

2006-11-28

IEEE 802.16i-06/001r5, November 2006

This a NetMan Task Group P802.16i Baseline Document.

## **~~Draft Amendment to IEEE Standard~~ for Local and Metropolitan Area Networks - Part 16: Management Information Base Extensions**

Sponsor

**LAN MAN Standards Committee**  
of the  
**IEEE Computer Society**

and the

**IEEE Microwave Theory and Techniques Society**

**Abstract:** This document provides updates to IEEE Std 802.16's MIB for the MAC, PHY and associated management procedures in order to accommodate recent extensions to the standard. The project will use protocol-neutral methodologies for network management to develop resource models and related solution sets for the management of devices in a multi-vendor 802.16 network.

**Keywords:** fixed broadband wireless access network, mobile broadband wireless access network, metropolitan area network, microwave, millimeter wave, WirelessMAN™ standards, WMAN MIB



~~Copyright © 2006 by the Institute of Electrical and Electronics Engineers, Inc.~~  
Three Park Avenue  
New York, NY 10016-5997, USA  
All rights reserved.

This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. USE AT YOUR OWN RISK! Because this is an unapproved draft, this document must not be utilized for any conformance/compliance purposes. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only. Prior to submitting this document to another standards development organization for standardization activities, permission must first be obtained from the Manager, Standards Licensing and Contracts, IEEE Standards Activities Department. Other entities seeking permission to reproduce this document, in whole or in part, must obtain permission from the Manager, Standards Licensing and Contracts, IEEE Standard Activities Department.

IEEE Standards Activities Department  
Standards Licensing and Contracts  
445 Hoes Lane, P.O. Box 1331  
Piscataway, NJ 08855-1331, USA

**IEEE Standards** documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS.**”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board  
445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

## Introduction

(This introduction is not part of IEEE Draft P802.16i, Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks - Part 16: Management Information Base Extensions.)

## Participants

This document was developed by the IEEE 802.16 Working Group on Broadband Wireless Access, which develops the WirelessMAN™ Standard for Wireless Metropolitan Area Networks.

IEEE 802.16 Working Group Officers

**Roger B. Marks**, *Chair*

**Jose Puthenkulam**, *Vice Chair*

**Peiyong Zhu**, *Secretary*

Primary development was carried out by the Working Group's Network Management Task Group Officers.

**Phillip Barber**, *Chair*

**Changhoi Koo**, *Vice Chair*

**Itzik Kitroser**, *Vice Chair*

**Joey Chou**, *802.16f Chief Technical Editor*

The following members of the IEEE 802.16 Working Group on Broadband Wireless Access participated in the Working Group Letter Ballot in which the draft of this standard was prepared and finalized for IEEE Ballot:

*[to be determined]*

The following participated as non-members in the Working Group Letter Ballot:

*[to be determined]*

The following members of the IEEE Balloting Committee voted on this standard, whether voting for approval or disapproval, or abstaining.

*[to be determined]*

The following persons, who were not members of the IEEE Balloting Committee, participated (without voting) in the IEEE Sponsor Ballot in which the draft of this standard was approved:

*[to be determined]*

When the IEEE-SA Standards Board approved this standard on *[date]*, it had the following membership:

*[to be determined]*

Also included is the following nonvoting IEEE-SA Standards Board liaison:

*[to be determined]*

This draft is intended for IEEE-SA Sponsor Ballot with individuals as the ballot group members.



## Contents

1		
2		
3		
4	1. Overview.....	1
5	1.1 Scope.....	1
6	1.2 Purpose.....	1
7	1.3 Reference Models .....	1
8	1.3.1 Management Reference Models .....	2
9		
10	2. References.....	3
11	4. Abbreviations and Acronyms .....	4
12	9. Configuration.....	6
13	9.3 ASN.1 Management Information Base.....	6
14	9.3.1 SNMP.....	6
15	9.3.2 Relationship with interface MIB.....	6
16	9.3.2.1 MIB-2 integration .....	7
17	9.3.2.2 Usage of MIB-II tables .....	7
18	9.3.3 Events and traps .....	10
19	9.3.4 MIB Modules .....	10
20	9.3.4.1 WMAN-IF-MIB module.....	10
21	9.3.4.2 WMAN-DEV-MIB module .....	11
22	9.3.4.3 WMAN-IF2-MIB module.....	11
23	9.3.4.4 WMAN-IF2M-MIB module .....	11
24		
25	13. 802.16 MIB Modulesstructure for SNMP .....	13
26	13.1 Structure of MIB modules .....	13
27	13.1.1 wmanIfMib .....	13
28	13.1.2 wmanDevMib .....	13
29	13.1.3 wmanIf2Mib .....	14
30	13.1.3.1 wmanIf2BsObjects.....	14
31	13.1.3.2 wmanIf2SsObjects .....	20
32	13.1.3.3 wmanIf2CommonObjects .....	23
33	13.1.4 wmanIf2mMib .....	24
34	13.1.4.1 wmanIf2mBsObjects .....	25
35	13.2 ASN.1 Definitions of MIB Modules.....	26
36	13.2.1 wmanIfMib .....	26
37	13.2.2 wmanDevMib .....	26
38	13.2.3 wmanIf2Mib .....	27
39	13.2.4 wmanIf2mMib .....	287
40		
41	Annex E. ....	310
42	Annex F. Proposal for Adding Mobility Handover and Paging group MIBs.....	311
43	1. Introduction.....	311
44	2. Proposed Text Introduction .....	311
45	2.1 wmanIfBsObjects.....	311
46	2.1.1 wmanIfBsMobility.....	311
47	2.1.1.1 wmanIfBsHandoverConfiguration.....	311
48	2.1.1.2 wmanIfBsPagingGroupTable .....	311
49	3. ASN.1 Definitions of 802.16 MIB for SNMP .....	312
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



