



# **DiagMon Functions Supplemental Specification**

Candidate Version 1.0 – 29 Nov 2010

---

**Open Mobile Alliance**

OMA-TS-DiagMon\_Functions-V1\_0-20101129-C

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

# Contents

<b>1. SCOPE</b> .....	<b>6</b>
<b>1.1 DEPENDENCIES</b> .....	<b>6</b>
<b>2. REFERENCES</b> .....	<b>7</b>
<b>2.1 NORMATIVE REFERENCES</b> .....	<b>7</b>
<b>2.2 INFORMATIVE REFERENCES</b> .....	<b>8</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>9</b>
<b>3.1 CONVENTIONS</b> .....	<b>9</b>
<b>3.2 DEFINITIONS</b> .....	<b>9</b>
<b>3.3 ABBREVIATIONS</b> .....	<b>9</b>
<b>4. INTRODUCTION</b> .....	<b>10</b>
<b>4.1 MOID REGISTRATION</b> .....	<b>10</b>
<b>5. DEVICE</b> .....	<b>11</b>
<b>5.1 BATTERY</b> .....	<b>11</b>
5.1.1 Battery Info.....	11
<b>5.2 MEMORY</b> .....	<b>14</b>
5.2.1 Introduction.....	14
5.2.2 Non-applicable nodes from DiagMon MO definition.....	14
5.2.3 Function Description.....	15
<b>5.3 DIAGNOSTIC</b> .....	<b>17</b>
5.3.1 Trap Event Logging Function.....	17
<b>5.4 PANIC LOGS</b> .....	<b>22</b>
5.4.1 Introduction.....	22
5.4.2 Non-applicable nodes from DiagMon MO definition.....	22
5.4.3 Function Description.....	22
<b>5.5 RESTART</b> .....	<b>24</b>
5.5.1 Introduction.....	24
5.5.2 Non-applicable nodes from DiagMon MO definition.....	24
5.5.3 Function Description.....	25
5.5.4 Additional Information .....	28
<b>6. RADIO PROTOCOL (3GPP, 3GPP2, WIFI, BLUETOOTH, WIMAX, ETC.)</b> .....	<b>29</b>
<b>6.1 RF METRICS</b> .....	<b>29</b>
6.1.1 3GPP GSM Metrics.....	29
6.1.2 3GPP UMTS Metrics.....	34
6.1.3 3GPP LTE Metrics.....	43
<b>6.2 AIR INTERFACE METRICS</b> .....	<b>49</b>
6.2.1 Mobility Management Rejection.....	49
<b>7. IP LAYER</b> .....	<b>54</b>
<b>7.1 DATA SESSION METRICS</b> .....	<b>54</b>
7.1.1 Data Call and Data Session.....	54
<b>8. APPLICATION LAYER</b> .....	<b>60</b>
<b>8.1 SMS</b> .....	<b>60</b>
8.1.1 SMS Options.....	60
8.1.2 SMS Usage .....	62
<b>8.2 MMS</b> .....	<b>64</b>
8.2.1 MMS Usage .....	64
<b>8.3 BROWSING</b> .....	<b>66</b>
8.3.1 Browsing Usage.....	66
<b>8.4 APPLICATION MONITORING</b> .....	<b>69</b>
8.4.1 Introduction.....	69
8.4.2 Non-applicable nodes from DiagMon MO definition.....	70
8.4.3 Function Description.....	70

- 8.5 NFC (NEAR FIELD COMMUNICATION) .....73
  - 8.5.1 Introduction..... 73
  - 8.5.2 Non-applicable nodes from DiagMon MO definition ..... 73
  - 8.5.3 Function Description..... 73
- 9. USER INTERFACE LAYER .....80
  - 9.1 SETTINGS .....80
    - 9.1.1 User Equipment Setting ..... 80
  - 9.2 PHONE BOOK.....83
    - 9.2.1 Introduction..... 83
    - 9.2.2 Non-applicable nodes from DiagMon MO definition ..... 83
    - 9.2.3 Function Description..... 83
- APPENDIX A. CHANGE HISTORY (INFORMATIVE).....86
  - A.1 APPROVED VERSION HISTORY .....86
  - A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY .....86
- APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....89
  - B.1 SCR FOR DIAGMON CLIENT (BATTERYINFO).....89
  - B.2 SCR FOR DIAGMON SERVER (BATTERYINFO) .....89
  - B.3 SCR FOR DIAGMON CLIENT (MEMORY).....89
  - B.4 SCR FOR DIAGMON SERVER (MEMORY) .....89
  - B.5 SCR FOR DIAGMON CLIENT (TRAPEVENTLOGGING).....89
  - B.6 SCR FOR DIAGMON SERVER (TRAPEVENTLOGGING) .....90
  - B.7 SCR FOR DIAGMON CLIENT (PANICLOG).....90
  - B.8 SCR FOR DIAGMON SERVER (PANICLOG) .....91
  - B.9 SCR FOR DIAGMON CLIENT (RESTART) .....91
  - B.10 SCR FOR DIAGMON SERVER (RESTART).....92
  - B.11 SCR FOR DIAGMON CLIENT (RFPARMS\_3GPP\_GSM).....92
  - B.12 SCR FOR DIAGMON SERVER (RFPARMS\_3GPP\_GSM) .....93
  - B.13 SCR FOR DIAGMON CLIENT (RFPARMS\_3GPP\_UMTS) .....93
  - B.14 SCR FOR DIAGMON SERVER (RFPARMS\_3GPP\_UMTS).....94
  - B.15 SCR FOR DIAGMON CLIENT (RFPARMS\_3GPP\_LTE).....94
  - B.16 SCR FOR DIAGMON SERVER (RFPARMS\_3GPP\_LTE) .....95
  - B.17 SCR FOR DIAGMON CLIENT (MOBILITY\_MANAGEMENT\_REJECTION).....95
  - B.18 SCR FOR DIAGMON SERVER (MOBILITY\_MANAGEMENT\_REJECTION) .....96
  - B.19 SCR FOR DIAGMON CLIENT (DATACALLANDDATASESSION) .....96
  - B.20 SCR FOR DIAGMON SERVER (DATACALLANDDATASESSION) .....97
  - B.21 SCR FOR DIAGMON CLIENT (SMSOPTIONS).....97
  - B.22 SCR FOR DIAGMON SERVER (SMSOPTIONS) .....97
  - B.23 SCR FOR DIAGMON CLIENT (SMSUSAGE) .....98
  - B.24 SCR FOR DIAGMON SERVER (SMSUSAGE).....98
  - B.25 SCR FOR DIAGMON CLIENT (MMSUSAGE) .....98
  - B.26 SCR FOR DIAGMON SERVER (MMSUSAGE).....99
  - B.27 SCR FOR DIAGMON CLIENT (BROWSINGUSAGE).....99
  - B.28 SCR FOR DIAGMON SERVER (BROWSINGUSAGE) .....100
  - B.29 SCR FOR DIAGMON CLIENT (APPMONLOG).....100
  - B.30 SCR FOR DIAGMON SERVER (APPMONLOG) .....100
  - B.31 SCR FOR DIAGMON CLIENT (NFC) .....101
  - B.32 SCR FOR DIAGMON SERVER (NFC).....101
  - B.33 SCR FOR DIAGMON CLIENT (UE SETTING).....102
  - B.34 SCR FOR DIAGMON SERVER (UE SETTING) .....102
  - B.35 SCR FOR DIAGMON CLIENT (PHONEBOOK) .....102
  - B.36 SCR FOR DIAGMON SERVER (PHONEBOOK).....103
- APPENDIX C. NFC XML SCHEMAS .....104
  - C.1 NFCRFSIGNAL .....104
  - C.2 NFCMODEERROR.....104
  - C.3 NFCTYPEERROR .....105

# Figures

Figure 1 - Battery Info Function..... 11

Figure 2 - Memory Function..... 15

Figure 3 - Trap Event Logging Function ..... 18

Figure 4 - Panic Logs Function..... 22

Figure 5 - Restart Function..... 25

Figure 6 - 3GPP GSM RF Metrics Function ..... 30

Figure 7 - 3GPP UMTS RF Metrics Function..... 35

Figure 8 - 3GPP LTE RF Metrics Function ..... 44

Figure 9 - Mobility Management Rejection Function ..... 50

Figure 10 - Data Call and Data Session Function ..... 54

Figure 11 - SMS Options Function..... 60

Figure 12 - SMS Usage Function ..... 62

Figure 13 - MMS Usage Function..... 64

Figure 14 - Browsing Usage Function ..... 67

Figure 15 - Application Monitoring Function ..... 70

Figure 16 - NFC Info Function ..... 73

Figure 17 - User Equipment Setting Function..... 80

Figure 18 - Phone Book Function..... 83

# Tables

No table of figures entries found.

# 1. Scope

This document defines the specific Diagnostics and Monitoring Functions using the framework as defined in [DiagMonTS].

Existing DiagMon framework features are reused and this document provides information on the standardised MO format of a set of DiagMon Functions. This specification defines:

- The standardized Management Object Identifier (MOID) format for each of the DiagMon Functions covered in this specification
- Any additional information to execute the DiagMon Function
- The location and format of the data as result of DiagMon Function execution on the device

## 1.1 Dependencies

The management objects in this TS have a dependency on OMA Device Management v1.2 [DM1\_2] or later compatible version, as well as OMA Diagnostic and Monitoring v1.0 [DiagMon1\_0].

## 2. References

### 2.1 Normative References

- [3GPP-TS\_23.003] 3GPP TS 23.003 Technical Specification Group Core Network and Terminals; Numbering, addressing and identification
- [3GPP-TS\_23.032] 3GPP TS 23.032: Technical Specification Group Core Network and Terminals; Access Network Discovery and Selection Function (ANDSF) Management Object (MO)
- [3GPP-TS\_23.038] 3GPP TS 23.038 “Alphabets and language-specific information”,  
URL:[http://www.3gpp.org/ftp/Specs/archive/23\\_series/23.038/](http://www.3gpp.org/ftp/Specs/archive/23_series/23.038/)
- [3GPP-TS\_24.008] 3GPP TS 24.008: Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols
- [3GPP-TS\_25.213] 3GPP TS 25.213: Technical Specification Group Radio Access Network; Spreading and modulation (FDD).
- [3GPP-TS\_25.215] 3GPP TS 25.215: Technical Specification Group Radio Access Network; Physical layer; Measurements (FDD)
- [3GPP-TS\_25.225] 3GPP TS 25.225: Technical Specification Group Radio Access Network; Physical layer; Measurements (TDD)
- [3GPP-TS\_25.331] 3GPP TS 25.331: Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification
- [3GPP-TS\_25.433] 3GPP TS 25.433: Technical Specification Group Radio Access Network; UTRAN Iub interface Node B Application Part (NBAP) signalling
- [3GPP-TS\_36.101] 3GPP TS 36.101: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception
- [3GPP-TS\_36.211] 3GPP TS 36.211: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation
- [3GPP-TS\_36.213] 3GPP TS 36.213: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures
- [3GPP-TS\_36.214] 3GPP TS 36.214: Technical Specification Group Radio Access Network; Physical layer; Measurements
- [3GPP-TS\_36.331] 3GPP TS 36.331: Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)
- [3GPP-TS\_44.018] 3GPP TS 44.018: Technical Specification Group GSM/EDGE Radio Access Network; Mobile Radio Interface Layer 3 Specification; Radio Resource Control (RRC) Protocol
- [DiagMon1\_0] “OMA DiagMon Management Object Enabler Release Definition”, Version 1.0, Open Mobile Alliance, OMA-ERELD-DiagMon\_V1\_0,  
URL:<http://www.openmobilealliance.org/>
- [DiagMonTS] “DiagMon Management Object”, Version 1.0, Open Mobile Alliance, OMA-TS-DiagMonMO-V1\_0,  
URL:<http://www.openmobilealliance.org/>
- [DM1\_2] “OMA Device Management Enabler Release Definition”, Version 1.2, Open Mobile Alliance, OMA-ERELD-DM\_V1\_2,  
URL:<http://www.openmobilealliance.org/>
- [DMTND] “OMA Device Management Tree and Description, Version 1.2”. Open Mobile Alliance™, OMA-TS-DM\_TND-V1\_2.  
URL:<http://www.openmobilealliance.org/>
- [ISO8601] ISO 8601:2004, Data elements and interchange formats -- Information interchange -- Representation of dates and times.  
URL:<http://www.iso.ch/>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997,

- URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005,  
URL:<http://www.ietf.org/rfc/rfc4234.txt>
- [SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures,  
URL:<http://www.openmobilealliance.org/>
- [TrapMO] “DiagMon Trap Management Object”, Version 1.0, Open Mobile Alliance, OMA-TS-DiagMonTrapMO-V1\_0,  
URL:<http://www.openmobilealliance.org/>

## 2.2 Informative References

- [DMSTDOBJ1\_3] “OMA Device Management Standardized Objects, Version 1.3”. Open Mobile Alliance™. OMA-TS-DM\_StdObj-V1\_3.  
URL:<http://www.openmobilealliance.org/>
- [NFC-WP] “Essentials for Successful NFC Mobile Ecosystems”  
URL:[http://www.nfc-forum.org/resources/white\\_papers/NFC\\_Forum\\_Mobile\\_NFC\\_Ecosystem\\_White\\_Paper.pdf](http://www.nfc-forum.org/resources/white_papers/NFC_Forum_Mobile_NFC_Ecosystem_White_Paper.pdf)
- [OMADICT] “Dictionary for OMA Specifications”, Version 2.7, Open Mobile Alliance™,  
OMA-ORG-Dictionary-V2\_7,  
URL:<http://www.openmobilealliance.org/>



## 3. Terminology and Conventions

### 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

### 3.2 Definitions

**DiagMon Functions**            An extension to the DiagMon framework that allows the Diagnostics & Monitoring System to access Diagnostics and Monitoring information from the Device, and can request the Device to execute Diagnostic and Monitoring operations.

### 3.3 Abbreviations

<b>DiagMon</b>	Diagnostics and Monitoring
<b>DM</b>	Device Management
<b>MO</b>	Management Object
<b>MOID</b>	Management Object Identifier
<b>NFC</b>	Near Field Communication
<b>OMA</b>	Open Mobile Alliance
<b>OMNA</b>	Open Mobile Naming Authority
<b>SDO</b>	Standards Developing Organization
<b>URL</b>	Uniform Resource Locator

## 4. Introduction

This section is informative.

In order to provide advanced customer care services, the DiagMon [DIAGMON] enabler provides an interface between the Management Authority and Devices – based on OMA DM.

The DiagMon Function specification defines a number of standardised instantiations of the [DIAGMON] framework for the purposes of obtaining diagnostics and monitoring information in a predictable and industry-wide manner.

### 4.1 MOID Registration

Open Mobile Naming Authority (OMNA) maintains a registry of Management Object Identifiers (MOIDs) for standardized DiagMon functions. This registry can be found at the following URL:

<http://www.openmobilealliance.org/tech/omna>

The OMNA DiagMon Registry is open to OMA Working Groups, external Standards Developing Organizations (SDOs) and vendors. The registration process binds a DiagMon Function to a globally unique MOID.

## 5. Device

### 5.1 Battery

#### 5.1.1 Battery Info

##### 5.1.1.1 Introduction

If a device contains a battery, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This continuously available function will allow the Server to determine a variety of battery information, such as the percentage of charge remaining in the battery or batteries. Since this function is always available, no operations are allowed. No configuration is allowed either.

These values are useful, for example, in determining if there is enough power to do time-intensive operations such as firmware update.

##### 5.1.1.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- Operations (the function is always available)
- DiagMonConfig (the function does not have any configurable parameters)
- ServerID (The Diagnostics and Monitoring data is either retrieved by using Get command or delivered to the alternate data server).
- Status (the function is always available)

##### 5.1.1.3 Function Description

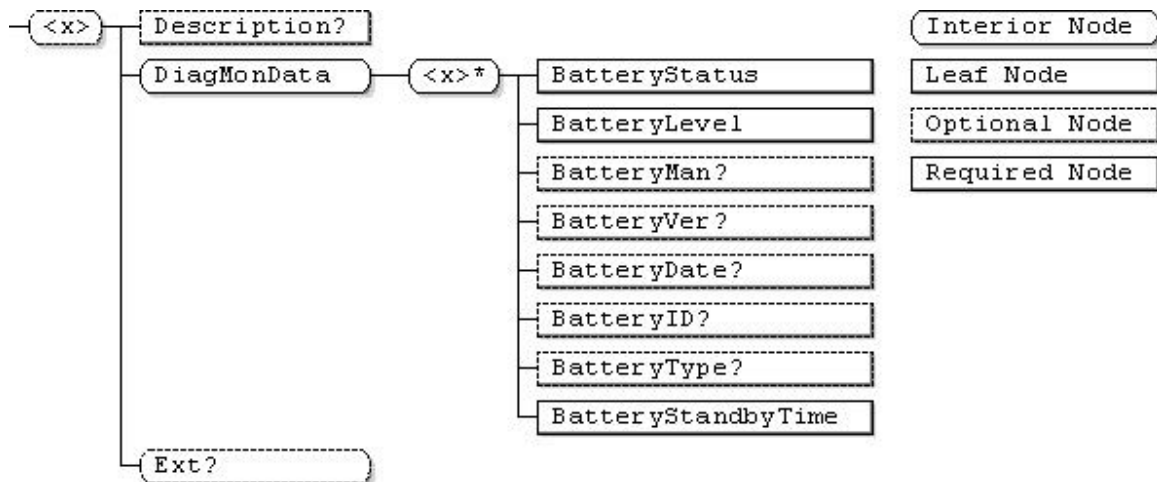


Figure 1 - Battery Info Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Battery Management Object. Identifier for the Battery MO MUST be: "urn:oma:mo:oma-diag:batteryinfo:1.0".

