

# ETSI TS 132 503 V9.1.0 (2010-07)

---

*Technical Specification*

**Universal Mobile Telecommunications System (UMTS);  
LTE;  
Telecommunication management;  
Self-configuration of network elements;  
Integration Reference Point (IRP);  
Common Object Request Broker Architecture (CORBA)  
Solution Set (SS)  
(3GPP TS 32.503 version 9.1.0 Release 9)**

---



---

Reference

RTS/TSGS-0532503v910

---

Keywords

LTE, UMTS

**ETSI**

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

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

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

---

**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, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**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 2010.  
All rights reserved.

**DECT**<sup>TM</sup>, **PLUGTESTS**<sup>TM</sup>, **UMTS**<sup>TM</sup>, **TIPHON**<sup>TM</sup>, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP**<sup>TM</sup> is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**LTE**<sup>TM</sup> is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM**<sup>®</sup> and the GSM logo are Trade Marks registered and owned by the GSM Association.

---

## 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 <http://webapp.etsi.org/key/queryform.asp>.

---

# Contents

Intellectual Property Rights .....	2
Foreword.....	2
Foreword.....	4
Introduction .....	4
Introduction .....	4
1 Scope .....	5
2 References .....	5
3 Definitions and abbreviations.....	5
3.1 Definitions .....	5
3.2 Abbreviations .....	6
4 Architectural Features .....	6
5 Mapping .....	6
5.1 Operation and Notification mapping .....	6
5.2 Operation parameter mapping .....	7
5.3 Notification parameter mapping.....	9
5.4 Information Object Class (IOC) mapping .....	10
5.4.1 IOC ENBLevelArcfData.....	10
5.4.1 IOC EUTranCellLevelArcfData .....	10
5.4.1 IOC AntennaLevelArcfData .....	10
<b>Annex A (normative): IDL specifications .....</b>	<b>11</b>
A.1 IDL specification (file name "SelfConfIRPConstDefs.idl") .....	11
A.2 IDL specification (file name "SelfConfIRPSystem.idl") .....	14
A.3 IDL specification (file name "SelfConfIRPNotifications.idl") .....	17
<b>Annex B (normative): CORBA IDL, NRM definitions .....</b>	<b>20</b>
<b>Annex C (informative): Change history .....</b>	<b>21</b>
History .....	22

---

## Foreword

This Technical Specification has been produced by the 3<sup>rd</sup> 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 3<sup>rd</sup> Generation Partnership Project Technical Specification Group Services and System Aspects, Telecommunication management; as identified below:

---

## Introduction

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

- 32.501: Self-Configuration of Network Elements; Concepts and Integration Reference Point (IRP) Requirements
- 32.502: Self-Configuration of Network Elements Integration Reference Point (IRP); Information Service (IS)
- 32.503: Self-Configuration of Network Elements Integration Reference Point (IRP); Common Object Request Broker Architecture (CORBA) Solution Set (SS)**

---

# 1 Scope

The present document is the "CORBA Solution Set" of Software Management IRP for the IRP whose semantics is specified in Software Management IRP Information Service (3GPP TS 32.502 [7]).

---

# 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 TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".
- [3] 3GPP TS 32.102: "Telecommunication management; Architecture".
- [4] 3GPP TS 21.905: "Vocabulary for 3GPP Specifications".
- [5] 3GPP TR 32.816: "Telecommunication management; Study on Management of Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC)".
- [6] 3GPP TS 32.501: "Telecommunication management; Self-Configuration of Network Elements; Concepts and Integration Reference Point (IRP) Requirements".
- [7] 3GPP TS 32.502: "Telecommunication management; Self-Configuration of Network Elements Integration Reference Point (IRP); Information Service (IS)".
- [8] OMG TC Document telecom/98-11-01: "OMG Notification Service".  
<http://www.omg.org/technology/documents/>
- [9] 3GPP TS 32.531: "Telecommunication management; Software management; Concepts and Integration Reference Point (IRP) Requirements".
- [10] 3GPP TS 32.532: "Telecommunication management; Software management Integration Reference Point (IRP); Information Service (IS)".
- [11] 3GPP TS 32.533: "Telecommunication management; Software management Integration Reference Point (IRP); Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

---

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 32.101 [2], TS 32.102 [3] and TR 21.905 [4] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TS 32.502 [7], TS 32.501 [6], TS 32.532 [10], TS 32.531 [9], TS 32.101 [1], TS 32.102 [2] and TS 21.905 [4], in that order.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [4], TS 32.sco [6] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TS 32.502 [7], TS 32.501 [6], TS 32.532 [10], TS 32.531 [9], TS 32.101 [1], TS 32.102 [2] and TS 21.905 [4], in that order..

