoesNotExistFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

### B.2.3.4 Operation deleteMO

#### B.2.3.4.1 Input parameters

Mapping from IS deleteMO input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| baseObjectInstance,  scope,  filter | string queryXpathExp | M |

Here is the XML schema fragment of the deleteMO request:

<!-- deleteMO Request -->

<element name="deleteMO">

<complexType>

<sequence>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

#### B.2.3.4.2 Output parameters

Mapping from IS deleteMO output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| deletionList | basicCMIRPData:MOSequenceTypedeletionList | M |
| status | basicCMIRPData:deleteMOFault | M |

Here is the XML schema fragment of the deleteMO response:

<!-- deleteMO Response -->

<element name="deleteMOResponse">

<complexType>

<sequence>

<element name="deletionList" type="basicCMIRPData:MOSequenceType">

</element>

</sequence>

</complexType>

</element>

#### B.2.3.4.3 Fault definition

<!-- deleteMO Fault -->

<element name="deleteMOFault">

<complexType>

<choice>

<element name="deleteMOFault" type="string"/>

<element name="invalidObjectInstanceFault" type="string"/>

<element name="deleteNotAllowedFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

### B.2.3.5 Operation setMOAttributes

#### B.2.3.5.1 Input parameters

Mapping from IS setMOAttributes input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| baseObjectInstance,  scope,  filter,  modificationList | stringqueryXpathExp | M |
| modificationList | basicCMIRPData:AttributeModificationSetType modificationList | M |

Here is the XML schema fragment of the setMOAttributes request:

<!-- setMOAttributes Request -->

<element name="setMOAttributes">

<complexType>

<sequence>

<element name="queryXpathExp" type="string"/>

<element name="modificationList" type="basicCMIRPData:AttributeModificationSetType"/>

</sequence>

</complexType>

</element>

<complexType name=”AttributeModificationSetType”>

<sequence>

<element name="AttributeModification" maxOccurs="unbounded">

<complexType>

<sequence>

<any/>

<element name="operator" type=”basicCMIRPData:ModifyOperatorType”/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

<simpleType name="ModifyOperatorType">

<restriction base="string">

<enumeration value="REPLACE"/>

<enumeration value="ADDValues"/>

<enumeration value="REMOVEValues"/>

<enumeration value="SETToDefault"/>

</restriction>

</simpleType>

#### B.2.3.5.2 Output parameters

Mapping from IS setMOAttributes output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| modificationListOut | basicCMIRPData:MOSequenceTypemodificationListOut | M |
| status | basicCMIRPData:setMOAttributesFault | M |

Here is the XML schema fragment of the setMOAttributes response:

<!-- setMOAttributes Response -->

<element name="setMOAttributesResponse">

<complexType>

<sequence>

<element name="modificationListOut" type="basicCMIRPData:MOSequenceType"/>

</sequence>

</complexType>

</element>

#### B.2.3.5.3 Fault definition

<!-- setMOAttributes Fault -->

<element name="setMOAttributesFault">

<complexType>

<choice>

<element name="setMOAttributesFault" type="string"/>

<element name="modifyNotAllowedFault" type="string"/>

<element name="noSuchAttributeFault" type="string"/>

<element name="invalidAttributeValueFault" type="string"/>

<element name="missingAttributeValueFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

# B.3 Solution Set definitions

## B.3.1 WSDL definition structure

Clause B.3.2 provides a graphical representation of the Basic CM IRP service.

Clause B.3.3 defines the services which are supported the Basic CM IRP agent.

## B.3.2 Graphical Representation

The WSDL structure is presented in Figure B.3.2:

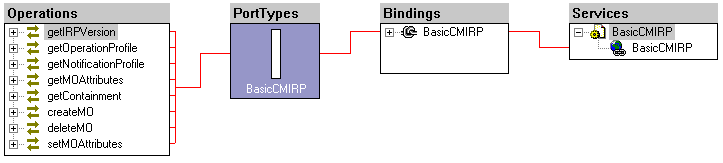


Figure B.3.2: BasicCM IRP SOAP Solution Set WSDL structure

## B.3.3 WSDL specification “BasicCMIRPSystem.wsdl”

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

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:basicCMIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#BasicCMIRPSystem"

xmlns:basicCMIRPData="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#BasicCMIRPData"

xmlns:genericIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem"

targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#BasicCMIRPSystem">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem"/>

<types>

<schema targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#BasicCMIRPData" xmlns="http://www.w3.org/2001/XMLSchema">