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData/<x>

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This node is a placeholder for zero or more instances of battery data.

<x>/DiagMonData/<x>/BatteryStatus

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node is used to indicate the status of the battery. The value of BatteryStatus MUST be one of the following values:

BatteryStatus	Meaning	Description
0	Normal	The battery is operating normally and not on power.
1	Charging	The battery is currently charging.
2	Charge Complete	The battery is fully charged and still on power.
3	Damaged	The battery has some problem.
4	Low Battery	The battery is low on charge.
5	Not Installed	The battery is not installed.
6	Unknown	The battery information is not available.

**<x>/DiagMonData/<x>/BatteryLevel**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the current battery level as a percentage (with a range from 0 to 100). In the case where the battery level is not known, the value of the BatteryLevel node SHALL be 0.

**<x>/DiagMonData/<x>/BatteryMan**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This node contains the name of the battery manufacturer.

**<x>/DiagMonData/<x>/BatteryVer**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This node contains the version of the battery.

**<x>/DiagMonData/<x>/BatteryDate**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	date	Get

This node contains the date of the battery manufacture. If the date of manufacture is not available, this value MUST be null.

**<x>/DiagMonData/<x>/BatteryID**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This node contains the ID of the battery.

**<x>/DiagMonData/<x>/BatteryType**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This node contains the type of the battery technology (e.g. NiMH, LiPi).

**<x>/DiagMonData/<x>/BatteryStandbyTime**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

The BatteryStandByTime parameter contains the estimated number of minutes of operation based upon the current drain and charge of all batteries. This value will allow the Server to determine what the estimated standby time is when operating on battery. The time will be the aggregate of all batteries standby time.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 5.2 Memory

### 5.2.1 Introduction

The DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve device memory information which is useful in determining if there is enough device memory or storage to run an application, store SMS, etc.

### 5.2.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- Operations (the function is always available)
- DiagMonConfig (the function does not have any configurable parameters)
- ServerID (The Diagnostics and Monitoring data is either retrieved by using Get command or delivered to the alternate data server).
- Status (the function is always available).

### 5.2.3 Function Description

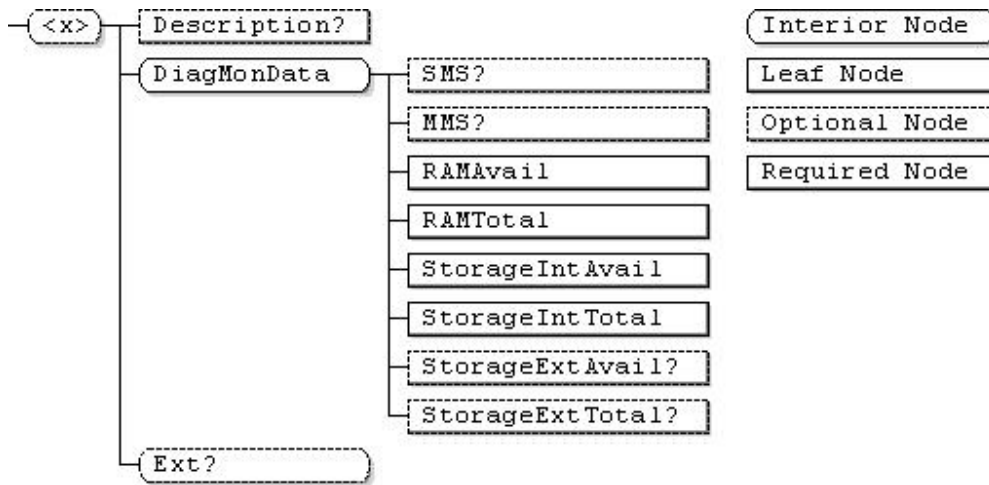


Figure 2 - Memory Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Memory Management Object. Identifier for the Memory MO MUST be: "urn:oma:mo:oma-diag:memory:1.0".

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData/SMS

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This node contains the estimated amount of available SMS storage (expressed in kilobytes).

**<x>/DiagMonData/MMS**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This node contains the estimated amount of available MMS storage (expressed in kilobytes).

**<x>/DiagMonData/RAMAvail**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the estimated current available amount of runtime memory resource for system or application software running (expressed in kilobytes).

**<x>/DiagMonData/RAMTotal**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the total amount of runtime memory resource for system or application software running (expressed in kilobytes).

**<x>/DiagMonData/StorageIntAvail**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the estimated current available amount of storage space, which can store data and software in the Device (expressed in megabytes).

**<x>/DiagMonData/StorageIntTotal**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the total amount of storage space, which can store data and software in the Device (expressed in megabytes).

**<x>/DiagMonData/StorageExtAvail**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the estimated current available amount of external storage space (expressed in megabytes). The external storage can store data and software. The external storage is typically a memory card (i.e. SD card,) or any R/W storage media.



**<x>/DiagMonData/StorageExtTotal**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the total amount of external storage space (expressed in megabytes). The external storage can store data and software. The external storage is typically a memory card (i.e. SD card,) or any R/W storage media

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 5.3 Diagnostic

### 5.3.1 Trap Event Logging Function

#### 5.3.1.1 Introduction

If the device exposes trap event functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

While there are cases in which the trap events should be reported immediately to the management system as they are generated, there are other cases in which flexible event retrieval or reporting mechanism is more desirable, e.g. overnight retrieval by the Server for several related events logged during the day. The Trap Event Logging Function will provide a standard method to selectively collect trap events, of single or multiple types, and then to retrieve the log at the time of interest and/or convenience depending on the characteristics of management tasks.

#### 5.3.1.2 Non-applicable nodes from DiagMon MO definition

None

#### 5.3.1.3 Function Description

The Trap Event Logging Function collects the specified trap events and stores them in the log, which is separately managed by each instance of the function and can be retrieved by the management system through Get command (note that it is not prevented to use periodic report functions described in the DiagMon framework for automated reporting).

Through the Trap Event Logging Function Management Objects described in the next section, the management server can specify one or more trap events to be collected, possibly together with other information related to the events generated, and the way they are stored. In addition, it also allows the Server to remotely control the execution status of the function.

#### 5.3.1.4 Management Objects

Figure below is graphical representation of the Trap Event Logging Function Management Objects.

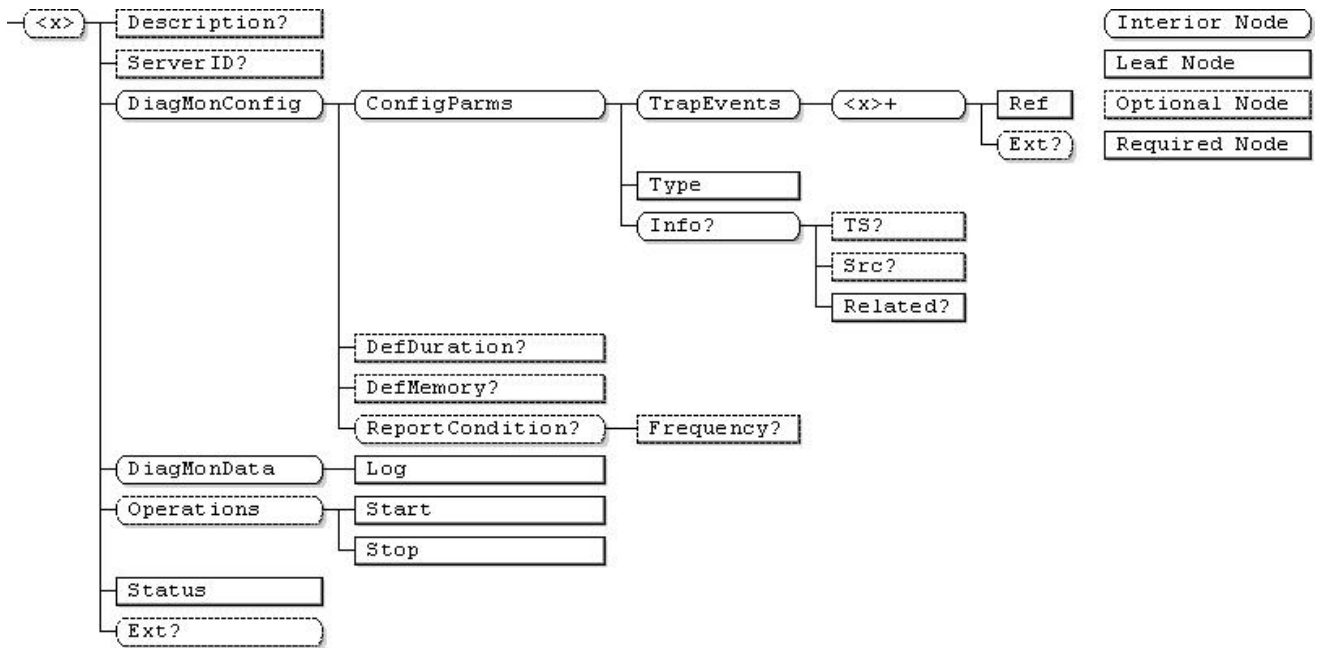


Figure 3 - Trap Event Logging Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Trap Event Management Object. Identifier for the Trap Event MO MUST be: "urn:oma:mo:oma-diag:trapeventlogging:1.0".

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ConfigParms**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ConfigParms/TrapEvents**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is a placeholder for the configuration nodes that are used to specify the events (one or more) to be collected and other information needed for collecting those events. For those specified by these nodes, the Trap Event Logging Function SHALL collect and store events into the log as they occur.

**<x>/DiagMonConfig/ConfigParms/TrapEvents/<x>**

Status	Tree Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is a placeholder for those nodes to specify each event to be collected and other information needed for collecting those trap events.

**<x>/DiagMonConfig/ConfigParms/TrapEvents/<x>/Ref**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the Trap Identifier or vendor specified equivalent to indicate the trap event that is required to be logged.

**<x>/DiagMonConfig/ConfigParms/TrapEvents/<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor specific extension in terms of specifying and collecting trap events.

**<x>/DiagMonConfig/ConfigParms/Type**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the type of logging buffer. The Trap Event Logging Function SHALL reserve some buffer in memory to store the collected trap events, but the size of the buffer is determined by the vendor. It is also up to the vendor to decide which types of buffer to support. In this specification, there are two different types of buffer specified: linear and circular, and hence the content of this node can be “linear” or “circular” (value is case-insensitive).

**<x>/DiagMonConfig/ConfigParms/Info**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node is a placeholder for configuration nodes to specify the elements of information to be collected together with the associated trap events.

**<x>/DiagMonConfig/ConfigParms/Info/TS**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This optional node specifies whether the time stamp is to be recorded as the associated trap event is collected. If the content of this node is “True”, then the time stamp MUST be recorded. Otherwise, or if the node is not present, time stamp is not recorded.

**<x>/DiagMonConfig/ConfigParms/Info/Src**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This optional node specifies whether the source of trap event, i.e. URI of the MO that generated the Trap, is to be recorded as the event is collected. If the content of this node is “True”, then the source information MUST be recorded. Otherwise, or if the node is not present, it is not recorded.

**<x>/DiagMonConfig/ConfigParms/Info/Related**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies other associated information to be collected together with the event.

**<x>/DiagMonConfig/DefDuration**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DefMemory**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition/Frequency**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is a placeholder for the event log node.

**<x>/DiagMonData/Log**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node stores the log of trap events encapsulated in XML format. The XML schema of the data is left to implementation.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

<x>/Operations/Stop

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

<x>/Status

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

<x>/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 5.4 Panic Logs

### 5.4.1 Introduction

If the device exposes panic log functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve panic logs and device crash information. This value is useful in determining if there is a device or application malfunction and retrieve associated information on the event occurrences.

### 5.4.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- DiagMonConfig subnodes (ConfigParms and DefDuration are not necessary)

### 5.4.3 Function Description

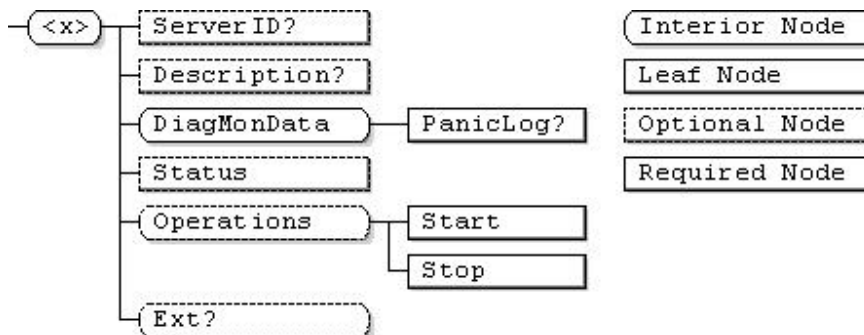


Figure 4 - Panic Logs Function

.../ <x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Panic Log Management Object. Identifier for the Panic Log MO MUST be: "urn:oma:mo:oma-diag:paniclog:1.0".

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonData/PanicLog

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	xml	Get

This node contains the panic log and associated data encapsulated in XML format describing panic/device crash logs. The XML schema of the data is left to implementation.

<x>/Operations

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 5.5 Restart

### 5.5.1 Introduction

If the device supports restart functionality, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to remotely restart a device. As an optional capability, different “restart levels” may be supported to enable varying degrees of device initialization. This function MUST be invoked explicitly.

### 5.5.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- DiagDataURL
- DefDuration
- DefMemory
- ReportCondition
- Status
- Stop



### 5.5.3 Function Description

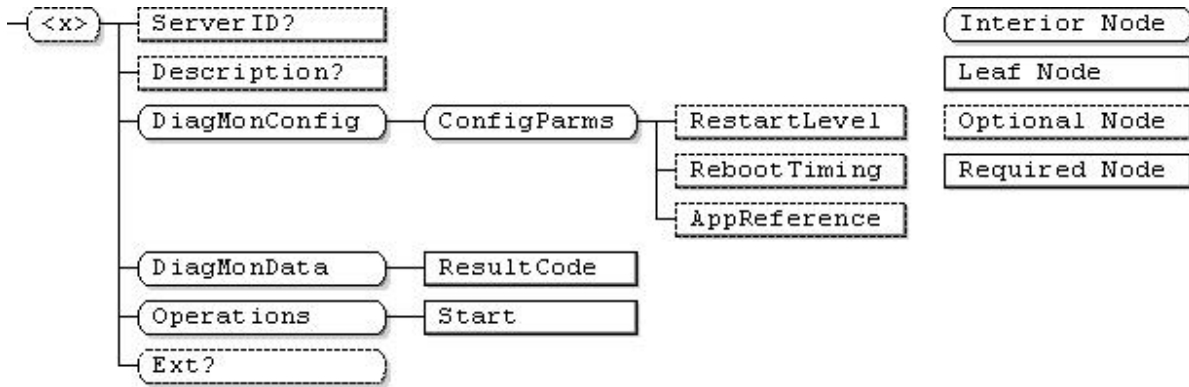


Figure 5 - Restart Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Restart Management Object. Identifier for the Restart MO MUST be: "urn:oma:mo:oma-diag:restart:1.0".

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

## &lt;x&gt;/DiagMonConfig/ConfigParms

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

## &lt;x&gt;/DiagMonConfig/ConfigParms/RestartLevel

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get

This optional node specifies the level at which the device restart operation is.

Integer Value	RestartLevel	Description
10	Restart operating system	Restart the operating system.
20	Restart all applications	Restart all applications
30	Restart single application	Restart of a single application, as identified by the AppReference node
40-100	Reserved	

Note: The DiagMon Function MUST fail if the RestartLevel is set to 30 (Restart Single Application) and the AppReference node is not present or if the value of that node does not map to an application which is currently running on the device.

## &lt;x&gt;/DiagMonConfig/ConfigParms/RebootTiming

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get, Replace

This optional node specifies the timing of the requested reboot.

Integer Value	Restart Timing	Description
10	Immediate	Reboot immediately. This forces the device to reboot immediately.
20	Requested	Reboot requested. This indicates that a graceful reboot has been requested and the device will reboot at the next practical opportunity.
30	Reserved	

## &lt;x&gt;/DiagMonConfig/ConfigParms/AppReference

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get, Replace

This optional node specifies the application to be restarted in the case of single application restart. The semantics of the application reference is up to implementation.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/ResultCode**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node provides the status of the last restart operation. The allowed range of values for this node is as per the following table:

Code	Meaning
1100	No information on status of the last restart operation
1200	Success
1240 - 1249	Success: vendor specified result code for added information
1400	Failure
1402	Failure: reboot by software not supported
1401	Failure: reboot request rejected due to device policy
1470 - 1499	Failure: vendor specified result code for added information

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 5.5.4 Additional Information

The status of this DiagMon function MUST be reported, either asynchronously, using the Generic Alert mechanism [DMPRO] or synchronously, by storing the status in the DM Tree for later retrieval.