---

## 4 Architectural Features

The overall architectural feature of Software Management IRP is specified in 3GPP TS 32.502 [7].

---

## 5 Mapping

### 5.1 Operation and Notification mapping

Software Management IRP: IS 3GPP TS (see 3GPP TS 32.502 [7]) defines semantics of operations and notifications visible across the Itf-N. Table 5.1.1 indicates mapping of these operations and notifications to their equivalents defined in this SS.

**Table 5.1.1: Mapping from IS Notification/Operation to SS equivalents**

IS Operation/ notification Self-Configuration IRP: IS 3GPP TS 32.502[7]	SS Method	Qualifier
listScManagementCapabilities	listScManagementCapabilities	M
listScManagementProfiles	listScManagementProfiles	M
createScManagementProfile	createScManagementProfile	M
deleteScManagementProfile	deleteScManagementProfile	M
listScProcesses	listScProcesses	M
resumeScProcess	resumeScProcess	M
resumeScProcessWithArcfData	resumeScProcessWithArcfData	M
swFallback	swFallback	M
terminateScManagementProcess	terminateScManagementProcess	M
changeScManagementProfile	changeScManagementProfile	O
notifyScManagementProfileCreation	notifyScManagementProfileCreation	M
notifyScManagementProfileDeletion	notifyScManagementProfileDeletion	M
notifyScManagementProcessCreation	notifyScManagementProcessCreation	M
notifyScManagementProcessStage	notifyScManagementProcessStage	M
notifyScManagementProcessDeletion	notifyScManagementProcessDeletion	M
notifyNewScManagementCapabilitiesAvailability	notifyNewScManagementCapabilitiesAvailability	O
notifyScManagementProfileChange	notifyScManagementProfileChange	O

## 5.2 Operation parameter mapping

Reference 3GPP TS 32.302 [6] defines semantics of parameters carried in operations across the Itf-N. The following set of tables indicate the mapping of these parameters, as per operation, to their equivalents defined in this SS.

**Table 5.2-1: Mapping from IS `listScManagementCapabilities` parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::ListScManagementCapabilities, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS `listSwMCapabilities` parameters to SS equivalents

**Table 5.2-2: Mapping from IS `listScManagementProfiles` parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::ListScManagementProfile, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS `listSwMProfiles` parameters to SS equivalents

**Table 5.2-3: Mapping from IS `createScManagementProfile` parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::CreateScManagementProfile, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS `createSwMProfile` parameters to SS equivalents

**Table 5.2-4: Mapping from IS `deleteScManagementProfile` parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::DeleteScManagementProfile, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS `deleteSwMProfile` parameters to SS equivalents

**Table 5.2-5: Mapping from IS `listScManagementProcesses` parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::ListScManagementProcesses, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS `listSwMProcesses` parameters to SS equivalents



**Table 5.2-6: Mapping from IS resumeScManagementProcess parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::ResumeScManagementProcess, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS **resumeSwMProcess** parameters to SS equivalents

**Table 5.2-7: Mapping from IS swFallback parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::SwFallback, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS **swFallback** parameters to SS equivalents

**Table 5.2-8: Mapping from IS terminateScManagementProcess parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::TerminateScManagementProcess, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS **terminateSwMProcess** parameters to SS equivalents

**Table 5.2-9: Mapping from IS changeScManagementProfile parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
result	Exceptions: SelfConfIRPConstDefs::CreateScManagementProfile, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS **changeSwMProfile** parameters to SS equivalents