<!-- getMOAttributes Request -->

<element name="getMOAttributes">

<complexType>

<sequence>

<element name="invokeIdentifierIn" type="string"/>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

<!-- getMoAttributes Response -->

<element name="getMoAttributesResponse">

<complexType>

<sequence>

<element name="invokeIdentifierOut" type="string"/>

<element name="moiListOut" type="basicCMIRPData:MOSequenceType"/>

</sequence>

</complexType>

</element>

<complexType name="AnyMOType">

<sequence>

<!--MO instance location in XPath expression-->

<element name="moiLocation" type="string"/>

<!--each MO-->

<any/>

</sequence>

</complexType>

<complexType name="MOSequenceType">

<sequence>

<element name="mo" type="basicCMIRPData:AnyMOType" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<!-- getMoAttributes Fault -->

<element name="getMOAttributesFault">

<complexType>

<choice>

<element name="getMOAttributesFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="operationCancelledFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

<!-- getContainment Request -->

<element name="getContainment">

<complexType>

<sequence>

<element name="invokeIdentifierIn" type="string"/>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

<!-- getContainment Response -->

<element name="getContainmentResponse">

<complexType>

<sequence>

<!--top container element xpath location -->

<element name="invokeIdentifierOut" type="string"/>

<element name="topContainerLoc" type="string"/>

<!--each element contains only id attribute and other MO it contains -->

<any minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

</element>

<!-- getContainment Fault -->

<element name="getContainmentFault">

<complexType>

<choice>

<element name="getContainmentFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="operationCancelledFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

<!-- createMO Request -->

<element name="createMO">

<complexType>

<sequence>

<element name="mOIElementLoc" type="string"/>

<element name="referenceObjectInstance" type="string" minOccurs="0"/>

<element name="mO" type="basicCMIRPData:AnyMOType"/>

</sequence>

</complexType>

</element>

<!-- createMO Response -->

<element name="createMOResponse">

<complexType>

<sequence>

<element name="mO" type="basicCMIRPData:AnyMOType"/>

</sequence>

</complexType>

</element>

<!-- createMO Fault -->

<element name="createMOFault">

<complexType>

<choice>

<element name="createMOFault" type="string"/>

<element name="objectClassSpecificationMissmatchedFault" type="string"/>

<element name="InvalidObjectInstanceFault" type="string"/>

<element name="createNotAllowedFault" type="string"/>

<element name="noSuchObjectClassFault" type="string"/>

<element name="classInstanceConflictFault" type="string"/>

<element name="noSuchAttributeFault" type="string"/>

<element name="invalidAttributeValueFault" type="string"/>

<element name="missingAttributeValueFault" type="string"/>

<element name="parentObjectDoesNotExistFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

<!-- deleteMO Request -->

<element name="deleteMO">

<complexType>

<sequence>

<element name="queryXpathExp" type="string"/>

</sequence>

</complexType>

</element>

<!-- deleteMO Response -->

<element name="deleteMOResponse">

<complexType>

<sequence>

<element name="deletionList" type="basicCMIRPData:MOSequenceType">

</element>

</sequence>

</complexType>

</element>

<!-- deleteMO Fault -->

<element name="deleteMOFault">

<complexType>

<choice>

<element name="deleteMOFault" type="string"/>

<element name="invalidObjectInstanceFault" type="string"/>

<element name="deleteNotAllowedFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

<!-- setMOAttributes Request -->

<element name="setMOAttributes">

<complexType>

<sequence>

<element name="queryXpathExp" type="string"/>

<element name="modificationList" type="basicCMIRPData:AttributeModificationSetType"/>

</sequence>

</complexType>

</element>

<simpleType name="ModifyOperatorType">

<restriction base="string">

<enumeration value="REPLACE"/>

<enumeration value="ADDValues"/>

<enumeration value="REMOVEValues"/>

<enumeration value="SETToDefault"/>

</restriction>

</simpleType>

<complexType name="AttributeModificationSetType">

<sequence>

<element name="AttributeModification" maxOccurs="unbounded">

<complexType>

<sequence>

<any/>

<element name="operator" type="basicCMIRPData:ModifyOperatorType"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