Therefore, at least one of the following nodes MUST be defined for this DiagMon Function:

- ServerID (for asynchronous reporting)
- ResultCode (for synchronous reporting)

# 6. Radio Protocol (3GPP, 3GPP2, WiFi, Bluetooth, WiMAX, etc.)

## 6.1 RF Metrics

### 6.1.1 3GPP GSM Metrics

#### 6.1.1.1 Introduction

If the device exposes GSM RF functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve 3GPP GSM RF parameters, measured by the device, that provide an indication of the RF environment at the time of function invocation. The measured parameters include Network Measurement Results as specified in [3GPP-TS\_44018] as well as transmit and receive power and signal quality indicators as specified in [3GPP-TS\_25.215] and [3GPP-TS\_25.225]. This function is only applicable to 3GPP GSM devices or 3GPP UMTS devices operating on GSM.

This function MUST be invoked explicitly. The status of this DiagMon function can be reported asynchronously, using the Generic Alert mechanism [DMPRO] or it can be stored in the DM Tree for later retrieval.

#### 6.1.1.2 Non-applicable nodes from DiagMon MO definition

None

#### 6.1.1.3 Function Description

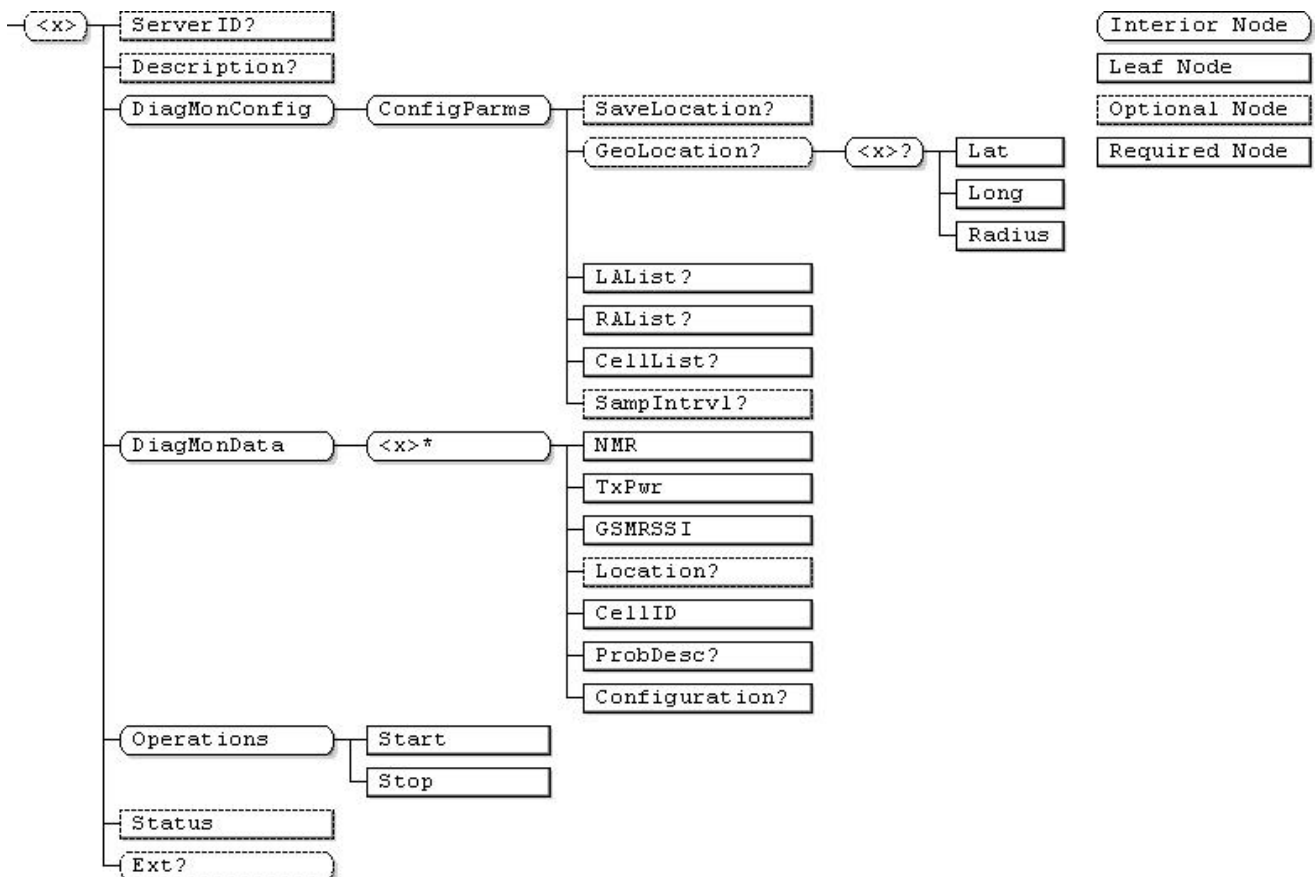


Figure 6 - 3GPP GSM RF Metrics Function

.../&lt;x&gt;

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the 3GPP GSM RF Metrics data. Management Object Identifier for the 3GPP GSM RF Metrics MO MUST be: "urn:oma:mo:oma-diag:RFParms\_3GPP\_GSM:1.0".

&lt;x&gt;/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms/SaveLocation

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get, Replace

This leaf node indicates whether the location information is requested to be saved. If the device is not able to determine the location, it is not required to save it.

**<x>/DiagMonConfig/ConfigParms/GeoLocation**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node acts as a placeholder for circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This node acts as a placeholder for one or more circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Lat**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the latitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Long**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the longitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Radius**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

This leaf node contains the radius of the circular area as defined in [3GPP-TS\_23.032], section 6.3.

**<x>/DiagMonConfig/ConfigParms/LAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of Location Areas. Each LA is defined using LA Identity (LAI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. LAI is specified in 3GPP TS 23.003 section 4.1. The list SHALL be LA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/RAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of Routing Areas. Each RA is defined using RA Identity (RAI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. LAI is specified in 3GPP TS 23.003 section 4.2. The list SHALL be RA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/CellList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of cells. Each cell is defined using Cell Identity (CI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. CI is specified in 3GPP TS 23.003 section 4.3. The list SHALL be CI values separated by commas.

**<x>/DiagMonConfig/ConfigParms/SampIntrvl**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This leaf node contains an integer value indicating the sampling interval in seconds at which Diagnostics and Monitoring data are collected on the device.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node identifies the collected Diagnostics and Monitoring data, which SHOULD be retrieved by the Diagnostics and Monitoring System. The collected DiagMon data results as identified by leaf nodes that are attached to this node are created dynamically and have appropriate format properties as per [DMTND]. Devices MUST support adding of leaf nodes under /<x>/DiagMonData. Devices SHOULD support these leaf nodes formatted in xml.

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This required node is a placeholder for the various collected RF data sets. The node name is constructed as the time the RF parameter measurements were taken, encoded per [ISO8601]



**<x>/DiagMonData/<X>/NMR**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node contains “Network Measurement Results”, as defined by [3GPP-TS\_44.018] “Measurement Results”, encapsulated in XML format.

**<x>/DiagMonData/<X>/TxPwr**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains a measure of the total transmit power, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225], “UE Transmitted Power”.

**<x>/DiagMonData/<X>/GSMRSSI**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the value of the GSM Received Signal Strength Indication, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225], “GSM Carrier RSSI”.

**<x>/DiagMonData/<X>/Location**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	Get

This optional leaf node contains the location where the measurements were taken, with the data encoded as per [3GPP-TS\_23.032], section 7.3.1. Note that if the location is not desired by the server, or if the location data is unavailable, the node will not be present.

**<x>/DiagMonData/<X>/CellID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the value of the cell identifier.

**<x>/DiagMonData/<X>/ProbDesc**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the text describing any problem that may have occurred during the measurement. It is up to the device to determine if a problem description should be attached or not.

**<x>/DiagMonData/<X>/Configuration**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the configuration of the UE at the time of measurement – this node may be used to store data from an error condition.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 6.1.2 3GPP UMTS Metrics

### 6.1.2.1 Introduction

If the device exposes UMTS RF functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve 3GPP UMTS RF parameters, measured by the device, that provide an indication of the RF environment at the time of function invocation. The measured parameters include Network Measurement Results as specified in [3GPP-TS\_44018] as well as transmit and receive power and signal

quality indicators as specified in [3GPP-TS\_25.215] and [3GPP-TS\_25.225]. This function is only applicable to 3GPP UMTS devices.

This function MUST be invoked explicitly. The status of this DiagMon function can be reported asynchronously, using the Generic Alert mechanism [DMPRO] or it can be stored in the DM Tree for later retrieval.

### 6.1.2.2 Non-applicable nodes from DiagMon MO definition

None

### 6.1.2.3 Function Description

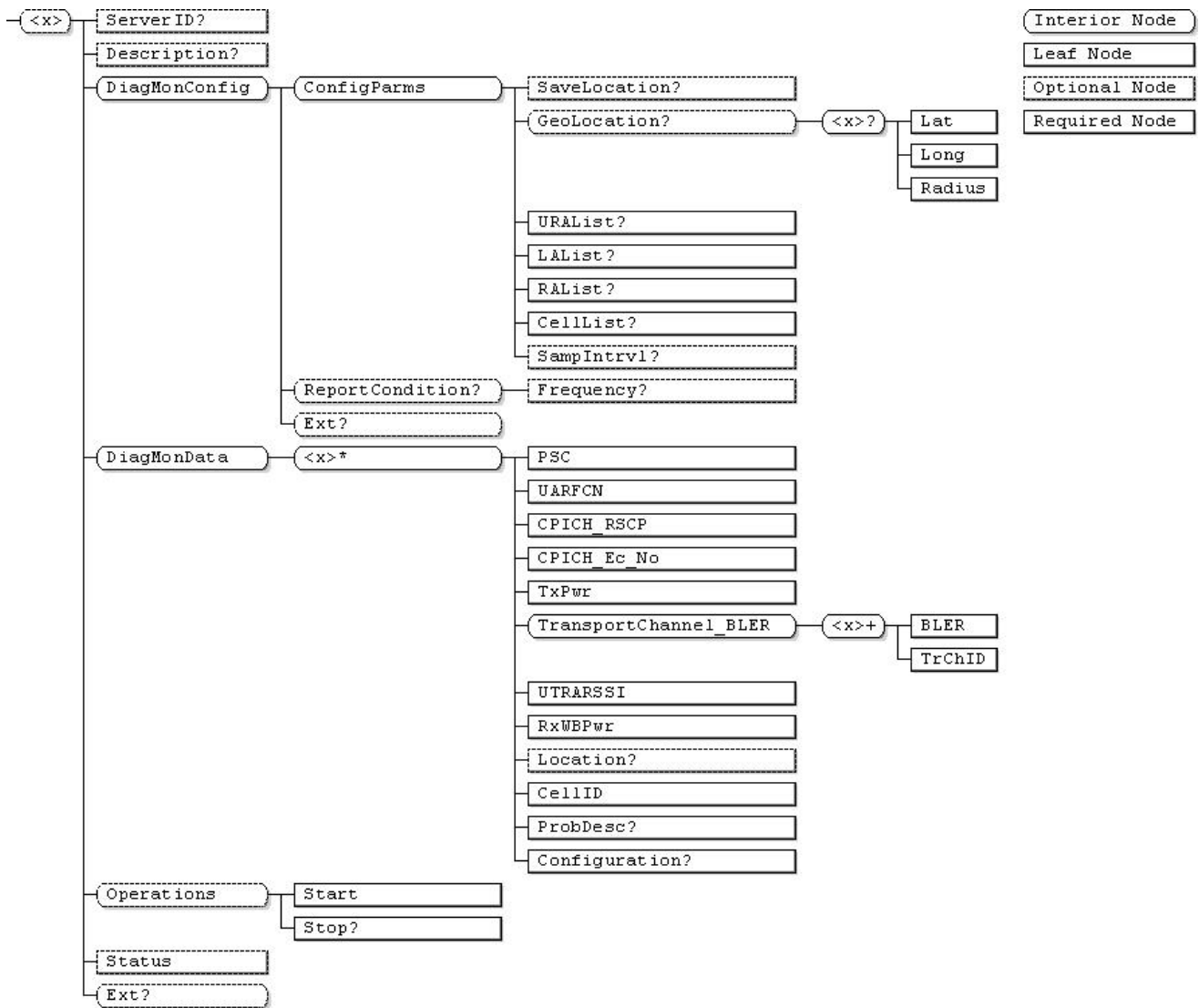


Figure 7 - 3GPP UMTS RF Metrics Function

.../&lt;x&gt;

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the 3GPP UMTS RF Metrics data. Management Object Identifier for the 3GPP UMTS RF Metrics MO MUST be: “urn:oma:mo:oma-diag:RFParms\_3GPP\_UMTS:1.0”.

&lt;x&gt;/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms/SaveLocation

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get, Replace

This leaf node indicates whether the location information is requested to be saved. If the device is not able to determine the location, it is not required to save it.

&lt;x&gt;/DiagMonConfig/ConfigParms/GeoLocation

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node acts as a placeholder for circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This node acts as a placeholder for one or more circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Lat**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the latitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Long**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the longitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Radius**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

This leaf node contains the radius of the circular area as defined in [3GPP-TS\_23.032], section 6.3.

**<x>/DiagMonConfig/ConfigParms/URAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of UTRAN Registration Areas. Each URA is defined using URA Identity, broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. URA Identity is specified in [3GPP-TS\_25.331] section 10.3.2.6. The list SHALL be URA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/LAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of Location Areas. Each LA is defined using LA Identity (LAI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. LAI is specified in 3GPP TS 23.003 section 4.1. The list SHALL be LA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/RAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of Routing Areas. Each RA is defined using RA Identity (RAI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. LAI is specified in 3GPP TS 23.003 section 4.2. The list SHALL be RA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/CellList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node defines the list of cells. Each cell is defined using Cell Identity (CI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. CI is specified in 3GPP TS 23.003 section 4.3. The list SHALL be CI values separated by commas.

**<x>/DiagMonConfig/ConfigParms/SampIntrvl**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This leaf node contains an integer value indicating the sampling interval in seconds at which Diagnostics and Monitoring data are collected on the device.

**<x>/DiagMonConfig/ReportCondition**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node acts as a placeholder to contain different reporting conditions which indicates when Diagnostics and Monitoring data is reported back to the Diagnostics and Monitoring System via Generic Alert and subsequent Get commands mechanism

**<x>/DiagMonConfig/ReportCondition/Frequency**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This leaf node contains an integer value indicating the frequency at which alert about Diagnostics and Monitoring data is reported back to the Diagnostics and Monitoring System. The Diagnostics and Monitoring System MAY use Get command on the DiagMonData node to retrieve diagnostics data.

Note: Each diagnostic function will define what the set of diagnostic data is to be collected.

Values	Description
0	The Device SHOULD NOT send generic alert to Diagnostics and Monitoring System when the set of diagnostics data has been completely collected.
1	The Device MUST send generic alert as soon as practical to Diagnostics and Monitoring System when the set of diagnostics data has been completely collected.
any other value greater than 1	The Device MUST send generic alert as soon as practical to Diagnostics and Monitoring System after the n <sup>th</sup> set of diagnostics data has been completely collected.

**<x>/DiagMonConfig/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node allows for the extensions on report configurations.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node identifies the collected Diagnostics and Monitoring data, which SHOULD be retrieved by the Diagnostics and Monitoring System. The collected DiagMon data results as identified by leaf nodes that are attached to this node are created dynamically and have appropriate format properties as per [DMTND]. Devices MUST support adding of leaf nodes under /<x>/DiagMonData. Devices SHOULD support these leaf nodes formatted in xml.

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This required node is a placeholder for the various collected RF data sets. The node name is constructed as the time the RF parameter measurements were taken, encoded per [ISO8601]

**<x>/DiagMonData/<X>/PSC**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the “Primary Scrambling Code”, as defined by [3GPP-TS\_25.213].

**<x>/DiagMonData/<X>/UARFCN**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the “Downlink UTRA Absolute Radio Frequency Channel Number”, as defined by [3GPP-TS\_25.433].

**<x>/DiagMonData/<X>/CPICH\_RSCP**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the “Common Pilot Channel Received Signal Code Power”, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225].

**<x>/DiagMonData/<X>/CPICH\_Ec\_No**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the ratio of the received energy per PN chip for the Common Pilot Channel to the total received noise power spectral density as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225].

**<x>/DiagMonData/<X>/TxPwr**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains a measure of the total transmit power, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225], “UE Transmitted Power”.

**<x>/DiagMonData/<X>/TransportChannel\_BLER**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a placeholder for the values of the transport channel block error rate measurements and channel IDs



**<x>/DiagMonData/<X>/TransportChannel\_BLER/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This node is a placeholder for the BLER measurements and channel ID.