## List of Figures

1	
2	
3	
4	Figure 1—Mobile BWA Network Management Reference Model..... 2
5	Figure 2—Mobile BWA Network Management Architecture - Context A ..... 2
6	Figure 3—Mobile BWA Network Management Architecture - Context B..... 2
7	Figure 4—SS / MS Network Entry ..... 9
8	Figure 5—wmanIf2Mib structure ..... 14
9	Figure 6—wmanIf2BsPacketCs structure ..... 14
10	Figure 7—wmanIf2BsCps structure ..... 15
11	Figure 8—wmanIf2BsPkmObjects structure..... 16
12	Figure 9—wmanIf2BsNotification structure ..... 17
13	Figure 10—wmanIf2BsPhy structure ..... 18
14	Figure 11—wmanIf2SsCps structure..... 20
15	Figure 12—wmanIf2SsPkmObjects structure ..... 21
16	Figure 13—wmanIf2SsNotification structure ..... 21
17	Figure 14—wmanIf2SsPhy structure..... 22
18	Figure 15—wmanIf2CmnPacketCs structure ..... 23
19	Figure 16—wmanIf2CmnCps structure..... 23
20	Figure 17—wmanIf2CmnPkmObjects structure ..... 24
21	Figure 18—wmanIf2mMib structure..... 25
22	Figure 19—wmanIf2mBsCm structure ..... 25
23	Figure 20—wmanIf2mBsPm structure..... 26
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

# List of Tables

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

Table 1—Example of the Usage of ifTable objects for base station .....	8
Table 2—Example of the Usage of ifTable objects for subscriber station.....	8
Table 3—List of ASN.1 MIB Modules .....	10

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11 Baseline document for Draft Amendment to IEEE Standard for Local and  
12  
13 metropolitan area networks  
14  
15

# 16 17 **Part 16: Management Information Base** 18 19 **Extensions** 20 21 22 23 24 25

26  
27 NOTE-The editing instructions contained in this amendment define how to merge the material contained  
28 herein into the existing base standard IEEE Std 802.16-2004.  
29

30 The editing instructions are shown *bold italic*. Four editing instructions are used: *change*, *delete*, *insert*, and  
31 *replace*. *Change* is used to make small corrections in existing text or tables. The editing instruction specifies  
32 the location of the change and describes what is being changed by using strike through (to remove old mate-  
33 rial) and underscore (to add new material). *Delete* removes existing material. *Insert* adds new material with-  
34 out disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are  
35 given in the editing instruction. *Replace* is used to make large changes in existing text, subclauses, tables, or  
36 figures by removing existing material and replacing it with new material. Editorial notes will not be carried  
37 over into future editions because the changes will be incorporated into the base standard.  
38  
39  
40

## 41 42 **1. Overview** 43 44

### 45 46 **1.1 Scope** 47

48 This document provides mobility enhancements to the IEEE Std 802.16 Management Information Base for  
49 the medium access control layer, physical layer, and associated management procedures. It uses protocol-  
50 neutral methodologies for network management to specify resource models and related solution sets for the  
51 management of devices in a multivendor 802.16 mobile network.  
52

### 53 54 **1.2 Purpose** 55

56 This amendment provides a definition of managed objects to enhance the standards-based management of  
57 802.16 devices.  
58  
59

### 60 61 **1.3 Reference Models** 62 63 64 65

### 1.3.1 Management Reference Models

Figure 1 illustrates the Management Reference Model (see also 3GPP TS 32.101). It shows the Operation System interfacing with other systems. A number of management interfaces are identified in Figure 1, namely:

1. Between the Network Elements (NEs) and the Element Manager (EM)
2. Between the Element Manager (EM) and the Network Manager (NM)
3. Between the Network Managers and the Enterprise Systems
4. Between Network Managers (NMs)
5. Between Enterprise Systems & Network Managers of different Organisations
6. Between Network Elements (NEs).