**Table 5.2-10: Mapping from IS resumeScProcessWithArcfData parameters to SS equivalents**

IS Operation parameter	SS Method parameter	Qualifier
valuesOfNeededRadioParameter	SelfConfIRPConstDefs::ValuesOfNeededRadioParameterOrNull	M
fileLocation	SelfConfIRPConstDefs::FileLocationOrNull	M
validationError	SelfConfIRPConstDefs::ValidationErrorInfoOrNull	CM
result	SelfConfIRPConstDefs::ResumeScProcessWithArcfDataResult SelfConfIRPConstDefs::ResumeScManagementProcessWithArcfData, GenericIRPManagementSystem::ParameterNotSupported, GenericIRPManagementSystem::InvalidParameter, GenericIRPManagementSystem::ValueNotSupported, GenericIRPManagementSystem::OperationNotSupported	M

For all other parameters see TS 32.533 [11] mapping from IS **resumeSwMProcess** parameters to SS equivalents

## 5.3 Notification parameter mapping

Reference 3GPP TS 32.502 [7] defines semantics of parameters carried in notifications. The following tables indicate the mapping of these parameters to their SS equivalents."

The following tables indicate the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [8]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [8], is:

```

Header
  Fixed Header
    domain_name
    type_name
    event_name
  Variable Header
Body
  filterable_body_fields
  remaining_body

```

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the Software Management IRP: IS [7] defined notification parameters.

**Table 5.3.1: Mapping for notifyScManagementProfileCreation**

See TS 32.533 [11] mapping from IS **notifySwMProfileCreation**

**Table 5.3.2: Mapping for notifyScManagementProfileDeletion**

See TS 32.533 [11] mapping from IS **notifySwMProfileDeletion**

**Table 5.3.3: Mapping for notifyScProcessCreation**

See TS 32.533 [11] mapping from IS **notifySwMProcessCreation**

**Table 5.3.4: Mapping for notifyScProcessStage**

See TS 32.533 [11] mapping from IS **notifySwMProcessStage** and in addition:

IS Parameters	<SS> Parameters	Qualifier	Comment
listOfNeededRadioParameters	SelfConfIRPConstDefs::ListOfNeededRadioParameters	O	
inputForRadioParameter-Determination	SelfConfIRPConstDefs::InputForRadioParameter-Determination	O	

**Table 5.3.5: Mapping for notifyScProcessDeletion**

See TS 32.533 [11] mapping from IS **notifySwMProcessDeletion**

**Table 5.3.6: Mapping for notifyNewScManagementCapabilityAvailability**

IS Parameters	<SS> Parameters	Qualifier	Comment
id	SwMIRPConstDefs::Id	M	
nEInformation	SwMIRPConstDefs::NEInformation	M	
stepsAndOfferedStopPointList	SwMIRPConstDefs::StepsAndOfferedStopPointList	M	
offeredFinalAdministrativeStateInformation	SwMIRPConstDefs::OfferedFinalAdministrativeStateInformation	M	
swVersionToBeInstalledOfferList	SwMIRPConstDefs::SwVersionToBeInstalledOfferListConditional	C	

**Table 5.3.7: Mapping for notifyScManagementProfileChange**See TS 32.533 [11] mapping from IS `notifySwMProfileChange`

## 5.4 Information Object Class (IOC) mapping

### 5.4.1 IOC ENBLevelArcfData

Attribute of IOC ENBFunction in 3GPP TS 32.762 [2]	SS Attribute	SS Type	Support Qualifier	Read Qualifier	Write Qualifier
identifierInArcfContext	identifierInArcfContext	string	M	-	-