<!-- setMOAttributes Response -->

<element name="setMOAttributesResponse">

<complexType>

<sequence>

<element name="modificationListOut" type="basicCMIRPData:MOSequenceType"/>

</sequence>

</complexType>

</element>

<!-- setMOAttributes Fault -->

<element name="setMOAttributesFault">

<complexType>

<choice>

<element name="setMOAttributesFault" type="string"/>

<element name="modifyNotAllowedFault" type="string"/>

<element name="noSuchAttributeFault" type="string"/>

<element name="invalidAttributeValueFault" type="string"/>

<element name="missingAttributeValueFault" type="string"/>

<element name="resourceLimitationFault" type="string"/>

<element name="complexityLimitationFault" type="string"/>

<element ref="basicCMIRPData:OperationNotSupportedFault"/>

<element ref="basicCMIRPData:InvalidParameterFault"/>

</choice>

</complexType>

</element>

<element name="OperationNotSupportedFault" type="string"/>

<element name="InvalidParameterFault" type="string"/>

<simpleType name="VersionNumberType">

<restriction base="string"/>

</simpleType>

<complexType name="VersionNumberSetType">

<sequence>

<element name="versionNumber" type="basicCMIRPData:VersionNumberType" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="ParameterSetType">

<sequence>

<element name="parameterName" type="string" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="OperationType">

<sequence>

<element name="operationName" type="string"/>

<element name="parameterSet" type="basicCMIRPData:ParameterSetType"/>

</sequence>

</complexType>

<complexType name="OperationSetType">

<sequence>

<element name="operation" type="basicCMIRPData:OperationType" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="NotificationType">

<sequence>

<element name="notificationName" type="string"/>

<element name="parameterSet" type="basicCMIRPData:ParameterSetType"/>

</sequence>

</complexType>

<complexType name="NotificationSetType">

<sequence>

<element name="notification" type="basicCMIRPData:NotificationType" maxOccurs="unbounded"/>

</sequence>

</complexType>

</schema>

</types>

<message name="getMOAttributesRequest">

<part name="parameter" element="basicCMIRPData:getMOAttributes"/>

</message>

<message name="getMOAttributesResponse">

<part name="parameter" element="basicCMIRPData:getMoAttributesResponse"/>

</message>

<message name="getMOAttributesFault">

<part name="parameter" element="basicCMIRPData:getMOAttributesFault"/>

</message>

<message name="getContainmentRequest">

<part name="parameter" element="basicCMIRPData:getContainment"/>

</message>

<message name="getContainmentResponse">

<part name="parameter" element="basicCMIRPData:getContainmentResponse"/>

</message>

<message name="getContainmentFault">

<part name="parameter" element="basicCMIRPData:getContainmentFault"/>

</message>

<message name="createMORequest">

<part name="parameter" element="basicCMIRPData:createMO"/>

</message>

<message name="createMOResponse">

<part name="parameter" element="basicCMIRPData:createMOResponse"/>

</message>

<message name="createMOFault">

<part name="parameter" element="basicCMIRPData:createMOFault"/>

</message>

<message name="deleteMORequest">

<part name="parameter" element="basicCMIRPData:deleteMO"/>

</message>

<message name="deleteMOResponse">

<part name="parameter" element="basicCMIRPData:deleteMOResponse"/>

</message>

<message name="deleteMOFault">

<part name="parameter" element="basicCMIRPData:deleteMOFault"/>

</message>

<message name="setMOAttributesRequest">

<part name="parameter" element="basicCMIRPData:setMOAttributes"/>

</message>

<message name="setMOAttributesResponse">

<part name="parameter" element="basicCMIRPData:setMOAttributesResponse"/>

</message>

<message name="setMOAttributesFault">

<part name="parameter" element="basicCMIRPData:setMOAttributesFault"/>

</message>

<portType name="BasicCMIRP">

<operation name="getIRPVersion">

<input message="genericIRPSystem:getIRPVersionRequest"/>

<output message="genericIRPSystem:getIRPVersionResponse"/>

<fault name="getIRPVersionFault" message="genericIRPSystem:getIRPVersionFault"/>

</operation>

<operation name="getOperationProfile">

<input message="genericIRPSystem:getOperationProfileRequest"/>

<output message="genericIRPSystem:getOperationProfileResponse"/>

<fault name="getOperationProfileFault" message="genericIRPSystem:getOperationProfileFault"/>

</operation>

<operation name="getNotificationProfile">

<input message="genericIRPSystem:getNotificationProfileRequest"/>

<output message="genericIRPSystem:getNotificationProfileResponse"/>

<fault name="getNotificationProfileFault" message="genericIRPSystem:getNotificationProfileFault"/>

</operation>

<operation name="getMOAttributes">

<input message="basicCMIRPSystem:getMOAttributesRequest"/>

<output message="basicCMIRPSystem:getMOAttributesResponse"/>

<fault name="getMOAttributesFault" message="basicCMIRPSystem:getMOAttributesFault"/>