**<x>/DiagMonData/<X>/TransportChannel\_BLER/<X>/BLER**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the value of the “Transport Channel Block Error Rate” as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225].

**<x>/DiagMonData/<X>/TransportChannel\_BLER/<X>/TrChID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get

This node contains the value of the “Transport Channel Identity” as defined by [3GPP-TS\_25.331].

**<x>/DiagMonData/<X>/UTRARSSI**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the value of the UTRAN Received Signal Strength Indication, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225], “UTRA Carrier RSSI”.

**<x>/DiagMonData/<X>/RxWBPwr**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the value of the total received wideband power, as defined by [3GPP-TS\_25.215] and [3GPP-TS\_25.225], “Received total wide band power”.

**<x>/DiagMonData/<X>/Location**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	Get

This optional leaf node contains the location where the measurements were taken, with the data encoded as per [3GPP-TS\_23.032], section 7.3.1. Note that if the location is not desired by the server, or if the location data is unavailable, the node will not be present.

**<x>/DiagMonData/<X>/CellID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the value of the cell identifier.

**<x>/DiagMonData/<X>/ProbDesc**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the text describing any problem that may have occurred during the measurement. It is up to the device to determine if a problem description should be attached or not.

**<x>/DiagMonData/<X>/Configuration**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the configuration of the UE at the time of measurement – this node may be used to store data from an error condition.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get

See [DiagMonTS] for description of this node.

<x>/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 6.1.3 3GPP LTE Metrics

### 6.1.3.1 Introduction

If the device exposes LTE RF functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve 3GPP LTE RF parameters, measured by the device, that provide an indication of the RF environment at the time of function invocation. The measured parameters include Network Measurement Results as specified in [3GPP-TS\_44018] as well as transmit and receive power and signal quality indicators as specified in [3GPP-TS\_36.214]. This function is only applicable to 3GPP LTE devices.

This function MUST be invoked explicitly. The status of this DiagMon function can be reported asynchronously, using the Generic Alert mechanism [DMPRO] or it can be stored in the DM Tree for later retrieval.

### 6.1.3.2 Non-applicable nodes from DiagMon MO definition

None

### 6.1.3.3 Function Description

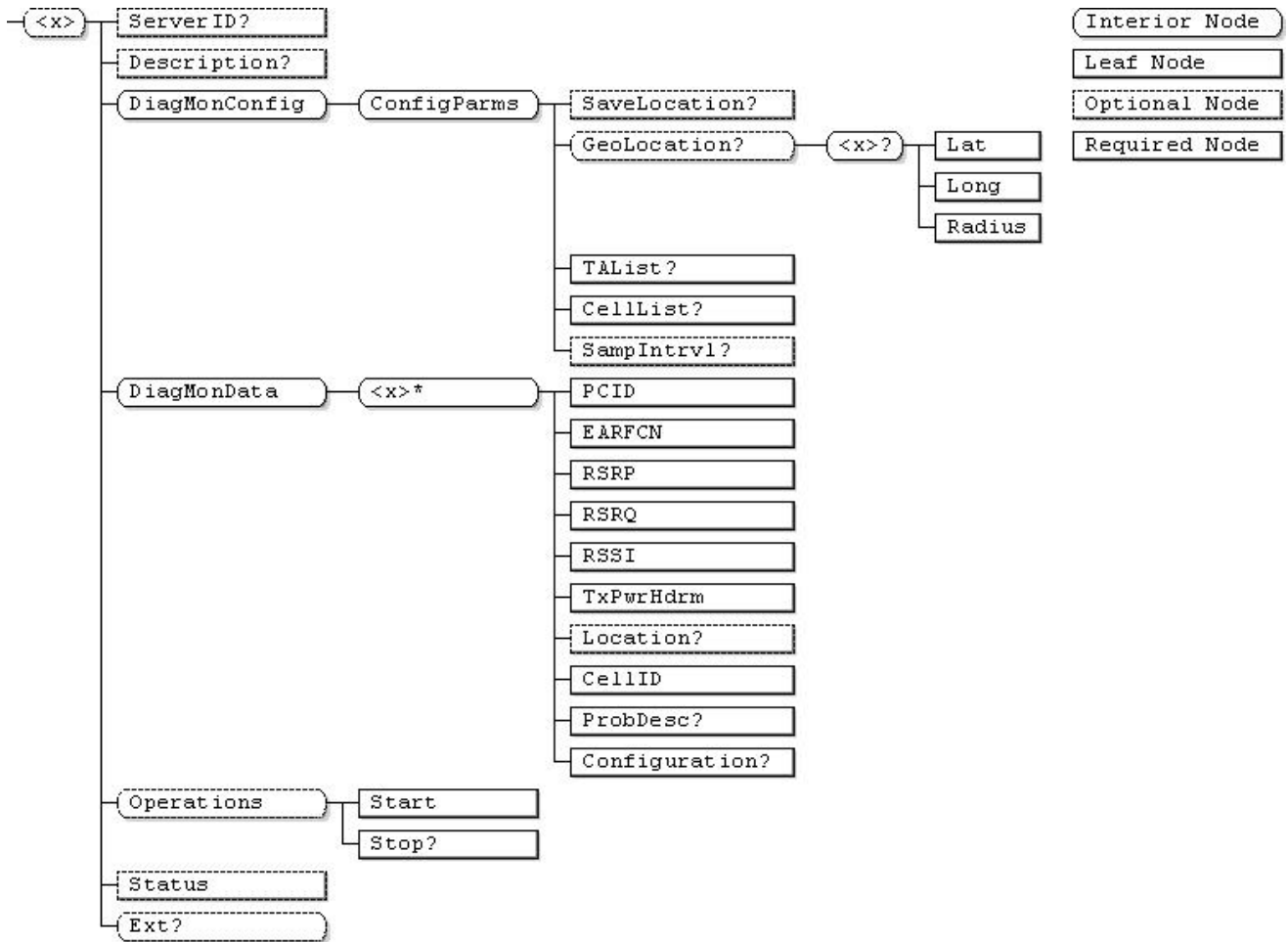


Figure 8 - 3GPP LTE RF Metrics Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the 3GPP LTE RF Metrics data. Management Object Identifier for the 3GPP LTE RF Metrics MO MUST be: “urn:oma:mo:oma-diag:RFParms\_3GPP\_LTE:1.0”.

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ConfigParms**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ConfigParms/SaveLocation**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get, Replace

This leaf node indicates whether the location information is requested to be saved. If the device is not able to determine the location, it is not required to save it.

**<x>/DiagMonConfig/ConfigParms/GeoLocation**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node acts as a placeholder for circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This node acts as a placeholder for one or more circular geographic locations.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Lat**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the latitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Long**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	bin	Get, Replace

This leaf node contains the longitude of the circular area as defined in [3GPP-TS\_23.032], section 6.1.

**<x>/DiagMonConfig/ConfigParms/GeoLocation/<x>/Radius**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get, Replace

This leaf node contains the radius of the circular area as defined in [3GPP-TS\_23.032], section 6.3.

**<x>/DiagMonConfig/ConfigParms/TAList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

This node defines the list of Location Areas. Each TA is defined using TA Identity (TAI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. TAI is specified in [3GPP-TS\_24.301] section 9.9.9.32. The list SHALL be LA values separated by commas.

**<x>/DiagMonConfig/ConfigParms/CellList**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get, Replace

This node defines the list of cells. Each cell is defined using Cell Identity (CI), broadcast by each UTRAN cell via RRC protocol over the Broadcast Channel. CI is specified in 3GPP TS 36.331. The list SHALL be CI values separated by commas.

**<x>/DiagMonConfig/ConfigParms/SampIntrvl**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This leaf node contains an integer value indicating the sampling interval in seconds at which Diagnostics and Monitoring data are collected on the device.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node identifies the collected Diagnostics and Monitoring data, which SHOULD be retrieved by the Diagnostics and Monitoring System. The collected DiagMon data results as identified by leaf nodes that are attached to this node are created dynamically and have appropriate format properties as per [DMTND]. Devices MUST support adding of leaf nodes under <x>/DiagMonData. Devices SHOULD support these leaf nodes formatted in xml.

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This required node is a placeholder for the various collected RF data sets. The node name is constructed as the time the RF parameter measurements were taken, encoded per [ISO8601]

**<x>/DiagMonData/<X>/PCID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the “Physical Cell ID”, as defined by [3GPP-TS\_36.211].

**<x>/DiagMonData/<X>/EARFCN**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the “Downlink E-UTRA Absolute Radio Frequency Channel Number”, as defined by [3GPP-TS\_36.101].

**<x>/DiagMonData/<X>/RSRP**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the “Received Signal Received Power”, as defined by [3GPP-TS\_36.214], encoded in float values.

**<x>/DiagMonData/<X>/RSRQ**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the “Reference Signal Received Quality” (RSRQ) is defined as the ratio  $N \times \text{RSRP} / (\text{E-UTRA carrier RSSI})$ , where N is the number of RB’s of the E-UTRA carrier RSSI measurement bandwidth, as defined by [3GPP-TS\_36.214].

**<x>/DiagMonData/<X>/RSSI**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the “Received Signal Strength Indication” (RSSI), as defined by [3GPP-TS\_36.214].

**<x>/DiagMonData/<X>/TxPwrHdrm**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains a measure of the power headroom (device maximum transmit power minus current transmit power), as defined by [3GPP-TS\_36.213], “Power Headroom”.

**<x>/DiagMonData/<X>/Location**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	xml	Get

This node contains the location where the measurements were taken, with the data encoded as per [3GPP-TS\_23.032], section 7.3.1. Note that if the location is not desired by the server, or if the location data is unavailable, the node will not be present.

**<x>/DiagMonData/<X>/CellID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This optional leaf node contains the value of the cell identifier – the CellGlobalIDeutra as defined in [3GPP-TS\_36.331]

**<x>/DiagMonData/<X>/ProbDesc**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the text describing any problem that may have occurred during the measurement. It is up to the device to determine if a problem description should be attached or not.

**<x>/DiagMonData/<X>/Configuration**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node contains the configuration of the UE at the time of measurement – this node may be used to store data from an error condition.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

See [DiagMonTS] for description of this node.



**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 6.2 Air Interface Metrics

### 6.2.1 Mobility Management Rejection

#### 6.2.1.1 Introduction

If the device operates on a 3G network, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function will allow the Server to retrieve information associated with the failure of a device to establish connection with a mobile network. This function stores Mobility Management Reject Codes (as defined in [3GPP-TS\_24.008]), along with Cell ID and time/date stamp associated with each connection failure.

#### 6.2.1.2 Non-applicable nodes from DiagMon MO definition

DiagMonConfig/ConfigParms

### 6.2.1.3 Function Description

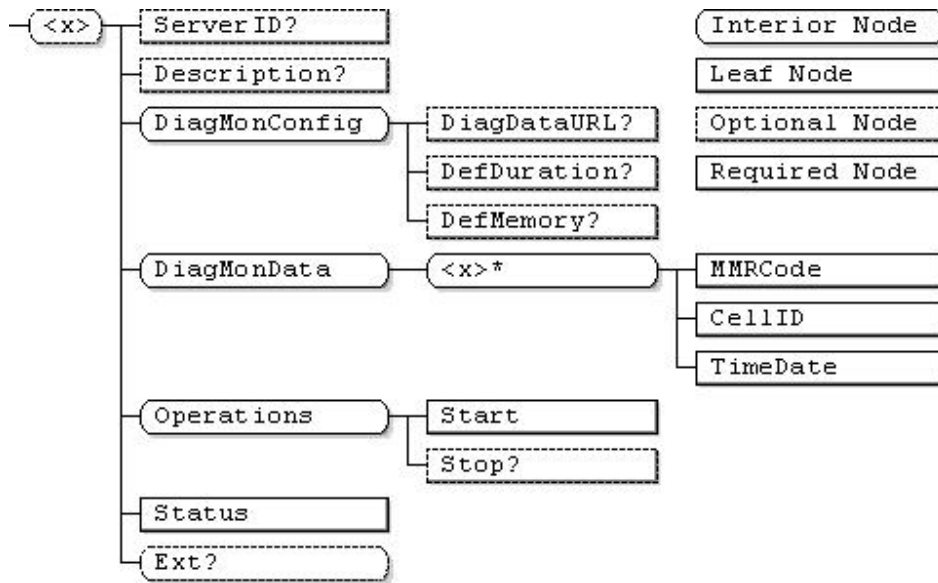


Figure 9 - Mobility Management Rejection Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Mobility Management Rejection Management Object. Identifier for the Mobility Management Rejection MO MUST be: "urn:oma:mo:oma-diag:mobility\_management\_rejection:1.0".

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DiagDataURL**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DefDuration**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DefMemory**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

This node identifies the collected Diagnostics and Monitoring data, which SHOULD be retrieved by the Diagnostics and Monitoring System. The collected DiagMon data results as identified by leaf nodes that are attached to this node are created dynamically and have appropriate format properties as per [DMTND]. Devices MUST support adding of leaf nodes under /<x>/DiagMonData.

**<x>/DiagMonData/<x>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for Mobility Management Reject information.

**<x>/DiagMonData/<x>/MMRCode**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the Mobility Management Reject Cause code as defined in Table 10.5.95 of [3GPP-TS\_24.008].

**<x>/DiagMonData/<x>/CellID**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the Cell ID of the serving cell associated with the Mobility Management Reject.

**<x>/DiagMonData/<x>/TimeDate**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node contains the Time and Date associated with the Mobility Management Reject, encoded per the UTC based [ISO8601] basic format.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

<x>/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 7. IP Layer

### 7.1 Data Session Metrics

#### 7.1.1 Data Call and Data Session

##### 7.1.1.1 Introduction

If the device supports data calls, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

The Data Call and Data Session function will allow a DiagMon enabler to collect metrics and details of each data call and of each data session.

##### 7.1.1.2 Non-applicable nodes from DiagMon MO definition

None

##### 7.1.1.3 Function Description

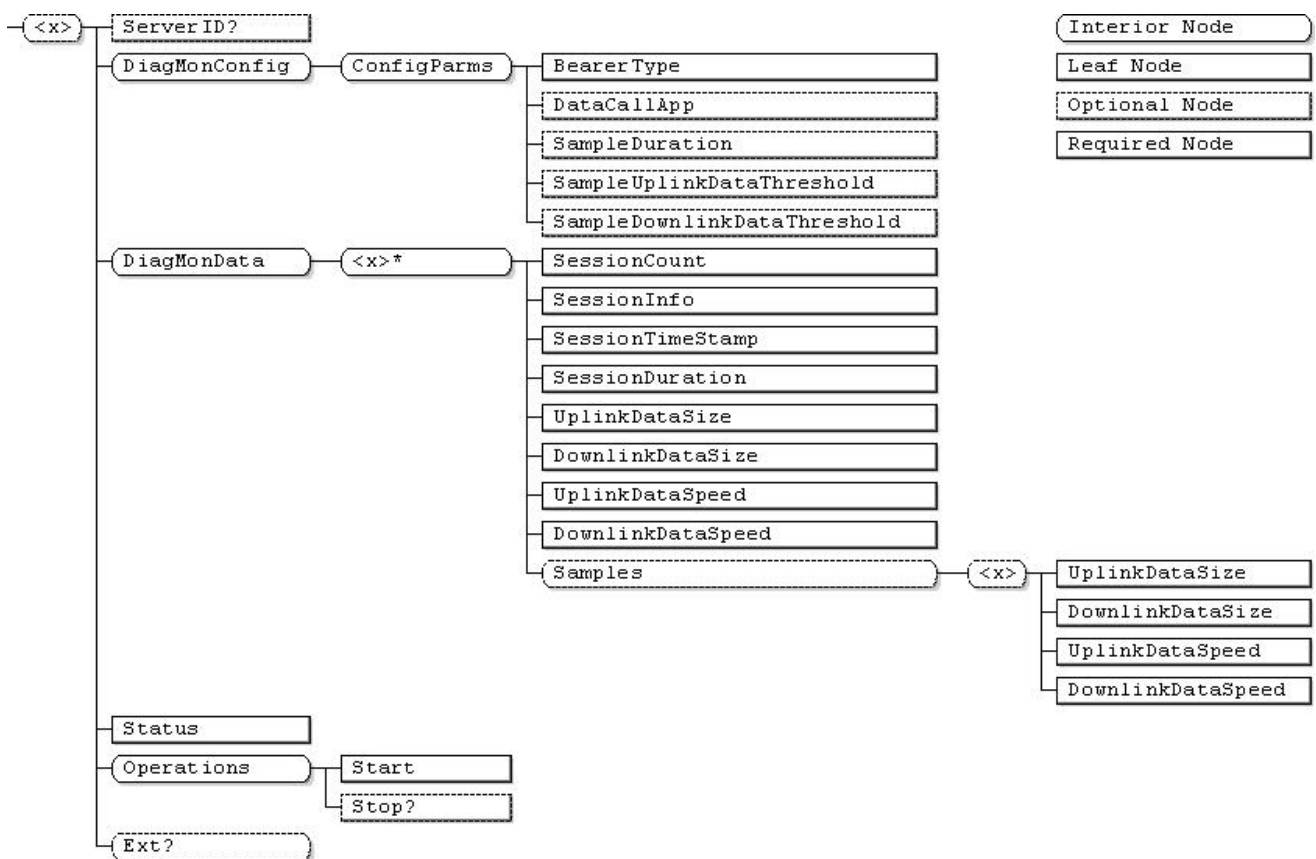


Figure 10 - Data Call and Data Session Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Data Call and Data Session Management Object. Identifier for the Data Call and Data Session MO MUST be: "urn:oma:mo:oma-diag:DataCallAndDataSession:1.0".