### 5.4.1 IOC EUtranCellLevelArcfData

Attribute of IOC ENBFunction in 3GPP TS 32.762 [2]	SS Attribute	SS Type	Support Qualifier	Read Qualifier	Write Qualifier
identifierInArcfContext	identifierInArcfContext	string	M	-	-
cellIdentity	cellIdentity	GenericNetworkResourcesIRPSystem::AttributeTypes::MOReferenceSet	M	-	M
pci	pci	GenericNetworkResourcesIRPSystem::AttributeTypes::MOReferenceSet	M	-	M
pciList	pciList	GenericNetworkResourcesIRPSystem::AttributeTypes::MOReferenceSet	CM	-	M
qRxLevMin	qRxLevMin	SelfConfIRPCConstDefs::qRxLevMin	M	-	-
threshXHigh	threshXHigh	SelfConfIRPCConstDefs::ThreshXHigh	M	-	-
threshXLow	threshXLow	SelfConfIRPCConstDefs::ThreshXLow	M	-	-
maxTxPower	maxTxPower	GenericNetworkResourcesIRPSystem::AttributeTypes::MaximumTransmissionPower	M	-	-
tac	tac	GenericNetworkResourcesIRPSystem::AttributeTypes::Tac	CM	-	-
qOffSetCell	qOffSetCell	SelfConfIRPCConstDefs::qOffSetCell	M	-	-
nrt	nrt	SelfConfIRPCConstDefs::Nrt	CM	-	-

### 5.4.1 IOC AntennaLevelArcfData

Attribute of IOC ENBFunction in 3GPP TS 32.762 [2]	SS Attribute	SS Type	Support Qualifier	Read Qualifier	Write Qualifier
identifierInArcfContext	identifierInArcfContext	string	M	-	-
antennaAzimuth	antennaAzimuth	SelfConfIRPCConstDefs::AntennaAzimuth	M	-	-
antennaTilt	antennaTilt	SelfConfIRPCConstDefs::AntennaTilt	M	-	-

## Annex A (normative): IDL specifications

### A.1 IDL specification (file name "SelfConfIRPConstDefs.idl")

```
// File: SelfConfIRPConstDefs.idl
#ifndef _SELF_CONF_IRP_CONST_DEFS_IDL_
#define _SELF_CONF_IRP_CONST_DEFS_IDL_

#include <KernelCmConstDefs.idl>
#include <NotificationIRPConstDefs.idl>
#include <SwMIRPConstDefs.idl>
#include <ManagedGenericIRPConstDefs.idl>
#include "GenericNetworkResourcesNRMDefs.idl"
#include "EUTranNetworkResourcesNRMDefs.idl"

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

/* ## Module: SelfConfIRPConstDefs */

module SelfConfIRPConstDefs
{
/*****
/* definition of types used in operations for Self-Configuration : */
*****/

    enum ResumeScProcessWithArcfDataResult { SUCCESS, FAILURE, VALIDATION_ERROR };

    struct ValueOfNeededRadioParameter
    {
        string radioParameterName;
        any radioParameterValue;
    };

    typedef sequence<ValueOfNeededRadioParameter> valuesOfNeededRadioParameter;

    /* valuesOfNeededRadioParameterOrNull is a type which can contain information or not.
    If the discriminator is true the information is present.
    Otherwise the value is null.
    */
    union valuesOfNeededRadioParameterOrNull switch (boolean)
    {
        case TRUE: valuesOfNeededRadioParameter value;
    };

    /* FileLocationOrNull is a type which can contain information or not.
    If the discriminator is true the information is present.
    Otherwise the value is null.
    */
    union FileLocationOrNull switch (boolean)
    {
        case TRUE: SwmIRPConstDefs::FileLocation value;
    };

    enum ValidationErrorReason
    {
        ParameterNotSupported,
        InvalidParameter,
        ValueNotSupported,
        MissingParameterValue,
        ConflictingParameterValue,
        SemanticsError,
        OtherError
    };
};
```

```

struct ParameterValidationError
{
    string radioParameterName;
    ValidationErrorReason validationErrorReason ;
};

typedef sequence<ParameterValidationError> ValidationErrorInfo;

/* ValidationErrorOrNull is a type which can contain information or not.
If the discriminator is true the information is present.
Otherwise the value is null.
*/
union ValidationErrorOrNull switch (boolean)
{
    case TRUE: ValidationError value;
};

typedef IdentifierInArcfContext;

typedef long CellIdentity;

typedef short Pci;

typedef EUTRANNetworkResourcesNRMDefs::PciListType PciListType

enum Q-offset { dB-24, dB-22, dB-20, dB-18, dB-16, dB-14, dB-12, dB-10, dB-8, dB-6, dB-5, dB-4, dB-
3, dB-2, dB-1, dB0, dB1, dB2, dB3, dB4, dB5, dB6, dB8, dB10, dB12, dB14, dB16, dB18, dB20, dB22,
dB24 };

typedef short Q-RxLevMin;

typedef short ReselectionThreshold;

typedef short ThreshX-High;

typedef short ThreshX-Low;

typedef short MaxTxPower;

typedef short Tac;

typedef short AntennaAzimuth;

typedef short AntennaTilt;

struct NrtEntry
{
    long tci;
    boolean isRemoveAllowed;
    boolean isHOAllowed;
};

typedef sequence<NrtEntry> Nrt;

/* all other types used in operations are imported from 32.533 */

/*****
/* definition of types in notifications for self-configuration : */
*****/

enum TriggerForDeletion { IRP_AGENT_TERMINATION, IRP_MANAGER_TERMINATION,
SELF_CONFIGURATION_SUCCESFULLY_CONCLUDED };

interface AttributeNameValue
{
    const string ID = "ID";
    const string NE_INFORMATION = "NE_INFORMATION";
    const string STEPS_AND_OFFERED_STOP_POINT_LIST = "STEPS_AND_OFFERED_STOP_POINT_LIST";
    const string OFFERED_FINAL_ADMINISTRATIVE_STATE_INFORMATION =
"OFFERED_FINAL_ADMINISTRATIVE_STATE_INFORMATION";
    const string SW_VERSION_TO_BE_INSTALLED_OFFER_LIST = "SW_VERSION_TO_BE_INSTALLED_OFFER_LIST";
    const string LIST_OF_NEEDED_RADIO_PARAMETERS = "LIST_OF_NEEDED_RADIO_PARAMETERS";
    const string INPUT_FOR_RADIO_PARAMETER_DETERMINATION = "INPUT_FOR_RADIO_PARAMETER_DETERMINATION";
    const string CELL_IDENTITY= "CELL_IDENTITY";
    const string PCI= "PCI";
};