</operation>

<operation name="getContainment">

<input message="basicCMIRPSystem:getContainmentRequest"/>

<output message="basicCMIRPSystem:getContainmentResponse"/>

<fault name="getContainmentFault" message="basicCMIRPSystem:getContainmentFault"/>

</operation>

<operation name="createMO">

<input message="basicCMIRPSystem:createMORequest"/>

<output message="basicCMIRPSystem:createMOResponse"/>

<fault name="createMOFault" message="basicCMIRPSystem:createMOFault"/>

</operation>

<operation name="deleteMO">

<input message="basicCMIRPSystem:deleteMORequest"/>

<output message="basicCMIRPSystem:deleteMOResponse"/>

<fault name="deleteMOFault" message="basicCMIRPSystem:deleteMOFault"/>

</operation>

<operation name="setMOAttributes">

<input message="basicCMIRPSystem:setMOAttributesRequest"/>

<output message="basicCMIRPSystem:setMOAttributesResponse"/>

<fault name="setMOAttributesFault" message="basicCMIRPSystem:setMOAttributesFault"/>

</operation>

</portType>

<binding name="BasicCMIRP" type="basicCMIRPSystem:BasicCMIRP">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="getIRPVersion">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#getIRPVersion"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getIRPVersionFault">

<soap:fault name="getIRPVersionFault" use="literal"/>

</fault>

</operation>

<operation name="getOperationProfile">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#getOperationProfile"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getOperationProfileFault">

<soap:fault name="getOperationProfileFault" use="literal"/>

</fault>

</operation>

<operation name="getNotificationProfile">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#getNotificationProfile"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getNotificationProfileFault">

<soap:fault name="getNotificationProfileFault" use="literal"/>

</fault>

</operation>

<operation name="getMOAttributes">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#getMOAttributes"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getMOAttributesFault">

<soap:fault name="getMOAttributesFault" use="literal"/>

</fault>

</operation>

<operation name="getContainment">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#getContainment"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getContainmentFault">

<soap:fault name="getContainmentFault" use="literal"/>

</fault>

</operation>

<operation name="createMO">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#createMO"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="createMOFault">

<soap:fault name="createMOFault" use="literal"/>

</fault>

</operation>

<operation name="deleteMO">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#deleteMO"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="deleteMOFault">

<soap:fault name="deleteMOFault" use="literal"/>

</fault>

</operation>

<operation name="setMOAttributes">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#setMOAttributes"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="setMOAttributesFault">

<soap:fault name="setMOAttributesFault" use="literal"/>

</fault>

</operation>

</binding>

<service name="BasicCMIRP">

<port name="BasicCMIRP" binding="basicCMIRPSystem:BasicCMIRP">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.606#BasicCMIRP"/>

</port>

</service>

</definitions>

Annex C (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **TSG #** | **TSG Doc.** | **CR** | **Rev** | **Subject/Comment** | **Old** | **New** |
| 05-2010 | SA-48 | SP-100273 | -- | -- | Presentation to SA for information and approval | -- | 1.0.0 |
| 06-2010 | SA-48 | -- | -- | -- | Publication | 1.0.0 | 10.0.0 |
| 09-2010 | SA-49 | SP-100487 | 001 | -- | Syntax errors in WSDL schema | 10.0.0 | 10.1.0 |
| 09-2010 | SA-49 | SP-100487 | 002 | -- | Correct the data type ‘MOAttributeName’ and inconsistency in BasicCMIRP Solution Set | 10.0.0 | 10.1.0 |
| 09-2012 | SA-57 | - | - | - | Automatic upgrade from previous Release version 10.1.0 | 10.1.0 | 11.0.0 |
| 09-2014 | SA-65 | SP-140559 | 003 | - | Update the link from Solution Set to Information Service due to the end of Release 12 | 11.0.0 | 12.0.0 |
| 2016-01 | - | - | - | - | Update to Rel-13 version (MCC) | 12.0.0 | 13.0.0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2016-06 | SA#72 | SP-160407 | 0004 | - | F | Update the link from IRP Solution Set to IRP Information Service | 13.1.0 |
| 2017-04 | SA#75 | - | - | - |  | Promotion to Release 14 without technical change | 14.0.0 |
| 2017-06 | SA#76 | SP-170502 | 0005 | - | F | Update the link from IRP Solution Set to IRP Information Service | 14.1.0 |
| 2018-06 | - | - | - | - | - | Update to Rel-15 version (MCC) | 15.0.0 |
| 2018-09 | SA#85 | SP-190752 | 0006 | - | F | Correction of NR definition to avoid misalignment with RAN2 | 15.1.0 |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | **16.0.0** |