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node

<x>/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node

<x>/DiagMonConfig/ConfigParms

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node

<x>/DiagMonConfig/ConfigParms/BearerType

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node specifies the BearerType for which all the individual data sessions information associated with the different data calls are collected. The values of the BearerType are defined in the DevDetail/CBT leaf node description in [DMSTDOBJ1\_3].

<x>/DiagMonConfig/ConfigParms/DataCallApp

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	chr	Get

This optional node specifies the Data Call application for which all the individual data sessions information with the relevant data call are collected. The formats of the DataCallApp are vendor specific.

**<x>/DiagMonConfig/ConfigParms/SampleDuration**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get

This node specifies the duration in seconds of data samples collected during data session.

**<x>/DiagMonConfig/ConfigParms/SampleUplinkDataThreshold**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get

This node specifies the minimum number of uplink data (in kilobytes) before data is to be recorded.

**<x>/DiagMonConfig/ConfigParms/SampleDownlinkDataThreshold**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get

This node specifies the minimum number of Downlink data (in kilobytes) before data is to be recorded.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for all the collected data sessions information of the data calls.

**<x>/DiagMonData/<X>/SessionCount**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the information about the serial counter of the session. The counter resets to zero on Start operation. For every subsequent session the counter increments by one.

**<x>/DiagMonData/<X>/SessionInfo**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node contains the information about the session such as bearer type, data call application and other supported information (e.g. protocol information on which session has happened, such as over HTTP, SIP, etc.), encapsulated in XML format.

Note that the XML schema of the data is left to implementation.



**<x>/DiagMonData/<X>/SessionTimeStamp**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node contains the time stamp of a session start.

The session start value is the time and date expressed as a UTC based [ISO8601] basic format.

**<x>/DiagMonData/<X>/SessionDuration**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the duration of the session expressed in seconds.

**<x>/DiagMonData/<X>/UplinkDataSize**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the amount of data uploaded in a session, expressed in kilobytes.

**<x>/DiagMonData/<X>/DownlinkDataSize**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the amount of data downloaded in a session, expressed in kilobytes.

**<x>/DiagMonData/<X>/UplinkDataSpeed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the speed of the uplink data of a session, expressed in kilobytes per second.

**<x>/DiagMonData/<X>/DownlinkDataSpeed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the speed of the downlink data of a session, expressed in kilobytes per second.

**<x>/DiagMonData/<X>/Samples**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node acts as a placeholder for all the samples collected if a SampleDuration valid value is set.

**<x>/DiagMonData/<X>/Samples/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This required node is a placeholder for the various collected data sets. The node name is constructed as the time the sample measurements were taken, encoded per [ISO8601]

**<x>/DiagMonData/<X>/Samples/<X>/UplinkDataSize**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the amount of data uploaded in a sample, expressed in kilobytes

**<x>/DiagMonData/<X>/Samples/<X>/DownlinkDataSize**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the amount of data downloaded in a sample, expressed in kilobytes

**<x>/DiagMonData/<X>/Samples/<X>/UplinkDataSpeed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the speed of the uplink data of a sample, expressed in kilobytes per second

**<x>/DiagMonData/<X>/Samples/<X>/DownlinkDataSpeed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	float	Get

This node contains the speed of the downlink data of a sample, expressed in kilobytes per second

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get

See [DiagMonTS] for description of this node

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node

## 8. Application Layer

### 8.1 SMS

#### 8.1.1 SMS Options

##### 8.1.1.1 Introduction

If the device exposes SMS option functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

It is possible for the SMS options to be changed, and since these options are only stored on the device, the Server will need to be able to query these options.

This continuously available function will allow the Server to determine SMS options, e.g. data coding.

##### 8.1.1.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- Operations (the function is always available)
- DiagMonConfig (the function does not have any configurable parameters)
- ServerID (The Diagnostics and Monitoring data is either retrieved by using Get command or delivered to the alternate data server).
- Status (the function is always available)

##### 8.1.1.3 Function Description

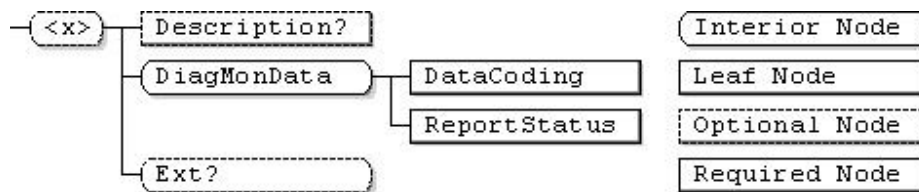


Figure 11 - SMS Options Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the SMS Options Management Object. Identifier for the SMS Options MO MUST be: “urn:oma:mo:oma-diag:SMSOptions:1.0”.

**<x>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/DataCoding**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node contains the data coding method of the short messages. The value MUST be one of the following values (which are case-insensitive) as defined in section 4 “SMS Data Coding Scheme” of [3GPP-TS\_23.038]:

Data Coding Scheme	Description
7bit	GSM 7 bit default alphabet.
ucs2	UCS2 (16bit).
8bit	8 bit data.

**<x>/DiagMonData/ReportStatus**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node indicates whether the delivery report is switched on. The value MUST be one of the following values:

Value	Meaning	Description
1	on	The delivery report option is switched on within Device.
0	off	The delivery report option is switched off within Device.
10	not supported	The delivery report option is not supported by the Device.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 8.1.2 SMS Usage

### 8.1.2.1 Introduction

If the device exposes SMS usage functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

There are circumstances where it is very useful for a Server to query a device about the number of SMS messages that have been sent and received over a period of time. The SMS Usage function will allow a Server to start the Client counting the number of SMS messages sent and received, and allow the Server to sample the values while the function is running.

The values of SMSSent and SMSReceived SHOULD be available on a real-time basis.

### 8.1.2.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- DiagMonConfig (does not have any configurable parameters).

### 8.1.2.3 Function Description

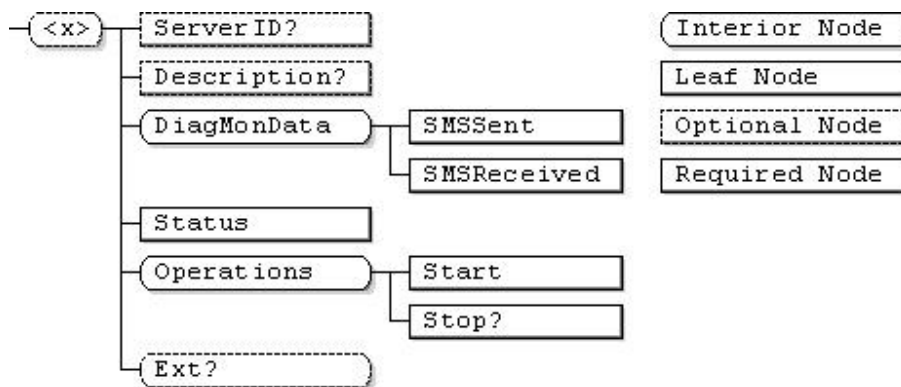


Figure 12 - SMS Usage Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the SMS Usage Management Object. Identifier for the SMS Usage MO MUST be: “urn:oma:mo:oma-diag:SMSUsage:1.0”.

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/SMSsent**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the number of short messages sent since the Start was executed. Note that Exec on Start will reset the value.

**<x>/DiagMonData/SMSReceived**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the number of short messages received since the Start was executed. Note that Exec on Start will reset the value.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

<x>/Operations/Stop

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

See [DiagMonTS] for description of this node.

<x>/Ext

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 8.2 MMS

### 8.2.1 MMS Usage

#### 8.2.1.1 Introduction

If the device exposes MMS usage functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

There are circumstances where it is very useful for a Server to query a device about the number of MMS messages that have been sent and received over a period of time. The MMS Usage function will allow a Server to start the Client counting the number of MMS messages sent and received, and allow the Server to sample the values while the function is running.

The values of MMSSent and MMSReceived SHOULD be available on a real-time basis.

#### 8.2.1.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- DiagMonConfig (the function does not have any configurable parameters).

#### 8.2.1.3 Function Description

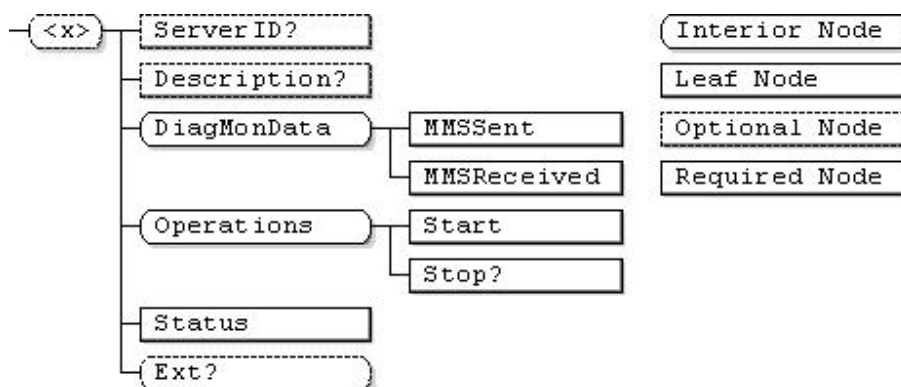


Figure 13 - MMS Usage Function



.../&lt;x&gt;

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the MMS Usage Management Object. Identifier for the MMS Usage MO MUST be: "urn:oma:mo:oma-diag:MMSUsage:1.0".

&lt;x&gt;/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonData

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonData/MMSSent

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the number of MMS sent since the Start was executed. Note that Exec on Start will reset the value.

&lt;x&gt;/DiagMonData/MMSReceived

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the number of MMS received since the Start was executed. Note that Exec on Start will reset the value.

&lt;x&gt;/Operations

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	null	Exec

See [DiagMonTS] for description of this node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 8.3 Browsing

### 8.3.1 Browsing Usage

#### 8.3.1.1 Introduction

The DiagMon Client SHOULD support this function. The DiagMon Server MUST support this function.

The Browsing Usage function will allow a DiagMon enabler to retrieve specific information about each browser which is installed and collect the amount of data and time used by the associated browser. Note that the extra information and the usage of data and time depend upon the support of the specific browser.

Additionally, the total data & time used by all supporting browsers will optionally be available.

After the device performs an Exec on the 'Start' node, the Client MUST set the value of DataUsage and TimeUsage to zero and start collecting the amount of data and time used. When the device performs an Exec on the 'Stop' node, the Client MUST stop collecting the data.

#### 8.3.1.2 Non-applicable nodes from DiagMon MO definition

The following nodes MUST NOT be used for this function:

- DiagMonConfig (the function does not have any configurable parameters)

### 8.3.1.3 Function Description

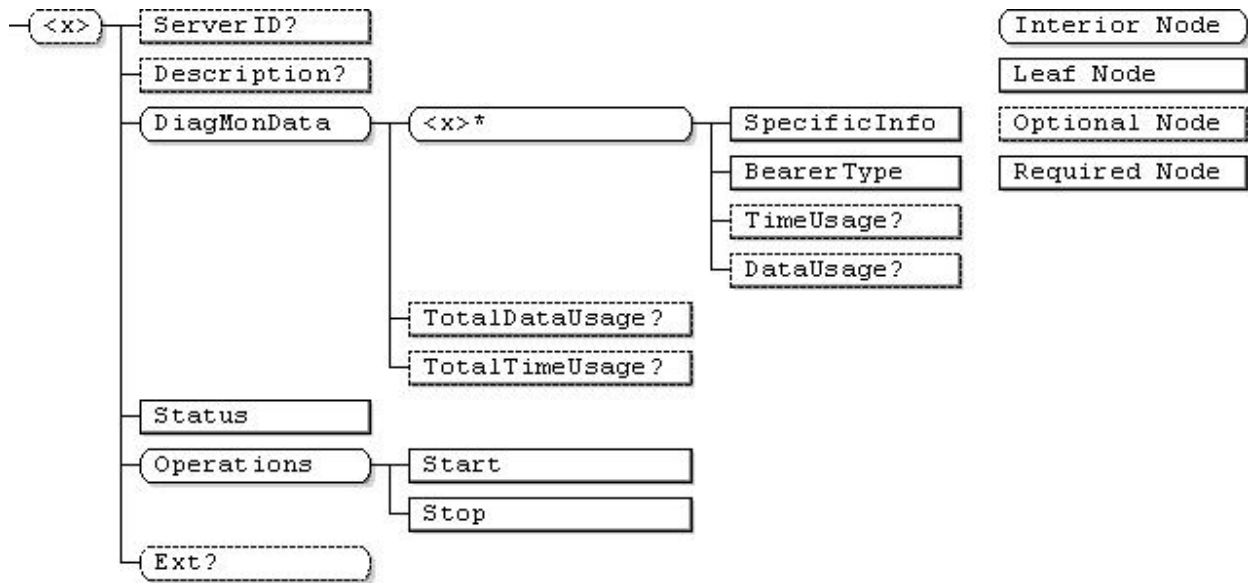


Figure 14 - Browsing Usage Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Browsing Usage Management Object. Identifier for the Browsing Usage MO MUST be: “urn:oma:mo:oma-diag:BrowsingUsage:1.0”.

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node is a placeholder for the data nodes.

**<x>/DiagMonData/<X>/SpecificInfo**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node contains the browser specific information such as browser name, version and other supported features encapsulated in XML format. The XML schema of the data is left to implementation.

**<x>/DiagMonData/<X>/BearerType**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the bearer type information, for which the usage information has been collected. The values of the BearerType are defined in the DevDetail/CBT leaf node description in [DMSTDOBJ1\_3].

**<x>/DiagMonData/<X>/TimeUsage**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the accumulated amount of time the specific browser was used, expressed in minutes.

**<x>/DiagMonData/<X>/DataUsage**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the amount of data used by this specific browser, expressed in megabytes.

**<x>/DiagMonData/TotalDataUsage**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the aggregated amount of data used by all the browsers, expressed in megabytes.

**<x>/DiagMonData/TotalTimeUsage**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This optional node contains the accumulated amount of time all browsers were used, expressed in minutes.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 8.4 Application Monitoring

### 8.4.1 Introduction

If the device exposes application monitoring functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve application usage metrics, e.g. usage frequency, the last usage date, etc. These metrics are useful in determining how often and how long an application has been used. A service provider could use this information to better allocate scarce R&D funds to support the more popular applications. The DiagMon Server will be required to start and stop the function.

### 8.4.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- DiagMonConfig/ConfigParms

### 8.4.3 Function Description

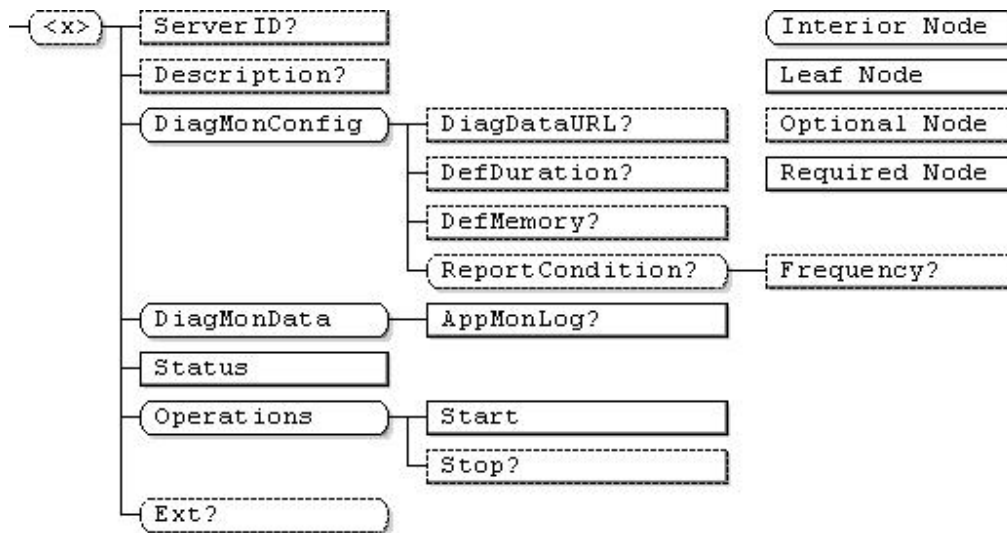


Figure 15 - Application Monitoring Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Application Monitoring Management Object. Identifier for the Application Monitoring MO MUST be: “urn:oma:mo:oma-diag:appmonlog:1.0”.