```

```
const string PCI_LIST= "PCI_LIST";
const string Q-RX_LEV_MIN= "Q-RX_LEV_MIN";
const string THRESHX-HIGH= "THRESHX-HIGH";
const string THRESHX-LOW= "THRESHX-LOW";
const string MAX_TX_POWER= "MAX_TX_POWER";
const string TAC= "TAC";
const string Q-OFFSET_CELL= "Q-OFFSET_CELL";
const string NRT= "NRT";
const string ANTENNA_AZIMUTH= "ANTENNA_AZIMUTH";
const string ANTENNA_TILT= "ANTENNA_TILT";
};
```

```
};
```

```
#endif // _SELF_CONF_IRP_CONST_DEFS_IDL_
```

## A.2 IDL specification (file name "SelfConfIRPSystem.idl")

```

//File: SelfConfIRPSystem.idl
#ifndef _SELF_CONF_IRP_SYSTEM_IDL_
#define _SELF_CONF_IRP_SYSTEM_IDL_

#include <SelfConfIRPConstDefs.idl>
#include <SwMIRPConstDefs.idl>
#include <GenericIRPManagementSystem.idl>

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

/* ## Module: SelfConfIRPSystem */

module SelfConfIRPSystem
{
    /*
    If the system fails to complete an operation, then it can provide a reason
    to qualify the exception. The semantics carried in this reason are outside
    the scope of the present document.
    */
    exception ListScManagementCapabilities { string reason; };
    exception ListScManagementProfiles { string reason; };
    exception CreateScManagementProfile { string reason; };
    exception DeleteScManagementProfile { string reason; };
    exception ListScProcesses { string reason; };
    exception ResumeScProcess { string reason; };
    exception ResumeScProcessWithArcfData { string reason; };
    exception SwFallback { string reason; };
    exception TerminateScProcess { string reason; };
    exception ChangeScManagementProfile { string reason; };

    interface SelfConfIRPOperations_1
    {
        /* for the purpose of this operation see 3GPP TS 32.502 */
        SwMIRPConstDefs::Result listScManagementCapabilities
        (
            in SwMIRPConstDefs::NEInformationOpt nEInformation,
            out SwMIRPConstDefs::CapabilitiesList capabilitiesList
        )
        raises
        (
            ListScManagementCapabilities,
            GenericIRPManagementSystem::ParameterNotSupported,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ValueNotSupported,
            GenericIRPManagementSystem::OperationNotSupported
        );

        /* for the purpose of this operation see 3GPP TS 32.502 */
        SwMIRPConstDefs::Result listScProfiles
        (
            in SwMIRPConstDefs::NEInformationOpt nEInformation,
            out SwMIRPConstDefs::ProfileList profileList
        )
        raises
        (
            ListScProfiles,
            GenericIRPManagementSystem::ParameterNotSupported,
            GenericIRPManagementSystem::InvalidParameter,
            GenericIRPManagementSystem::ValueNotSupported,
            GenericIRPManagementSystem::OperationNotSupported
        );

        /* for the purpose of this operation see 3GPP TS 32.502 */
        SwMIRPConstDefs::Result createScManagementProfile
        (
            in SwMIRPConstDefs::IdOpt id,
            in SwMIRPConstDefs::NEInformation nEInformation,
            in SwMIRPConstDefs::SwVersionToBeInstalledOpt swVersionToBeInstalled,