The resource model defined within this section focuses primarily on serving management interface "2" and to a lesser extent on management interface "1" from the above list.

#### Figure 1— Mobile BWA Network Management Reference Model

Figure 2 and Figure 3 identify system contexts of the Management Interface "2" in terms of its implementation, called IRP Agent, and the user of the IRP Agent, called IRP Manager (for a definition of IRP Manager and IRP Agent see 3GPP TS 32.102). An NE can be managed either

- via System Context A (IRP Agent is a standalone system) or
- Via System Context B (element management function and IRP Agent embedded within the NE).

The criterion for choosing System Context A or B to manage a particular NE is implementation dependent. An IRP Agent shall support one of the two System Contexts.

#### Figure 2— Mobile BWA Network Management Architecture— Context A

#### Figure 3— Mobile BWA Network Management Architecture— Context B

## 2. References

This standard shall be used in conjunction with the following publications. When the following specifications are superseded by an approved revision, the revision shall apply.

### *[Replace the following references]*

~~IETF RFC1902, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", January 1996~~

IETF RFC2578 "Structure of Management Information Version 2 (SMIv2) " April 1999

~~IETF RFC1903, "Textual Convention for Version 2 of the Simple Network Management Protocol (SNMPv2)", January 1996~~

IETF RFC2579 "Textual Conventions for SMIv2 " April 1999

~~IETF RFC2576, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework", March 2000~~

IETF RFC3584 "TCoexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework " August 2003

### *[Insert the following new references]*

3GPP TS 32.101, "Principles and High Level Requirements"; Release 6, [http://www.3gpp.org/ftp/specs/archive/32\\_series](http://www.3gpp.org/ftp/specs/archive/32_series)

3GPP TS 32.150, "Integration Reference Point (IRP) Concept and Definitions", Release 6, [http://www.3gpp.org/ftp/specs/archive/32\\_series](http://www.3gpp.org/ftp/specs/archive/32_series)

3GPP TS 32.151, "Integration Reference Point (IRP) Information Service (IS) Template", Release 6, [http://www.3gpp.org/ftp/specs/archive/32\\_series](http://www.3gpp.org/ftp/specs/archive/32_series)

3GPP TS 32.152, "Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) Repertoire", Release 6, [http://www.3gpp.org/ftp/specs/archive/32\\_series](http://www.3gpp.org/ftp/specs/archive/32_series)

3GPP TS 32.622, "Configuration Management (CM); Generic Network Resources Integration Reference Point (IRP); Network Resource Model (NRM)"; Release 6, [http://www.3gpp.org/ftp/specs/archive/32\\_series](http://www.3gpp.org/ftp/specs/archive/32_series)

3GPP2 S.S0028-002-C, "OAM&P for cdma2000 (3GPP2 Generic NRM IRP)" [http://www.3gpp2.org/Public\\_html/specs/index.cfm](http://www.3gpp2.org/Public_html/specs/index.cfm)

3GPP TS 32.102, "Telecommunication management; Architecture", Release 6, <http://www.3gpp.org/ftp/>

1 specs/archive/32\_series  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

#### 14 **4. Abbreviations and Acronyms**

15  
16 *[Insert a new definition in this sunclause]*  
17  
18

19  
20 ~~RDN Relative Distinguished Name~~  
21

22 ~~RP Integration Reference Point~~  
23

24 ~~IS Information Service~~  
25

26  
27 NRM Network Resource Model  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## 9. Configuration

*[Change subclause 9.3 as follows:]*

### 9.3 ASN.1 Management Information Base

The management information base for BS and SS is defined as two ASN.1 MIB modules: ~~wmanIfMib and wmanDevMib~~. The implementation of ~~both~~ ASN.1 MIB modules is mandatory for all BSs. The implementation of ~~both~~ ASN.1 MIB modules is mandatory for SSs that are managed using SNMP protocol.

The specific requirements for implementation of individual MIB modules are defined in section 9.3.3. The specific requirements for implementation of individual MIB objects in each MIB module are defined in conformance statements of the MIB modules.

~~The wmanIfMib MIB module (interface MIB) defines management objects relevant to 802.16 broadband wireless interface as defined in this standard.~~

~~The wmanDevMib MIB module (device MIB) defines management objects relevant to the device implementing 802.16 interface. The objects of this MIB module may refer explicitly to terms defined in the standard (e.g. configuration file encodings) but mainly provides the mandatory support required to implement, manage and test the equipment implementing 802.16 interface.~~

~~This document also provides an informative Annex — Annex D to define vendor specific managed objects, such as temperature, fan and power alarms, for IEEE 802.16-2004 based Base Station.~~

#### 9.3.1 SNMP

SNMP is a protocol to access the managed objects in BS and SS. The support of SNMP is optional for SS.