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DiagDataURL**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DefDuration**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/DefMemory**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition/Frequency**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/AppMonLog**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	xml	Get

This node contains the application usage metrics data (e.g. usage frequency, the last usage date, etc) encapsulated in XML format. The XML schema of the data is left to implementation.

The value of Frequency node defined in [DiagMonTS] specifies the rule when the device needs to report the metrics data to DiagMon Server or data server. The DiagMon Server can also obtain the metrics data using Get command targeting the AppMonLog node.

**<x>/Status**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Start**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.



## 8.5 NFC (Near Field Communication)

### 8.5.1 Introduction

If the device exposes NFC functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

The scope of the NFC Function is to allow the Diagnostic and Monitoring server to evaluate the status of a NFC (Near Field Communication) enabled device and retrieve information on the status and interconnection of its hardware components.

### 8.5.2 Non-applicable nodes from DiagMon MO definition

- DiagMonConfig/ConfigParms

### 8.5.3 Function Description

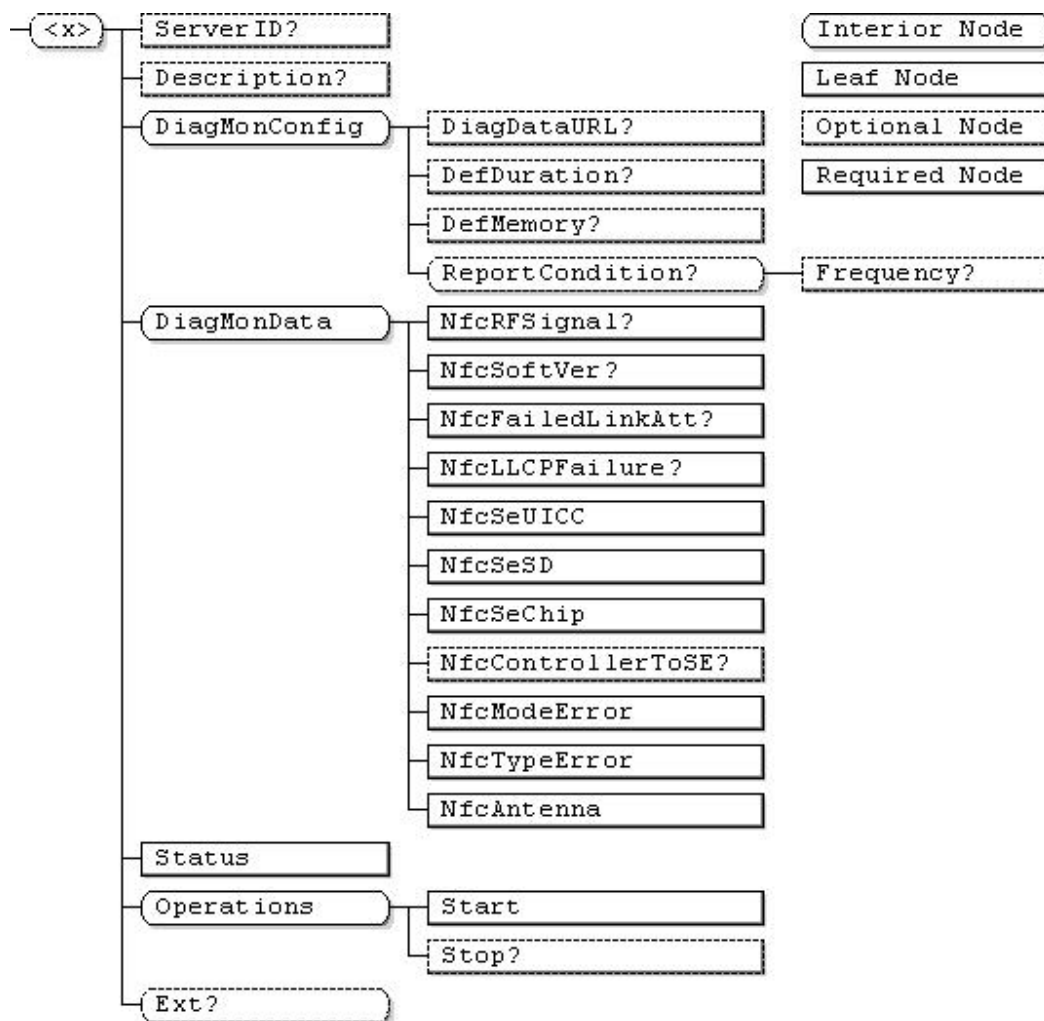


Figure 16 - NFC Info Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the NFC Management Object. Identifier for the NFC MO MUST be: "urn:oma:mo:oma-diag:NFC:1.0".

<x>/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonConfig/DiagDataURL

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonConfig/DefDuration

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

<x>/DiagMonConfig/DefMemory

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonConfig/ReportCondition/Frequency**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/NfcRFSignal**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	xml	Get

This node contains the log and associated data encapsulated in XML format describing RF Signal failures, including: type and sub-type of technology used, index modulation, communication result (including error code), average bit error and date and time of the failure.

Data description and associated XML schema is added in Appendix C 1

**<x>/DiagMonData/NfcSoftVer**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node collects the Software version installed in the controller.

<x>/DiagMonData/NfcFailedLinkAtt

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

Differently from the information provided by the NfcRFSignal, this node collects logs from failed link attempts between two NFC entities where the issue is not related to RF Signal failures.

Logs MUST include specific error code(s), date and time when the error(s) occurred.

Data	Description
Date and Time: [ISO8601] basic format	UTC based date and time of the failure
ErrorCode: <data>	Error Code

<x>/DiagMonData/NfcLLCPFailure

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node collects error logs related to failures within the LLCP layer (The LLCP is only used for Peer-2-Peer Communication and allows upper layers to have a reliable bi-directional data link over Lower layer - NFC Interface and Protocol).

Logs MUST include specific error code(s), date and time when the error(s) occurred.

<x>/DiagMonData/NfcSeUICC

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node defines the number of UICCs embedded as Secure Element in the NFC device.

<x>/DiagMonData/NfcSeSD

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node defines the number of SD Cards embedded as Secure Element in the NFC device.

<x>/DiagMonData/NfcSeChip

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node defines the number of chips embedded as Secure Element in the NFC device.

**<x>/DiagMonData/NfcControllerToSE**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This optional node collects log and associated data describing communication failures between the controller and the Secure Element(s), including: specific SE where the issue occurred, specific error code, and date and time of the error.

Secure Element	Description
1	Chip
2	UICC
3	SD Card

Data	Description
Secure Element: <1 to 3>	Secure Element
ErrorCode: <data>	Error Code
Date and Time: [ISO8601] basic format	UTC based date and time of the failure

**<x>/DiagMonData/NfcModeError**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node collects error log and associated data encapsulated in XML format describing failures within the NFC controller about mode switch.

Log to include: Date, time, NFC mode attempted to contact on the last failure(s) and last successful NFC mode. Data description and associated XML schema is added in Appendix C 2

Mode	Description
1	P2P
2	Read & Write
3	Card Emulation

<x>/DiagMonData/NfcTypeError

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node collects error log and associated data encapsulated in XML format describing failures within the NFC controller about type switch.

Log to include: Date, time and NFC technology type attempted to contact on the last failure(s).

Data description and associated XML schema is added in Appendix C 3

Type	Description
1	ISO 14443 Type A
2	ISO 14443 Type B
3	ISO 18092 Type F

<x>/DiagMonData/ NfcAntenna

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node specifies whether the NFC Controller can communicate with an NFC antenna

Presence	Description
1	Yes
0	No

<x>/Status

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get

See [DiagMonTS] for description of this node.

<x>/Operations

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

<x>/Operations/Start

Status	Tree Occurrence	Format	Min. Access Types
Required	One	null	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Stop**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	null	Get

See [DiagMonTS] for description of this node.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 9. User Interface Layer

### 9.1 Settings

#### 9.1.1 User Equipment Setting

##### 9.1.1.1 Introduction

If the device exposes equipment setting functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

The User Equipment Setting function is useful, when a customer care staff will examine the user configurable setting (e.g. 'LCD brightness', 'ring tone volume', and etc.), and determine any issues, and may update the setting if needed.

The User Equipment settings information can be retrieved in full or partial with specifying its category.

Since this function is always available, no start/stop operations are allowed.

A typical use case for this function is:

1. The DiagMon Client updates the value of 'UESetting' as per the 'Category' on a regular basis as per the 'RefreshInterval'
2. The DiagMon Server retrieves the user setting values from the 'UESetting' node from the DiagMon Client.
3. The DiagMon Server puts the new values in 'ReplacementSetting', then performs an Exec command on 'Operation/Modify' node. The DiagMon Client replaces the user configurable setting with them.

##### 9.1.1.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- Status (the function is always available)
- Operations/Start and Operations/Stop (the function is always available)

##### 9.1.1.3 Function Description

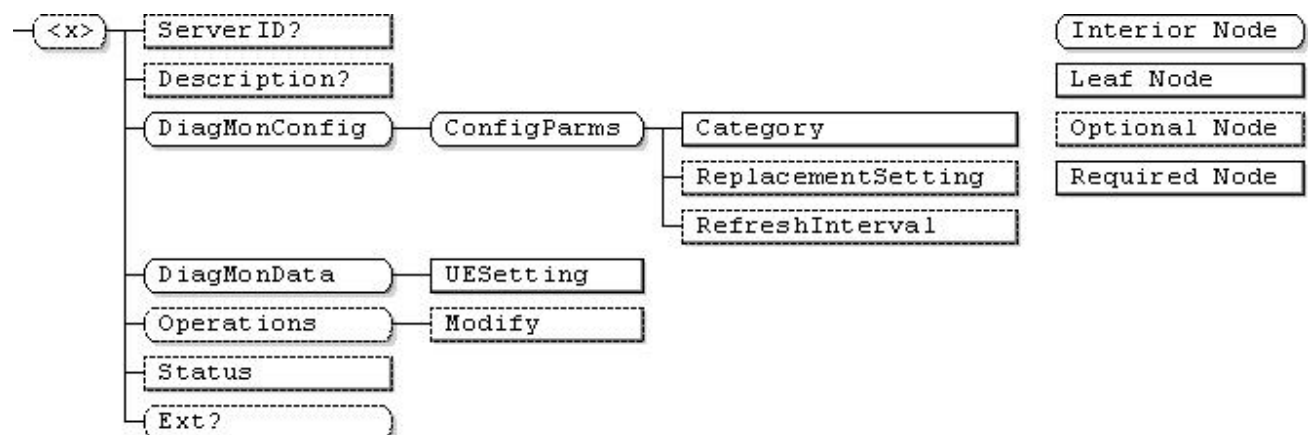


Figure 17 - User Equipment Setting Function



.../&lt;x&gt;

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the User Equipment Settings Management Object. Identifier for the User Equipment Settings MO MUST be: “urn:oma:mo:oma-diag:UESetting:1.0”.

&lt;x&gt;/ServerID

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/Description

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

&lt;x&gt;/DiagMonConfig/ConfigParms/Category

Status	Tree Occurrence	Format	Min. Access Types
Required	One	chr	Get, Replace

This node specifies the category of the group of setting to be retrieved. The value of the string is up to implementation

**<x>/DiagMonConfig/ConfigParams/ReplacementSetting**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	xml	Replace

This optional node specifies the replacing setting values. This node contains the replacing setting key and value pairs encapsulated in XML format. The XML schema of this value is left to implementation.

Replacement MUST be done when an Exec command on Operations/Modify node is issued. If this replacement functionality is supported by the device, then the nodes 'Operations', 'Operations/Modify', and 'DiagMonConfig/ConfigParams/ReplacementSetting' MUST be present.

**<x>/DiagMonConfig/ConfigParams/RefreshInterval**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get, Replace

This optional node indicates the delay of synchronizing device data with the value of UESetting node. If the node exists, a value of 0 indicates that the values are always in sync. If the node is not present, the synchronization is left to implementation. The value is expressed in minutes.

**<x>/DiagMonData/UESetting**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	xml	Get

This node contains the current User Equipment setting data encapsulated in XML format describing user configurable setting values (e.g. setting name, setting value, etc.) The XML schema of the value is left to implementation.

**<x>/Operations**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/Operations/Modify**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	null	Exec

After the Device performs an Exec on the Modify node, the Client MUST set the UE setting with the values of ConfigParams/ReplacementSetting.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## 9.2 Phone Book

### 9.2.1 Introduction

If the device exposes phonebook functionality to the DiagMon Client, then the DiagMon Client MUST support this function. The DiagMon Server MUST support this function.

This function allows the Diagnostics and Monitoring system to retrieve device phone book usage information. This value is useful in determining phone book usage status and retrieves associated information.

### 9.2.2 Non-applicable nodes from DiagMon MO definition

The following nodes SHOULD NOT be used for this function:

- Operations (the function is always available)
- DiagMonConfig (the function does not have any configurable parameters)
- ServerID (The Diagnostics and Monitoring data is either retrieved by using Get command or delivered to the alternate data server).
- Status (the function is always available).

### 9.2.3 Function Description

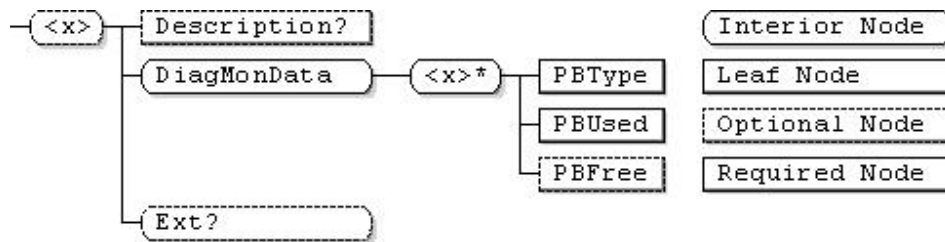


Figure 18 - Phone Book Function

.../<x>

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	node	Get

This interior node is a placeholder for the Phonebook Management Object. Identifier for the Phonebook MO MUST be: “urn:oma:mo:oma-diag:phonebook:1.0”.

**<x>/Description**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	node	Get

See [DiagMonTS] for description of this node.

**<x>/DiagMonData/<X>**

Status	Tree Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node is a placeholder for the data nodes.

**<x>/DiagMonData/<X>/PBType**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the current phonebook database type, and the valid values are:

Type	Description
0	TotalUsage
1	DeviceInternalMemory
2	SmartCard
3	External (MicroSD Card, etc.)
4-9	Reserved for vendor specific types

**<x>/DiagMonData/<X>/PBUsed**

Status	Tree Occurrence	Format	Min. Access Types
Required	One	int	Get

This node contains the number of contact records already saved in the current database.

**<x>/DiagMonData/<X>/PBFree**

Status	Tree Occurrence	Format	Min. Access Types
Optional	One	int	Get

This optional node contains the estimated number of additional contact records which could be saved in the current database.

Note: PBFree is optional, because some mobile devices provide practically unlimited storage, for example, internal memory of 16GB or more.

**<x>/Ext**

Status	Tree Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

See [DiagMonTS] for description of this node.