```

```

    in SwMIRPConstDefs::StepsAndSelectedStopPointList stepsAndSelectedStopPointList,
    in SwMIRPConstDefs::SelectedFinalAdministrativeState selectedFinalAdministrativeState
  )
raises
  (
    CreateScManagementProfile,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
  );

/* for the purpose of this operation see 3GPP TS 32.502 */
SwMIRPConstDefs::Result deleteScManagementProfile
  (
    in SwMIRPConstDefs::Id id
  )
raises
  (
    DeleteScManagementProfile,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
  );

/* for the purpose of this operation see 3GPP TS 32.502 */
SwMIRPConstDefs::Result listScProcesses
  (
    in SwMIRPConstDefs::NEIdentificationOpt nEIdentification,
    out SwMIRPConstDefs::ProcessList processList
  )
raises
  (
    ListScProcesses,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
  );

/* for the purpose of this operation see 3GPP TS 32.502 */
SwMIRPConstDefs::Result resumeScProcess
  (
    in SwMIRPConstDefs::Id id
  )
raises
  (
    ResumeScProcess,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
  );

/* for the purpose of this operation see 3GPP TS 32.502 */
SelfConfIRPConstDefs::ResumeScProcessWithArcfDataResult resumeScProcessWithArcfData
  (
    in SwMIRPConstDefs::Id id,
    in SelfConfIRPConstDefs::ListOfNeededRadioParameters listOfNeededRadioParameters,
    in SelfConfIRPConstDefs::InputForRadioParameterDetermination
inputForRadioParameterDetermination,
    out SelfConfIRPConstDefs::ValidationErrorInfoOrNull validationError
  )
raises
  (
    ResumeScProcessWithArcfData,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
  );

/* for the purpose of this operation see 3GPP TS 32.502 */
SwMIRPConstDefs::Result swFallback

```



```
(
    in SwMIRPConstDefs::Filter filter,
    out SwMIRPConstDefs::NEList nEList
)
raises
(
    SwFallback,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
);

/* for the purpose of this operation see 3GPP TS 32.502 */
SwMIRPConstDefs::Result terminateScProcess
(
    in SwMIRPConstDefs::Id id
)
raises
(
    TerminateScProcess,
    GenericIRPManagementSystem::ParameterNotSupported,
    GenericIRPManagementSystem::InvalidParameter,
    GenericIRPManagementSystem::ValueNotSupported,
    GenericIRPManagementSystem::OperationNotSupported
);
};

interface SelfConfIRPOperations_2
{
    /* for the purpose of this operation see 3GPP TS 32.502 */
    SwMIRPConstDefs::Result changeScManagementProfile
    (
        in SwMIRPConstDefs::Id id,
        in SwMIRPConstDefs::NEInformation nEInformation,
        in SwMConfIRPConstDefs::SwVersionToBeInstalledOpt swVersionToBeInstalled,
        in SwMConfIRPConstDefs::StepsAndSelectedStopPointList stepsAndSelectedStopPointList,
        in SwMIRPConstDefs::SelectedFinalAdministrativeState selectedFinalAdministrativeState
    )
    raises
    (
        ChangeScManagementProfile,
        GenericIRPManagementSystem::ParameterNotSupported,
        GenericIRPManagementSystem::InvalidParameter,
        GenericIRPManagementSystem::ValueNotSupported,
        GenericIRPManagementSystem::OperationNotSupported
    );
};
};

#endif // _SELF_CONF_IRP_SYSTEM_IDL_
```