The support of SNMP in this standard is compliant to SNMPv2, but is backward compatible to SNMPv1 through appropriate translation. The SNMP agent support for SNMPv3 is optional. In case of an agent that implements SNMPv3, it is required to implement at least all the mandatory groups of the standard MIBs required for SNMPv3: RFC3410, RFC3411, RFC3412, RFC3413, RFC3414 and RFC3415 as well as the MIB defining coexistence between SNMPv1, v2 and v3 in RFC 2576. The SNMPv3 framework may be considered as a mechanism to flexibly control access to this MIB module, and mitigate security vulnerability..

SNMP agent shall support RFC3418.

#### 9.3.2 Relationship with interface MIB

This subclause describes the integration of wmanIfMib with MIB-II under Interface Group MIB defined in IETF RFC2863.

This subclause describes the integration with MIB-II under Interface Group MIB defined in IETF RFC2863, as `wmanIfMib` will need to be integrated in the MIB tree. It describes where `wmanIfMib` is located in the MIB-II subtree, and how it can be accessed by NMS.

### 9.3.2.1 MIB-2 integration

The Internet Assigned Number Authority (IANA) has assigned the following ifTypes: ~~to point to multipoint broadband wireless access.~~

```

IANAifType ::= TEXTUAL-CONVENTION
  SYNTAX INTEGER {
    propBWAp2Mp (184)      -- prop broadband wireless access point to multipoint
    propBWAp2Mp (184),    -- prop broadband wireless access point to multipoint
    ieee80216WMAN (237), -- IEEE 802.16 WMAN interface
  }

```

~~Wireless MAN interface table is located under transmission subtree, as follows.~~

```

wmanIfMib ::= {transmission 184} -- WMAN interface table

```

The amendment 802.16f-2005 defined interface type for point-to-multipoint broadband wireless access interfaces as "propBWAp2Mp (184)". This interface type is now deprecated. All new implementations of SNMP agents should use the newly allocated ifType number ieee80216WMAN (237). For backwards compatibility purposed SNMP managers shall accept this deprecated interface type.

### 9.3.2.2 Usage of MIB-II tables

"Interfaces" group of MIB-II, in RFC2863, has been designed to manage various sub-layers (e.g. MAC and PHY) beneath the internetwork-layer for numerous media-specific interfaces. The implementation of ifTable in SNMP managed BS and SS is mandatory.

The implementation of the ifTable for BS must create one row for each BS sector. Each BS sector may support different MAC versions of IEEE 802.16 standard (e.g. IEEE 802.16-2004, IEEE 802.16e-2005). The following recommendations must be applied to each row defining BS sector:

- ifIndex value is implementation specific
- ifType must be set to ~~propBWAp2Mp (value of 184 as defined in 9.3.2.1)~~ ieee80216WMAN (value of 237 as defined in subclause 9.3.2.1)
- ifSpeed must be ~~null~~ set to "0"
- ifPhysAddress must be set to the MAC Address of the BS sector
- All other columnar objects must be initialized as specified in RFC2863.

ifTable	ifIndex	ifType (IANA)	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus
BS Sector 1	1	<del>propBWAp2Mp</del> <u>ieee80216WMAN</u>	<del>Null-0</del>	MAC address of BS sector	Administration Status	Operational Status
BS Sector 2	2	<del>propBWAp2Mp</del> <u>ieee80216WMAN</u>	<del>Null-0</del>	MAC address of BS sector	Administration Status	Operational Status

ifTable	ifIndex	ifType (IANA)	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus
BS Sector 3	3	<u>ieee80216WMAN</u> <u>propBWAp2Mp</u>	<u>Null_0</u>	MAC address of BS sector	Administra- tion Status	Operational Status
Ethernet	4	<u>ethernetCsmacd</u>	<u>Interface</u> <u>Speed</u>	MAC address	Administra- tion Status	Operational Status

**Table 1—Example of the Usage of ifTable objects for base station**

Table 1 shows an example of the usage of ifTable for BS that supports multiple sectors. Each sector may support one of the following MAC / PHY interfaces:

- IEEE 802.16-2004, OFDM 256
- IEEE 802.16-2004, OFDMA 2048
- IEEE 802.16e, OFDMA 128
- IEEE 802.16e, OFDMA 512
- IEEE 802.16e, OFDMA 1024

The implementation of the ifTable for SS must create one row for each SS WirelessMAN interface. Additional rows may be necessary to support other network interfaces, such as Ethernet. The following recommendations must be applied to each row:

- ifIndex value is implementation specific
- ifType must be set to propBWAp2Mp (value of 184 as defined in 9.3.2.1) ieee80216WMAN (value of 237 as defined in subclause 9.3.2.1)
- ifSpeed must be Null set to “0”
- ifPhysAddress must be set to the SS MAC Address (of the WirelessMAN interface)
- All other columnar objects must be initialized as specified in RFC2863

ifTable	ifIndex	ifType (IANA)	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus
SS	1	<u>ieee80216WMAN</u> <u>propBWAp2Mp</u>	<u>Null_0</u>	MAC address of SS	Administra- tion Status	Operational Status
Ethernet	2	<u>ethernetCsmacd</u>	<u>Null</u> <u>Interface</u> <u>Speed</u>	MAC address	Administra- tion Status	Operational Status

**Table 2—Example of the Usage of ifTable objects for subscriber station**

Table 2 shows an example of the usage of ifTable for SS that may support one of the following MAC / PHY interfaces:

- IEEE 802.16-2004, OFDM 256
- IEEE 802.16-2004, OFDMA 2048
- IEEE 802.16e, OFDMA 128
- IEEE 802.16e, OFDMA 512
- IEEE 802.16e, OFDMA 1024

Figure 4 shows a procedure describing how BS can determine the MAC / PHY standard interface and capability a SS / MS can support.