## Appendix A. Change History

(Informative)

### A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

### A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft versions OMA-TS-DiagMon_Functions-V1_0_0	12 Dec 2007	All	Initial baseline document
Draft versions OMA-TS-DiagMon_Functions-V1_0	26 Jun 2008	All	OMA-DM-Diag-2008-0013R04 OMA-DM-Diag-2008-0015R01 (partial) OMA-DM-Diag-2008-0021R02
	2 Sep 2008	A.2	OMA-DM-Diag-2008-0015R01 (complete) Editorial changes OMA-DM-Diag-2008-0030 OMA-DM-Diag-2008-0028 OMA-DM-Diag-2008-0039R01
	10 Oct 2008	2.1,5.4.1, 5.4.1, 10.5, 10.5.	OMA-DM-Diag-2008-0023R01 OMA-DM-Diag-2008-0044R02
	16 Oct 2008	5.4.1.4., 10.1.1.4., 10.5.4.1	Editorial changes
	20 Oct 2008	10.1.1.	OMA-DM-Diag-2008-0039R06
	22 Oct 2008	10.1.2.	OMA-DM-Diag-2008-0035R03
	23 Oct 2008	All	Editorial
	01 Dec 2008	9.1	OMA-DM-Diag-2008-0041R04 OMA-DM-Diag-2008-0067R01
	04 Feb 2009	5.1.1.4	OMA-DM-Diag-2008-0060R02
	09 Mar 2009	All 5.6	OMA-DM-Diag-2009-0013R01 OMA-DM-Diag-2009-0014R01
	27 Apr 2009	5, 9, B	OMA-DM-Diag-2009-0025 OMA-DM-Diag-2009-0030 Editorial changes
	04 May 2009	5 All	OMA-DM-DIAG-2009-0027R04 Editorial
	29 Jul 2009	9	OMA-DM-DIAG-2009-0031R04
	31 Aug 2009	9	OMA-DM-DIAG-2009-0041R10 OMA-DM-DIAG-2009-0046R01 OMA-DM-DIAG-2009-0047
	16 Sep 2009	All	OMA-DM-DIAG-2009-0038R03 OMA-DM-DIAG-2009-0052R01 OMA-DM-DIAG-2009-0053 Clerical changes
	02 Nov 2009	All	OMA-DM-Diag-2009-0034R08 OMA-DM-Diag-2009-0043R01 OMA-DM-Diag-2009-0044R03 OMA-DM-Diag-2009-0048R02 OMA-DM-Diag-2009-0049R04 OMA-DM-Diag-2009-0062 OMA-DM-Diag-2009-0064 OMA-DM-Diag-2009-0065R01

Document Identifier	Date	Sections	Description
	03 Nov 2009	All	OMA-DM-Diag-2009-0068 Implemented guidelines from OMA-DM-Diag-2009-0066
	04 Nov 2009	All	Missing table numbers restored in TOC General editorial clean-up by DSO
	10 Dec 2009	All	OMA-DM-Diag-2009-0069R02 OMA-DM-Diag-2009-0070 OMA-DM-Diag-2009-0072 OMA-DM-Diag-2009-0079R01 OMA-DM-Diag-2009-0084R05 OMA-DM-Diag-2009-0085 Many issues from closure review as documented in LA Minutes (OMA-DM-2009-0144R03, section 12) Action item DM-2009-A061 Action item DM-2009-A063 Clerical changes
	14 Dec 2009	All	OMA-DM-Diag-2009-0080 Fixed numbering of figures
	15 Dec 2009	2, 3, 5.4.4, App C	Sorting of definitions in alphabetical order, restoration of figure 5, formatting of headers in App C
	10 Jan 2010	7.1, 8.5	OMA-DM-Diag-2009-0088
	14 Jan 2010	7.1.1, 8.3.1	OMA-DM-Diag-2010-0011R01 OMA-DM-Diag-2010-0012
	17 Jan 2010	5.3.1 5.5.4 All	OMA-DM-Diag-2009-0086 Use normative language (MUST) Clerical
	20 Jan 2010	6.1 5 All	OMA-DM-Diag-2009-0035R05 OMA-DM-Diag-2010-0019R01 Clerical
	04 Feb 2010	All	Applied: OMA-DM-Diag-2010-0036R01-CR_RF_Metrics_SCR OMA-DM-Diag-2010-0026-CR_BatteryStatus_bug_fix OMA-DM-Diag-2010-0025-CR_UESettingChangeWithExec OMA-DM-Diag-2010-0021R01-CR_Remaining_SCR OMA-DM-Diag-2010-0027R02-CR_NFC_Function_bugfix OMA-DM-Diag-2010-0009R02-CR_Restart_SCR OMA-DM-Diag-2010-0007R02-CR_Phonebook_SCR OMA-DM-Diag-2010-0004R01-CR_MMSUsage_SCR OMA-DM-Diag-2010-0001R03-CR_UESetting_SCR OMA-DM-Diag-2009-0089R02-CR_SCR
	08 Feb 2010	All	Re-numbering of tables
	25 Mar 2010	All	Applied: OMA-DM-Diag-2010-0041R01-CR_DiagMon_DataSessionExtension, OMA-DM-Diag-2010-0030-CR_RF_Fixes
	26 Mar 2010	All	Major cleanup – inserted missing nodes, used the Alnas tool to create diagrams, etc.

Document Identifier	Date	Sections	Description
	14 Apr 2010	All	Applied: OMA-DM-Diag-2010-0060-CR_Major_Cleanup OMA-DM-Diag-2010-0061R01-CR_SCR_Update OMA-DM-Diag-2010-0062R01-CR_TS_Funcs_More_Cleanup OMA-DM-Diag-2010-0063R02-CR_Consistent_Bearer_Table OMA-DM-Diag-2010-0067R01-CR_DiagMon_Functions_TS_Dependency_Listing OMA-DM-Diag-2010-0070R01-CR_Panic_Log_updates OMA-DM-Diag-2010-0071-CR_NFC_Function_edits OMA-DM-Diag-2010-0072R01-CR_Battery_Update OMA-DM-Diag-2010-0073R01-CR_GSM_Update OMA-DM-Diag-2010-0074R01-CR_UMTS_Update OMA-DM-Diag-2010-0075R01-CR_LTE_Update OMA-DM-Diag-2010-0076-CR_Trap_Log_Function_Cleanup OMA-DM-Diag-2010-0077R01-CR_restart_update OMA-DM-Diag-2010-0078R01-CR_SmsOptions_Update OMA-DM-Diag-2010-0079R01-CR_MgmtReject_Update OMA-DM-Diag-2010-0080R01-CR_UESetting_Update OMA-DM-Diag-2010-0081R01-CR_SmsUsage_Update OMA-DM-Diag-2010-0082-CR_Memory_Update OMA-DM-Diag-2010-0083-CR_MMS_Update OMA-DM-Diag-2010-0084-CR_Browsing_Usage_Cleanup OMA-DM-Diag-2010-0085-CR_Phonebook_Update OMA-DM-Diag-2010-0086-CR_Data_Call_Data_Session_Cleanup
	18 Apr 2010	All	Editorial clean up of formatting by DSO (fonts, re-numbering of tables, restoration of headers and footers)
	27 Apr 2010	All	Applied: OMA-DM-Diag-2010-0059-CR_clarifying_support_of_DiagMon_functions Update of OMA-TS-DiagMon-Functions-V1_0-20100414-D history box (added two CRs applied) Deletion of empty box in 6.1.2.3
	02 May 2010	All	Fixed description of many standard nodes to use the agreed language ("See DiagMon TS...") Minor clerical changes
	17 May 2010	7.1.1.3, App A 2 and C	Applied: OMA-DM-Diag-2010-0088-CR_Data_Call_Data_Session_Tree_Occurrence_Fix Editorial clean-up of Appendix A 2 and C
	01 Jun 2010	All	Applied: OMA-DM-Diag-2010-0087R01-CR_TS_Funcs_YA_Cleanup OMA-DM-Diag-2010-0091-CR_MMS_Fix
Candidate version OMA-TS-DiagMon_Functions-V1_0	22 Jun 2010	N/A	Status changed to Candidate by TP TP ref# OMA-TP-2010-0240- INP_DiagMon_V1.1_ERP_and_ETR_for_Candidate_approval
Draft versions OMA-TS-DiagMon_Functions-V1_0	12 Aug 2010	9	Applied: OMA-DM-Diag-2010-0093R02-CR_UESettingMOFix
	09 Sep 2010	5.3.1.4	Applied: OMA-DM-Diag-2010-0109-CR_Trap_Log_Node_DDF_DIAG_fix
	01 Oct 2010	5.5.3	Applied: OMA-DM-Diag-2010-0112-CR_Restart_Bug_Fixes
	27 Oct 2010	5.1.1.3, 8.1.1.3, 9.1.1.3	Applied: OMA-DM-Diag-2010-0116-CR_DiagMon_Functions_Cleanup
	28 Oct 2010	5.1.1.3	Sorted rows of node BatteryStatus values to start from zero
Candidate version OMA-TS-DiagMon_Functions-V1_0	29 Nov 2010	N/A	Status changed to Candidate by TP TP ref# OMA-TP-2010-0510- INP_DiagMon_V1.1_ERP_for_notification



## Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

### B.1 SCR for DiagMon Client (BatteryInfo)

Item	Function	Reference	Requirement
DIAG-BI-C-001-O	Device contains a battery	Section 5.1.1	DIAG-BI-C-002 AND DIAG-BI-C-003
DIAG-BI-C-002-O	Support for the BatteryInfo MO	Section 5.1.1	
DIAG-BI-C-003-O	Allows retrieval of DiagMonData nodes	Section <b>Error!</b> <b>Reference source not found.</b> 1.1	

### B.2 SCR for DiagMon Server (BatteryInfo)

Item	Function	Reference	Requirement
DIAG-BI-S-001-M	Support for the BatteryInfo MO	Section 5.1.1	
DIAG-BI-S-002-M	Retrieves the DiagMonData nodes using Get command	Section 5.1.1	

### B.3 SCR for DiagMon Client (Memory)

Item	Function	Reference	Requirement
DIAG-MEM-C-001-M	Support for the Memory MO	Section 5.2	
DIAG-MEM-C-002-M	Support for all mandatory nodes	Section 5.2	

### B.4 SCR for DiagMon Server (Memory)

Item	Function	Reference	Requirement
DIAG-MEM-S-001-M	Support for the Memory MO	Section 5.2	
DIAG-MEM-S-002-M	Retrieve the DiagMonData nodes using Get command	Section 5.2	

### B.5 SCR for DiagMon Client (trapeventlogging)

Item	Function	Reference	Requirement
DIAG-TEL-C-001-O	Device exposes trap event functionality to DiagMon Client	Section 5.3.1	DIAG-TEL-C-002-O AND DIAG-TEL-C-003-O AND DIAG-TEL-C-004-O AND (DIAG-TEL-C-005-O OR DIAG-TEL-C-006-O) AND DIAG-TEL-C-007-O
DIAG-TEL-C-002-O	Support for the trapeventlogging MO	Section 5.3.1	
DIAG-TEL-C-003-O	Support for all mandatory nodes	Section 5.3.1	
DIAG-TEL-C-004-O	Allow setting of DiagMonConfig data	Section 5.3.1	

Item	Function	Reference	Requirement
DIAG-TEL-C-005-O	Function is executed asynchronously and result is returned using Generic Alert	Section 5.3.1	
DIAG-TEL-C-006-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 5.3.1	
DIAG-TEL-C-007-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Params nodes	Section 5.3.1	

## B.6 SCR for DiagMon Server (trapeventlogging)

Item	Function	Reference	Requirement
DIAG-TEL-S-001-M	Support for the trapeventlogging MO	Section 5.3.1	
DIAG-TEL-S-002-M	Invoke trapeventlogging function via Start Primitive	Section 5.3.1	
DIAG-TEL-S-003-M	Retrieve the DiagMonData nodes	Section 5.3.1	
DIAG-TEL-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 5.3.1	
DIAG-TEL-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 5.3.1	
DIAG-TEL-S-006-M	Stop function via Stop Primitive	Section 5.3.1	
DIAG-TEL-S-007-M	Retrieve the Status data	Section 5.3.1	
DIAG-TEL-S-008-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Params nodes	Section 5.3.1	

## B.7 SCR for DiagMon Client (paniclog)

Item	Function	Reference	Requirement
DIAG-PL-C-001-O	Device exposes panic log functionality to DiagMon Client	Section 5.4	DIAG-PL-C-002-O AND DIAG-PL-C-003-O AND DIAG-PL-C-004-O AND (DIAG-PL-C-004-O OR DIAG-PL-C-005-O)
DIAG-PL-C-002-O	Support for the paniclog MO	Section 5.4	
DIAG-PL-C-003-O	Support for all mandatory nodes	Section 5.4	

Item	Function	Reference	Requirement
DIAG-PL-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 5.4	
DIAG-PL-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 5.4	

## B.8 SCR for DiagMon Server (paniclog)

Item	Function	Reference	Requirement
DIAG-PL-S-001-M	Support for the paniclog MO	Section 5.4	
DIAG-PL-S-002-M	Invoke function via Start Primitive	Section 5.4	
DIAG-PL-S-003-M	Retrieve the DiagMonData nodes	Section 5.4	
DIAG-PL-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 5.4	
DIAG-PL-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 5.4	
DIAG-PL-S-006-M	Stop function via Stop Primitive	Section 5.4	
DIAG-PL-S-007-M	Retrieve the Status data	Section 5.4	

## B.9 SCR for DiagMon Client (Restart)

Item	Function	Reference	Requirement
DIAG-RST-C-001-O	Device supports restart functionality	Section 5.5	DIAG-RST-C-002-O AND DIAG-RST-C-003-O AND DIAG-RST-C-004-O AND DIAG-RST-C-005-O AND (DIAG-RST-C-006-O OR DIAG-RST-C-007-O) AND DIAG-RST-C-008-O
DIAG-RST-C-002-O	Support for the restart MO	Section 5.5	
DIAG-RST-C-003-O	Allow retrieval of DiagMonConfig data	Section 5.5	
DIAG-RST-C-004-O	Allow setting of DiagMonConfig data	Section 5.5	
DIAG-RST-C-005-O	Support of invocation of function by the server via Start Primitive	Section 5.5	
DIAG-RST-C-006-O	Asynchronous reporting of result using Generic Alert	Section 5.5	
DIAG-RST-C-007-O	Synchronous reporting of result by processing a	Section 5.5	

Item	Function	Reference	Requirement
	subsequent Get command		
DIAG-RST-C-008-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Params nodes	Section 5.5	

## B.10 SCR for DiagMon Server (Restart)

Item	Function	Reference	Requirement
DIAG-RST-S-001-M	Support for the restart MO	Section 5.5	
DIAG-RST-S-002-M	Support retrieval of DiagMonConfig data	Section 5.5	
DIAG-RST-S-003-M	Support setting of DiagMonConfig data	Section 5.5	
DIAG-RST-S-004-M	Invoke function via Start Primitive	Section 5.5	
DIAG-RST-S-005-M	Asynchronous reporting of result which is retrieved by receiving Generic Alert	Section 5.5	
DIAG-RST-S-006-M	Synchronous reporting of result which is retrieved by Get command	Section 5.5	
DIAG-RST-S-007-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Params nodes	Section 5.5	

## B.11 SCR for DiagMon Client (RFParms\_3GPP\_GSM)

Item	Function	Reference	Requirement
DIAG-GSM-C-001-O	Device exposes GSM RF functionality to DiagMon Client	Section 6.1.1	DIAG-GSM-C-002-O AND DIAG-GSM-C-003-O AND DIAG-GSM-C-004-O AND (DIAG-GSM-C-005-O OR DIAG-GSM-C-006-O) AND DIAG-GSM-C-007-O
DIAG-GSM-C-002-O	Support for the RFParms_3GPP_GSM MO	Section 6.1.1	
DIAG-GSM-C-003-O	Support for all mandatory nodes	Section 6.1.1	
DIAG-GSM-C-004-O	Allow setting of DiagMonConfig data	Section 6.1.1	
DIAG-GSM-C-005-O	Function is executed asynchronously and result is returned using Generic Alert	Section 6.1.1	
DIAG-GSM-C-006-O	Function is executed	Section 6.1.1	

Item	Function	Reference	Requirement
	synchronously and result is returned by receiving a subsequent Get command		
DIAG-GSM-C-007-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Params nodes	Section 6.1.1	

## B.12 SCR for DiagMon Server (RFParms\_3GPP\_GSM)

Item	Function	Reference	Requirement
DIAG-GSM-S-001-M	Support for the RFParms_3GPP_GSM MO	Section 6.1.1	
DIAG-GSM-S-002-M	Invoke function via Start Primitive	Section 6.1.1	
DIAG-GSM-S-003-M	Retrieve the DiagMonData nodes	Section 6.1.1	
DIAG-GSM-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 6.1.1	
DIAG-GSM-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 6.1.1	
DIAG-GSM-S-006-M	Stop function via Stop Primitive	Section 6.1.1	
DIAG-GSM-S-007-M	Retrieve the Status data	Section 6.1.1	
DIAG-GSM-S-008-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Params nodes	Section 6.1.1	

## B.13 SCR for DiagMon Client (RFParms\_3GPP\_UMTS)

Item	Function	Reference	Requirement
DIAG-UMTS-C-001-O	Device exposes UMTS RF functionality to DiagMon Client	Section 6.1.2	DIAG-UMTS-C-002-O AND DIAG-UMTS-C-003-O AND DIAG-UMTS-C-004-O AND (DIAG-C-UMTS-005-O OR DIAG-C-UMTS-006-O) AND DIAG-UMTS-C-007-O
DIAG-UMTS-C-002-O	Support for the RFParms_3GPP_UMTS MO	Section 6.1.2	
DIAG-UMTS-C-003-O	Support for all mandatory nodes	Section 6.1.2	
DIAG-UMTS-C-004-O	Allow setting of DiagMonConfig data	Section 6.1.2	
DIAG-UMTS-C-005-O	Function is executed	Section 6.1.2	

Item	Function	Reference	Requirement
	asynchronously and result is returned using Generic Alert		
DIAG-UMTS-C-006-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 6.1.2	
DIAG-UMTS-C-007-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Params nodes	Section 6.1.2	

### B.14 SCR for DiagMon Server (RFParms\_3GPP\_UMTS)

Item	Function	Reference	Requirement
DIAG-UMTS-S-001-M	Support for the RFParms_3GPP_UMTS MO	Section 6.1.2	
DIAG-UMTS-S-002-M	Invoke function via Start Primitive	Section 6.1.2	
DIAG-UMTS-S-003-M	Retrieve the DiagMonData nodes	Section 6.1.2	
DIAG-UMTS-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 6.1.2	
DIAG-UMTS-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 6.1.2	
DIAG-UMTS-S-006-M	Stop function via Stop Primitive	Section 6.1.2	
DIAG-UMTS-S-007-M	Retrieve the Status data	Section 6.1.2	
DIAG-UMTS-S-008-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Params nodes	Section 6.1.2	