## A.3 IDL specification (file name "SelfConfIRPNotifications.idl")

```
//File: SelfConfIRPNotifications.idl
#ifndef _SELF_CONF_IRP_NOTIFICATIONS_IDL_
#define _SELF_CONF_IRP_NOTIFICATIONS_IDL_

#include <SelfConfIRPConstDefs.idl>
#include <SwMIRPConstDefs.idl>
#include <NotificationIRPNotifications.idl>
#include <SwMIRPNotifications.idl>

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

/* ## Module: SelfConfIRPNotifications
This contains the specification of notifications of Software Management.
=====
*/
module SelfConfIRPNotifications
{

    /* Constant definitions for the notifyScManagementProfileCreation notification */

    interface NotifyScManagementProfileCreation: SwMIRPNotifications::NotifySwMPProfileCreation
    {
        const string EVENT_TYPE = "notifyScManagementProfileCreation";
    };

    /* Constant definitions for the notifyScManagementProfileDeletion notification */

    interface NotifyScManagementProfileDeletion: SwMIRPNotifications::NotifySwMPProfileDeletion
    {
        const string EVENT_TYPE = "notifyScManagementProfileDeletion";
    };

    /* Constant definitions for the notifyScProcessCreation notification */

    interface NotifyScProcessCreation: SwMIRPNotifications::NotifySwMPProcessCreation
    {
        const string EVENT_TYPE = "notifyScProcessCreation";
    };

    /* Constant definitions for the notifyScProcessStage notification */

    interface NotifyScProcessStage: SwMIRPNotifications::NotifySwMPProcessStage
    {
        const string EVENT_TYPE = "notifyScProcessStage";

        /**
         * This constant defines the name of the listOfNeededRadioParameters property,
         * which is transported in the remaining_body.
         * The data type for the value of this property is
         * SelfConfIRPConstDefs::ListOfNeededRadioParameters.
         */
        const string ID =
            SelfConfIRPConstDefs::AttributeNameValue::LIST_OF_NEEDED_RADIO_PARAMETERS;

        /**
         * This constant defines the name of the inputForRadioParameterDetermination property,
         * which is transported in the remaining_body.
         * The data type for the value of this property is
         * SelfConfIRPConstDefs::InputForRadioParameterDetermination.
         */
        const string ID =
            SelfConfIRPConstDefs::AttributeNameValue::INPUT_FOR_RADIO_PARAMETER_DETERMINATION;
    };
};