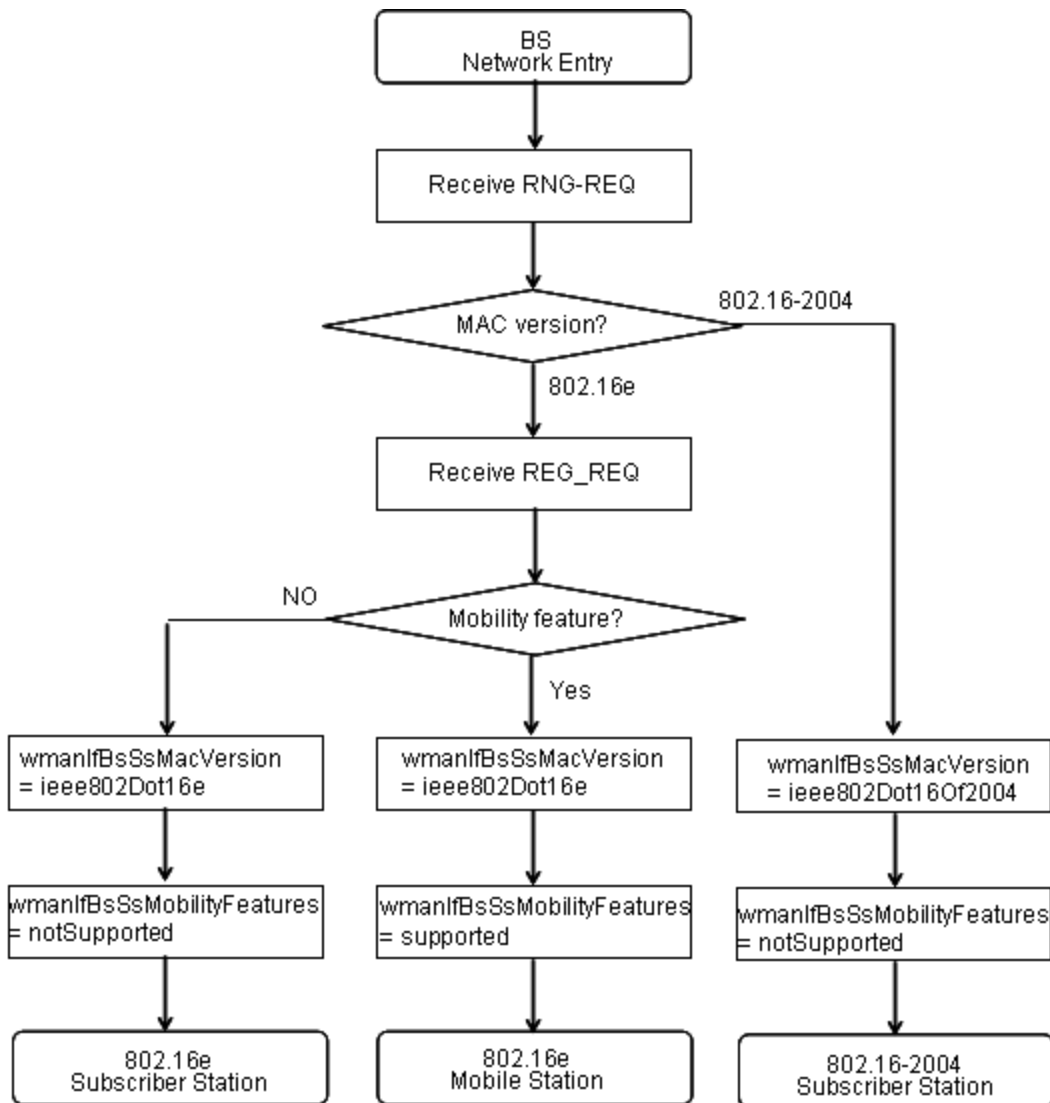


Figure 4—SS / MS Network Entry

1. Receive RNG-REQ from SS / MS
2. If MAC version is 802.16-2004, then
  - a) wmanIfBsSsMacVersion = ieee802Dot16Of2004
  - b) wmanIfBsSsMobilityFeatures = No Supported
  - c) Go to step 5
3. Receive REG-REQ from SS / MS
4. If Mobility Feature is supported, then
  - a) wmanIfBsSsMacVersion = ieee802Dot16e
  - b) wmanIfBsSsMobilityFeatures = Supported
 otherwise
  - a) wmanIfBsSsMacVersion = ieee802Dot16e
  - b) wmanIfBsSsMobilityFeatures = Not Supported

5. Continue network entry procedure

### 9.3.3 Events and traps

The wmanIfMib defines objects for reporting events through mechanisms, such as traps and non-volatile logging. However, the definition and coding of events is vendor-specific. In order to assist the network operators who must troubleshoot multi-vendor equipment, the circumstances and meaning of each event should be reported as human-readable text. Therefore, the trap definitions should include the event reason encoded as display String, and is shown in the following example.

```

trapName NOTIFICATION-TYPE
    OBJECTS      {ifIndex,
                  eventReason,
                  other useful objects
                }
    MAX-Access   read-only
    STATUS       current
    DESCRIPTION
        "trap description"
 ::= { Object Id }.

```

*[Insert a new subclause 9.3.4]*

### 9.3.4 MIB Modules

Table 3 lists all defined ASN.1 MIB modules, their status and module identity OID. The subsequent subclauses give more details about each defined MIB module.

MIB module name	Revision	Status	Module Identity OID
WMAN-IF-MIB	1	Deprecated	iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).transmission(10).wmanIfMib(184)
WMAN-DEV-MIB	1	Active	iso(1).std(0).iso8802(8802).wman(16).wmanDevMib(1)
WMAN-IF2-MIB	1	Active	iso(1).std(0).iso8802(8802).wman(16).wmanIf2Mib(2)
WMAN-IF2M-MIB	1	Active	iso(1).std(0).iso8802(8802).wman(16).wmanIf2mMib(3)

**Table 3—List of ASN.1 MIB Modules**

#### 9.3.4.1 WMAN-IF-MIB module

The WMAN-IF-MIB MIB module defines management objects relevant to the broadband wireless interface as defined in the standard IEEE802.16-2004.

The current status of the WMAN-IF-MIB MIB module is deprecated and the module should not be implemented by the equipment compliant with the amendment IEEE802.16i. The WMAN-IF2-MIB module should be implemented instead.

The WMAN-IF-MIB MIB module is identified by module identity name wmanIfMib and shall be accessed through the following OID:

iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).transmission(10).wmanIfMib(184)

#### **9.3.4.2 WMAN-DEV-MIB module**

The WMAN-DEV-MIB MIB module defines management objects relevant to the device implementing the IEEE 802.16-2004 and 802.16-2005e standards. The objects of this MIB module may refer explicitly to terms defined in the standard (e.g. configuration file encodings) but mainly provide the mandatory support required to manage the devices implementing the IEEE 802.16 interface.

The current status if the WMAN-DEV-MIB is active. The BS shall implement this MIB module. SS shall implement this MIB module if it is managed using SNMP protocol.

The WMAN-DEV-MIB MIB module is identified by module identity name wmanDevMib and shall be accessed through the following OID:

iso(1).std(0).iso8802(8802).wman(16).wmanDevMib(1)

#### **9.3.4.3 WMAN-IF2-MIB module**

The WMAN-IF2-MIB MIB module defines all management objects that are common to all broadband wireless interfaces as defined in the IEEE Std 802.16 standard.

The current status of the WMAN-IF2-MIB MIB module is active. The BS shall implement this MIB module. The SS shall implement this MIB module if it is managed using SNMP protocol.

The WMAN-IF2-MIB MIB module is identified by module identity name wmanIf2Mib and shall be accessed through the following OID:

iso(1).std(0).iso8802(8802).wman(16).wmanIf2Mib(2)

#### **9.3.4.4 WMAN-IF2M-MIB module**

The WMAN-IF2M-MIB MIB module defines all management objects that are specific to mobile broadband wireless interfaces as defined in the IEEE Std 802.16 standard.

The current status of the WMAN-IF2M-MIB MIB module is active. The BS shall implement this MIB module if it supports mobility. The MS shall implement this MIB module if it is managed using SNMP protocol.