### B.15 SCR for DiagMon Client (RFParms\_3GPP\_LTE)

Item	Function	Reference	Requirement
DIAG-LTE-C-001-O	Device exposes LTE RF functionality to DiagMon Client	Section 6.1.3	DIAG-LTE-C-002-O AND DIAG-LTE-C-003-O AND DIAG-LTE-C-004-O AND (DIAG-C-LTE-005-O OR DIAG-C-LTE-006-O) AND DIAG-LTE-C-007-O
DIAG-LTE-C-002-O	Support for the RFParms_3GPP_LTE MO	Section 6.1.3	
DIAG-LTE-C-003-O	Support for all mandatory nodes	Section 6.1.3	

Item	Function	Reference	Requirement
DIAG-LTE-C-004-O	Allow setting of DiagMonConfig data	Section 6.1.3	
DIAG-LTE-C-005-O	Function is executed asynchronously and result is returned using Generic Alert	Section 6.1.3	
DIAG-LTE-C-006-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 6.1.3	
DIAG-LTE-C-007-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Params nodes	Section 6.1.3	

## B.16 SCR for DiagMon Server (RFParms\_3GPP\_LTE)

Item	Function	Reference	Requirement
DIAG-LTE-S-001-M	Support for the RFParms_3GPP_LTE MO	Section 6.1.3	
DIAG-LTE-S-002-M	Invoke RFParms_3GPP_LTE function via Start Primitive	Section 6.1.3	
DIAG-LTE-S-003-M	Retrieve the DiagMonData nodes	Section 6.1.3	
DIAG-LTE-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 6.1.3	
DIAG-LTE-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 6.1.3	
DIAG-LTE-S-006-M	Stop function via Stop Primitive	Section 6.1.3	
DIAG-LTE-S-007-M	Retrieve the Status data	Section 6.1.3	
DIAG-LTE-S-008-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Params nodes	Section 6.1.3	

## B.17 SCR for DiagMon Client (mobility\_management\_rejection)

Item	Function	Reference	Requirement
DIAG-MMR-C-001-O	Device operates on a 3G network	Section 6.2.1	DIAG-MMR-C-002-O AND DIAG-MMR-C-003-O AND (DIAG-MMR-C-004-O OR DIAG-MMR-C-005-O)
DIAG-MMR-C-002-O	Support for the	Section 6.2.1	

Item	Function	Reference	Requirement
	mobility_management_rejection MO		
DIAG-MMR-C-003-O	Support for all mandatory nodes	Section 6.2.1	
DIAG-MMR-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 6.2.1	
DIAG-MMR-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 6.2.1	

## B.18 SCR for DiagMon Server (mobility\_management\_rejection)

Item	Function	Reference	Requirement
DIAG-MMR-S-001-M	Support for the mobility_management_rejection MO	Section 6.2.1	
DIAG-MMR-S-002-M	Invoke function via Start Primitive	Section 6.2.1	
DIAG-MMR-S-003-M	Retrieve the DiagMonData nodes	Section 6.2.1	
DIAG-MMR-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 6.2.1	
DIAG-MMR-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 6.2.1	
DIAG-MMR-S-006-M	Stop function via Stop Primitive	Section 6.2.1	
DIAG-MMR-S-007-M	Retrieve the Status data	Section 6.2.1	

## B.19 SCR for DiagMon Client (DataCallAndDataSession)

Item	Function	Reference	Requirement
DIAG-DCDS-C-001-O	Device supports data calls	Section 7.1.1	DIAG-DCDS-C-002-O AND DIAG-DCDS-C-003-O AND (DIAG-DCDS-C-004-O OR DIAG-DCDS-C-005-O) AND DIAG-DCDS-C-006-O AND DIAG-DCDS-C-007-O
DIAG-DCDS-C-002-O	Support for the DataCallAndDataSession MO	Section 7.1.1	
DIAG-DCDS-C-003-O	Support for all mandatory nodes	Section 7.1.1	
DIAG-DCDS-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 7.1.1	



Item	Function	Reference	Requirement
DIAG-DCDS-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 7.1.1	
DIAG-DCDS-C-006-O	Allow setting of DiagMonConfig data	Section 7.1.1	
DIAG-DCDS-C-007-O	Function behaviour and collected data are affected by setting of DiagMonConfig/Config Parms nodes	Section 7.1.1	

## B.20 SCR for DiagMon Server (DataCallAndDataSession)

Item	Function	Reference	Requirement
DIAG-DCDS-S-001-M	Support for the DataCallAndDataSession MO	Section 7.1.1	
DIAG-DCDS-S-002-M	Invoke DataCallAndDataSession function via Start Primitive	Section 7.1.1	
DIAG-DCDS-S-003-M	Retrieve the DiagMonData nodes	Section 7.1.1	
DIAG-DCDS-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 7.1.1	
DIAG-DCDS-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 7.1.1	
DIAG-DCDS-S-006-M	Stop function via Stop Primitive	Section 7.1.1	
DIAG-DCDS-S-007-M	Retrieve the Status data	Section 7.1.1	
DIAG-DCDS-S-008-M	Modify behaviour of function by modifying values of DiagMonConfig/Config Parms nodes	Section 7.1.1	

## B.21 SCR for DiagMon Client (SmsOptions)

Item	Function	Reference	Requirement
DIAG-SO-C-001-O	Device exposes SMS option functionality to DiagMon Client	Section 8.1.1	DIAG-SO-C-002-O AND DIAG-SO-C-003-O
DIAG-SO-C-002-O	Support for the SmsOptions MO	Section 8.1.1	
DIAG-SO-C-003-O	Allows retrieval of DiagMonData nodes	Section 8.1.1	

## B.22 SCR for DiagMon Server (SmsOptions)

Item	Function	Reference	Requirement
DIAG-SO-S-001-M	Support for the SmsOptions MO	Section 8.1.1	
DIAG-SO-S-002-M	Retrieve the DiagMonData nodes using Get command	Section 8.1.1	

## B.23 SCR for DiagMon Client (SmsUsage)

Item	Function	Reference	Requirement
DIAG-SU-C-001-O	Device exposes SMS usage functionality to DiagMon Client	Section 8.1.2	DIAG-SU-C-002-O AND DIAG-SU-C-003-O AND (DIAG-SU-C-004-O or DIAG-SU-C-005-O)
DIAG-SU-C-002-O	Support for the SmsUsage MO	Section 8.1.2	
DIAG-SU-C-003-O	Allows retrieval of DiagMonData nodes	Section 8.1.2	
DIAG-SU-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 8.1.2	
DIAG-SU-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 8.1.2	

## B.24 SCR for DiagMon Server (SmsUsage)

Item	Function	Reference	Requirement
DIAG-SU-S-001-M	Support for the SmsUsage MO	Section 8.1.2.2	
DIAG-SU-S-002-M	Invoke SmsUsage function via Start Primitive	Section 8.1.2.4	
DIAG-SU-S-003-M	Retrieve the DiagMonData nodes	Section 8.1.2.4	
DIAG-SU-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 8.1.2.4	
DIAG-SU-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 8.1.2.4	
DIAG-SU-S-006-M	Stop function via Stop Primitive	Section 8.1.2.4	
DIAG-SU-S-007-M	Retrieve the Status data	Section 8.1.2.4	

## B.25 SCR for DiagMon Client (MMSUsage)

Item	Function	Reference	Requirement
------	----------	-----------	-------------

Item	Function	Reference	Requirement
DIAG-MU-C-001-O	Device exposes MMS usage functionality to DiagMon Client	Section 8.2.1	DIAG-MU-C-002-O AND DIAG-MU-C-003-O AND (DIAG-MU-C-004-O OR DIAG-MU-C-005-O)
DIAG-MU-C-002-O	Support for the MMSUsage MO	Section 8.2.1	
DIAG-MU-C-003-O	Allows retrieval DiagMonData nodes	Section 8.2.1	
DIAG-MU-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 8.2.1	
DIAG-MU-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 8.2.1	

## B.26 SCR for DiagMon Server (MMSUsage)

Item	Function	Reference	Requirement
DIAG-MU-S-001-M	Support for the MMSUsage MO	Section 8.2.1	
DIAG-MU-S-002-M	Invoke MMSUsage function via Start Primitive	Section 8.2.1	
DIAG- MU-S-003-M	Retrieve the DiagMonData nodes	Section 8.2.1	
DIAG- MU-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 8.2.1	
DIAG- MU-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 8.2.1	
DIAG- MU-S-006-M	Stop function via Stop Primitive	Section 8.2.1	
DIAG- MU-S-007-M	Retrieve the Status data	Section 8.2.1	

## B.27 SCR for DiagMon Client (BrowsingUsage)

Item	Function	Reference	Requirement
DIAG-BU-C-001-O	Support for the BrowsingUsage MO	Section 8.3.1	DIAG-BU-C-002-O AND (DIAG-BU-C-003-O OR DIAG-BU-C-004-O)
DIAG-BU-C-002-O	Support for all mandatory nodes	Section 8.3.1	
DIAG-BU-C-003-O	Function is executed asynchronously and result is returned using Generic Alert	Section 8.3.1	
DIAG-BU-C-004-O	Function is executed synchronously and result is returned by receiving	Section 8.3.1	

Item	Function	Reference	Requirement
	a subsequent Get command		

## B.28 SCR for DiagMon Server (BrowsingUsage)

Item	Function	Reference	Requirement
DIAG-BU-S-001-M	Support for the BrowsingUsage MO	Section 8.3.1	
DIAG-BU-S-002-M	Invoke BrowsingUsage function via Start Primitive	Section 8.3.1	
DIAG-BU-S-003-M	Retrieve the DiagMonData nodes	Section 8.3.1	
DIAG-BU-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 8.3.1	
DIAG-BU-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 8.3.1	
DIAG-BU-S-006-M	Stop function via Stop Primitive	Section 8.3.1	
DIAG-BU-S-007-M	Retrieve the Status data	Section 8.3.1	

## B.29 SCR for DiagMon Client (AppMonLog)

Item	Function	Reference	Requirement
DIAG-AML-C-001-O	Device exposes application monitoring functionality to DiagMon Client	Section 8.4	DIAG-AML-C-002-O AND DIAG-AML-C-003-O AND (DIAG-AML-C-004-O OR DIAG-AML-C-005-O)
DIAG-AML-C-002-O	Support for the AppMonLog MO	Section 8.4	
DIAG-AML-C-003-O	Allows retrieval of DiagMonData nodes	Section 8.4	
DIAG-AML-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 8.4	
DIAG-AML-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 8.4	

## B.30 SCR for DiagMon Server (AppMonLog)

Item	Function	Reference	Requirement
DIAG-AML-S-001-M	Support for the AppMonLog MO	Section 8.4	
DIAG-AML-S-002-M	Invoke AppMonLog function via Start	Section 8.4	

Item	Function	Reference	Requirement
	Primitive		
DIAG-AML-S-003-M	Retrieve the DiagMonData nodes	Section 8.4	
DIAG-AML-S-004-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 8.4	
DIAG-AML-S-005-M	Function is executed synchronously and result is retrieved by Get command	Section 8.4	
DIAG-AML-S-006-M	Stop function via Stop Primitive	Section 8.4	
DIAG-AML-S-007-M	Retrieve the Status data	Section 8.4	

### B.31 SCR for DiagMon Client (NFC)

Item	Function	Reference	Requirement
DIAG-NFC-C-001-O	Device exposes NFC functionality to DiagMon Client	Section 8.5	DIAG-NFC-C-002-O AND DIAG-NFC-C-003-O AND (DIAG-NFC-C-004-O OR DIAG-NFC-C-005-O)
DIAG-NFC-C-002-O	Support for the NFC MO	Section 8.5	
DIAG-NFC-C-003-O	Support for all mandatory nodes	Section 8.5	
DIAG-NFC-C-004-O	Function is executed asynchronously and result is returned using Generic Alert	Section 8.5	
DIAG-NFC-C-005-O	Function is executed synchronously and result is returned by receiving a subsequent Get command	Section 8.5	

### B.32 SCR for DiagMon Server (NFC)

Item	Function	Reference	Requirement
DIAG-NFC-S-001-M	Support for the NFC MO	Section 8.5	
DIAG-NFC-S-002-M	Retrieve the NFC/DiagMonData nodes	Section 8.5	
DIAG-NFC-S-003-M	Function is executed asynchronously and result is retrieved by receiving Generic Alert	Section 8.5	
DIAG-NFC-S-004-M	Function is executed synchronously and result is retrieved by Get command	Section 8.5	
DIAG-NFC-S-005-M	Stop function via Stop Primitive	Section 8.5	
DIAG-NFC-S-006-M	Retrieve the Status data	Section 8.5	

### B.33 SCR for DiagMon Client (UE Setting)

Item	Function	Reference	Requirement
DIAG-UES-C-001-O	Device exposes user equipment setting functionality to DiagMon Client	Section 9.1.1	DIAG-UES-C-002-O AND DIAG-UES-C-003-O AND DIAG-UES-C-004-O AND (DIAG-UES-C-005-O OR DIAG-UES-C-006-O)
DIAG-UES-C-002-O	Support for the UESetting MO	Section 9.1.1	
DIAG-UES-C-003-O	Allows Retrieval of DiagMonConfig/UESetting	Section 9.1.1	
DIAG-UES-C-004-O	Allows Setting of DiagMonConfig/ConfigParms/Category data	Section 9.1.1	
DIAG-UES-C-005-O	Allows modification on UE Setting with ReplacementSetting and subsequent execution of Operations/Modify	Section 9.1.1	
DIAG-UES-C-006-O	Allows Retrieval of DiagMonConfig/ConfigParms/RefreshInterval data	Section 9.1.1	

### B.34 SCR for DiagMon Server (UE Setting)

Item	Function	Reference	Requirement
DIAG-UES-S-001-M	Support for the UESetting MO	Section 9.1.1	
DIAG-UES-S-002-M	Retrieve the DiagMonConfig/UESetting using Get command	Section 9.1.	
DIAG-UES-S-003-M	Set the DiagMonConfig/ConfigParms/Category using Replace command	Section 9.1.4	
DIAG-UES-S-004-M	Set the DiagMonConfig/ConfigParms/ReplacementSetting using Replace command	Section 9.1.1	
DIAG-UES-S-005-M	Retrieve the DiagMonConfig/ConfigParms/ RefreshInterval data	Section 9.1.1	
DIAG-UES-S-006-M	Modify the UE Setting via the Modify primitive	Section 9.1.1	

### B.35 SCR for DiagMon Client (Phonebook)

Item	Function	Reference	Requirement
DIAG-PB-C-001-O	Device exposes phonebook functionality to DiagMon Client	Section 9.2	DIAG-PB-C-002-O AND DIAG-PB-C-003-O

Item	Function	Reference	Requirement
DIAG-PB-C-002-O	Support for the Phonebook MO	Section 9.2	
DIAG-PB-C-003-O	Allows retrieval of DiagMonData nodes by a Get command	Section 9.2	

### B.36 SCR for DiagMon Server (Phonebook)

Item	Function	Reference	Requirement
DIAG-PB-S-001-M	Support for the Phonebook MO	Section 9.2	
DIAG-PB-S-002-M	Retrieve the DiagMonData nodes using a Get command	Section 9.2	

## Appendix C. NFC XML Schemas

This section includes the XML schemas for the NFC DiagMon function and the data description

### C.1 NfcRFSignal

#### NfcRFSignal XML Schema:

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
  <xs:element name='NfcRFSignal'>
    <xs:complexType mixed='true'>
      <xs:sequence>
        <xs:element name='Date' type='xs:dateTime'/>
        <xs:element name='TechType' type='xs:string'/>
        <xs:element name='TechSubType' type='xs:string'/>
        <xs:element name='IndMod' type='xs:string'/>
        <xs:element name='CommRes' type='xs:string'/>
        <xs:element name='BER' type='xs:string'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

#### NFC DiagMon Data Description:

Data	Description
Date:	Date and Time of failure
TechType:	Technology Type used
TechSubType:	Technology SubType used
IndMod:	Index Modulation
CommRes:	communication result (+ error)
BER:	average bit error rate

### C.2 NfcModeError

#### NfcModeError XML Schema:

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
  <xs:element name='NfcModeError'>
    <xs:complexType mixed='true'>
      <xs:sequence>
        <xs:element name='NM' type='xs:integer'/>
        <xs:element name='Date' type='xs:dateTime'/>
        <xs:element name='ErrorCode' type='xs:string'/>
        <xs:element name='LSNM' type='xs:string'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



```

    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

**NFC DiagMon Data Description:**

Data	Description
Date:	Date and Time of failure
NM:	NFC Mode (Int Value)
LSNM:	Last Successful NFC Mode (Int Value)
ErrorCode:	Error Code

### C.3 NfcTypeError

**NfcTypeError XML Schema:**

```

<?xml version='1.0' encoding='utf-8'?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema' >
  <xs:element name='NfcTypeError' >
    <xs:complexType mixed='true' >
      <xs:sequence >
        <xs:element name='NTE' type='xs:integer' />
        <xs:element name='Date' type='xs:dateTime' />
        <xs:element name='ErrorCode' type='xs:string' />
      </xs:sequence >
    </xs:complexType >
  </xs:element >
</xs:schema >

```

**NFC DiagMon Data Description:**

Data	Description
Date:	Date and Time of failure
NTE:	NFC Type Error
ErrorCode:	Error Code