```

```
};

/* Constant definitions for the notifyScProcessDeletion notification */
interface NotifyScProcessDeletion: SwMIRPNotifications::NotifyScProcessDeletion
{
    const string EVENT_TYPE = "notifyScProcessDeletion";
};

/* Constant definitions for the notifyNewScManagementCapabilityAvailability notification */
interface NotifyNewScManagementCapabiliyAvailability: NotificationIRPNotifications::Notify
{
    const string EVENT_TYPE = "notifyNewScManagementCapabilityAvailability";

    /**
     * This constant defines the name of the id property,
     * which is transported in the filterable_body_fields.
     * The data type for the value of this property is
     * SwMIRPConstDefs::Id.
     */
    const string ID =
        SelfConfIRPConstDefs::AttributeNameValue::ID;

    /**
     * This constant defines the name of the nEInformation property,
     * which is transported in the filterable_body_fields.
     * The data type for the value of this property is
     * SwMIRPConstDefs::NEInformation.
     */
    const string ID =
        SelfConfIRPConstDefs::AttributeNameValue::NE_INFORMATION;

    /**
     * This constant defines the name of the stepsAndOfferedStopPointList property,
     * which is transported in the remaining_body.
     * The data type for the value of this property is
     * SwMIRPConstDefs::StepsAndOfferedStopPointList.
     */
    const string ID =
        SelfConfIRPConstDefs::AttributeNameValue::STEPS_AND_OFFERED_STOP_POINT_LIST;

    /**
     * This constant defines the name of the offeredFinalAdministrativeStateInformation property,
     * which is transported in the remaining_body.
     * The data type for the value of this property is
     * SwMIRPConstDefs::OfferedFinalAdministrativeStateInformation.
     */
    const string ID =
        SelfConfIRPConstDefs::AttributeNameValue::OFFERED_FINAL_ADMINISTRATIVE_STATE_INFORMATION;

    /**
     * This constant defines the name of the swVersionToBeInstalledOfferList property,
     * which is transported in the remaining_body.
     * The data type for the value of this property is
     * SwMIRPConstDefs::SwVersionToBeInstalledOfferListOpt.
     */
    const string ID =
        SelfConfIRPConstDefs::AttributeNameValue::SW_VERSION_TO_BE_INSTALLED_OFFER_LIST;
};

/* Constant definitions for the notifyScManagementProfileChange notification */
interface NotifyScManagementProfileChange: SwMIRPNotifications::NotifySwMProfileChange
{
    const string EVENT_TYPE = "notifyScManagementProfileChange";
};

};
```

```
#endif // _SELF_CONF_IRP_NOTIFICATIONS_IDL_
```

## Annex B (normative): CORBA IDL, NRM definitions

```

//File:ArcfNRMDefs.idl
#ifndef _ArcfNRMDEFS_IDL_
#define _ArcfNRMDEFS_IDL_
#include "GenericNetworkResourcesNRMDefs.idl"
#include "EUTranNetworkResourcesNRMDefs.idl"
#pragma prefix "3gppsa5.org"
/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module ArcfNRMDefs
{

    /*
    * Definitions for MO class ENBLevelArcfData
    */
    interface ENBLevelArcfData: GenericNetworkResourcesNRMDefs::Top
    {
    const string id = "id";
    const string CLASS = "ENBLevelArcfData";
    // Attribute Names
    //
    const string identifierInArcfContext= "identifierInArcfContext";
    };

    /*
    * Definitions for MO class EUTranCellLevelArcfData
    */
    interface EUTranCellLevelArcfData: GenericNetworkResourcesNRMDefs::Top
    {
    const string CLASS = "EUTranCellLevelArcfData";
    // Attribute Names
    //
    const string id = "id";
    const string identifierInArcfContext= "identifierInArcfContext";
    const string cellIdentity = "cellIdentity";
    const string pci = "pci";
    const string pciList = "pciList";
    const string qRxLevMin = "qRxLevMin";
    const string threshXHigh = "threshXHigh";
    const string threshXLow = "threshXLow";
    const string maxTxPower = "maxTxPower";
    const string qOffSetCell = "qOffSetCell";
    const string tac = "tac";
    const string nrt = "nrt";

    };

    /*
    * Definitions for MO class AntennaLevelArcfData
    */
    interface AntennaLevelArcfData: GenericNetworkResourcesNRMDefs::Top
    {
    const string CLASS = "AntennaLevelArcfData";
    // Attribute Names
    //
    const string id = "id";
    const string identifierInArcfContext= "identifierInArcfContext";
    const string antennaAzimuth = "antennaAzimuth";
    const string antennaTilt = "antennaTilt";
    };

};

#endif // _EUTRANNETWORKRESOURCESNRMDEFS_IDL_

```

---

## Annex C (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2008-12	SP-42	SP-080715			Submitted to SA#42 for SA information and approval	1.0.0	8.0.0
2009-12	-	-	-	-	Update to Rel-9 version	8.0.0	9.0.0
2010-06	SP-48	SP-100261	001	-	Alignment of CORBA SS with TS 32.502 for ARCF	9.0.0	9.1.0

---

## History

<b>Document history</b>		
V9.0.0	February 2010	Publication
V9.1.0	July 2010	Publication