The WMAN-IF2M-MIB MIB module is identified by module identity name wmanIf2mMib and shall be accessed through the following OID:

iso(1).std(0).iso8802(8802).wman(16).wmanIf2mMib(3)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



1  
2  
3  
4  
5  
6  
7  
8  
9  
10 *[Change subclause 13 as follows:]*

11  
12  
13 **13. ~~802.16~~ MIB Modules structure for SNMP**

14  
15  
16 *[Delete the 1st paragraph in subclause 13]*

17  
18 *[Change subclause 13.1 as follows:]*

19  
20  
21 **13.1 Structure of MIB modules**

22  
23  
24 *[Move subclause heading 13.1 in 802.16f to subclause 13.1.1]*

25  
26  
27 **13.1.1 wmanIfMib**

28  
29  
30 *[Move 2nd paragraph of subclause 13.1 in 802.16f to subclause 13.1.1]*

31  
32 The wmanIfMib is composed of three groups:

- 33  
34
- 35 • wmanIfBsObjects: contains managed objects to be implemented in the SNMP agent in BS.
  - 36 • wmanIfSsObjects: contains managed objects to be implemented in the SNMP agent in SS.
  - 37 • wmanIfCommonObjects: contains common managed objects to be implemented in the SNMP agent in
  - 38 BS and SS.
- 39  
40

41  
42 *[Move subclause heading 13.2 in 802.16f to subclause 13.1.2]*

43  
44 **13.1.2 wmanDevMib**

45  
46 *[Move 3rd paragraph of subclause 13.1 in 802.16f to subclause 13.1.2]*

47  
48 The wmanDevMib is composed of three groups:

- 49  
50
- 51 • wmanDevBsObjects: contains managed objects to be implemented in the SNMP agent in BS.
  - 52 • wmanDevSsObjects: contains managed objects to be implemented in the SNMP agent in SS.
  - 53 • wmanDevCommonObjects: contains managed objects to be implemented in the SNMP agent in BS/SS.
- 54  
55  
56  
57  
58

59 *[Add a new subclause 13.1.3]*

60  
61  
62  
63  
64  
65

### 13.1.3 wmanIf2Mib

Figure 5 shows the high level MIB structure of wmanIf2Mib for 802.16. The MIB structure is organized based on the the reference model as defined in IEEE 802.16-2004 standard.

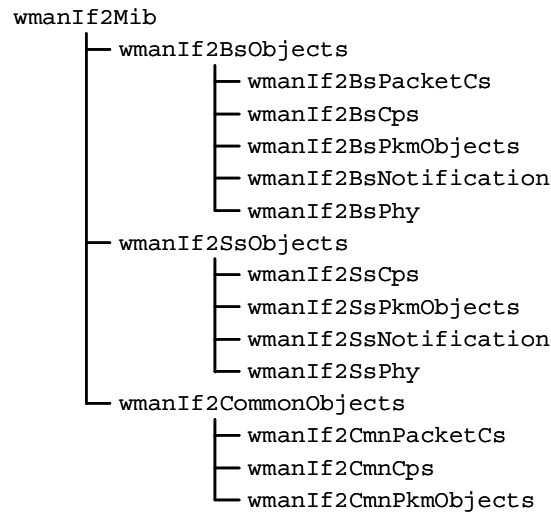


Figure 5—wmanIf2Mib structure

#### 13.1.3.1 wmanIf2BsObjects

##### 13.1.3.1.1 wmanIf2BsPacketCs

Figure 6 shows the structure of wmanIf2BsPacketCs subtree that contains BS managed objects related to the Packet CS management entity layer.

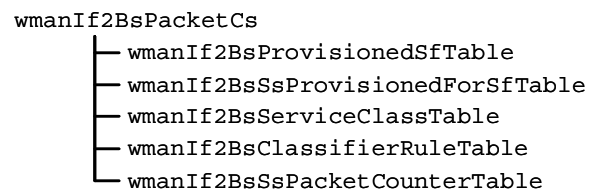


Figure 6—wmanIf2BsPacketCs structure

##### 13.1.3.1.1.1 wmanIf2BsProvisionedSfTable

wmanIf2BsProvisionedSfTable contains provisioned service flow profiles for SSs, and pointers to wmanIf2BsServiceClassTable and wmanIf2BsClassifierRuleTable for QoS parameters and classifier rules respectively.

### 13.1.3.1.1.2 wmanIf2BsProvisionedForSfTable

wmanIf2BsProvisionedForSfTable maps the MAC addresses of SSs to the service flows provisioned in wmanIf2BsProvisionedSfTable. It enables downlink multicast services where MAC addresses of multiple SSs can be mapped to the same service flow.

### 13.1.3.1.1.3 wmanIf2BsServiceClassTable

Each entry of the wmanIf2BsServiceClassTable contains QoS parameter set, as defined in subclause 6.3.14 and 11.13 in IEEE 802.16-2004 standard.

### 13.1.3.1.1.4 wmanIf2BsClassifierRuleTable

wmanIf2BsClassifierRuleTable contains the packet classifier rules associated with service flows.

### 13.1.3.1.1.5 wmanIf2BsSsPacketCounterTable

wmanIf2BsSsPacketCounterTable contains counters to keep track of the number of packets and octets that have been received or transmitted on the per service flow basis.

### 13.1.3.1.2 wmanIf2BsCps

Figure 7 shows the structure of wmanIf2BsCps subtree that contains BS managed objects related to the MAC CPS management entity layer.

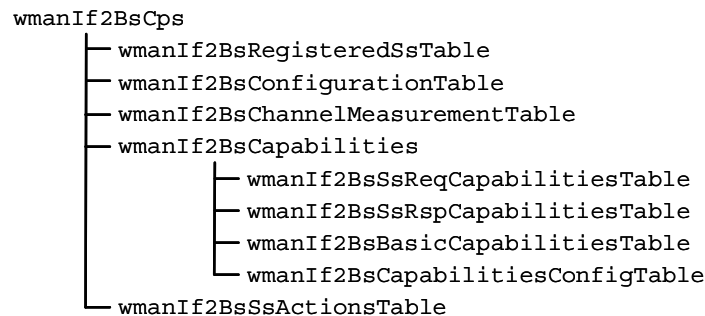


Figure 7—wmanIf2BsCps structure

#### 13.1.3.1.2.1 wmanIf2BsRegisteredSsTable

Each entry in the wmanIf2BsRegisteredSsTable contains the information of SS that has been registered through REG-REQ and REG-RSP messages.

#### 13.1.3.1.2.2 wmanIf2BsConfigurationTable

wmanIf2BsConfigurationTable contains objects for BS system parameters and constants as defined in subclause 10.1 of IEEE 802.16-2004 standard. wmanIf2BsConfigurationTable also contains objects that define the default behaviour of the BS for 2nd Management Channel scheduling and SFID allocation as well as configuration parameters of the CPS scheduler and AAS system.

### 13.1.3.1.2.3 wmanIf2BsChannelMeasurementTable

wmanIf2BsChannelMeasurementTable contains channel measurement information on the uplink signal that were received from SS, and the downlink signal were obtained from SS using REP-REQ/RSP messages..

### 13.1.3.1.2.4 wmanIf2BsCapabilities

#### 13.1.3.1.2.4.1 wmanIf2BsSsReqCapabilitiesTable

wmanIf2BsSsReqCapabilitiesTable contains the basic capability information of SSs that have been reported by SSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages.

#### 13.1.3.1.2.4.2 wmanIf2BsSsRspCapabilitiesTable

wmanIf2BsSsRspCapabilitiesTable contains the basic capability information of SSs that have been negotiated and agreed between BS and SS via RNG-REQ/RSP, SBC-REQ/RSP and REG-REQ/RSP messages.

#### 13.1.3.1.2.4.3 wmanIf2BsBasicCapabilitiesTable

wmanIf2BsBasicCapabilitiesTable contains the basic capabilities of the BS as implemented in BS hardware and software. These capabilities along with the configuration for them (wmanIf2BsCapabilitiesConfigTable) are used for negotiation of basic capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP messages.

#### 13.1.3.1.2.4.4 wmanIf2BsCapabilitiesConfigTable

wmanIf2BsCapabilitiesConfigTable contains the configuration for basic capabilities of BS. The table is intended to be used to restrict the Capabilities implemented by BS, for example in order to comply with local regulatory requirements. The BS should use the configuration along with the implemented Capabilities (wmanIf2BsBasicCapabilitiesTable) for negotiation of basic capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP messages.

### 13.1.3.1.2.5 wmanIf2BsSsActionsTable

wmanIf2BsSsActionsTable contains all the actions specified for SSs in the standard. The actions are routed down to SS using nsolicited MAC messages: REG-RSP, DREG-REQ and RES-CMD. The table also contains the parameters of the actions in cases where they are specified by the standard.

### 13.1.3.1.3 wmanIf2BsPkmObjects

Figure 8 shows the structure of wmanIf2BsPkmObjects subtree that contains BS managed objects related to the MAC privacy management entity.

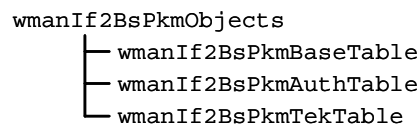


Figure 8—wmanIf2BsPkmObjects structure

#### 13.1.3.1.3.1 wmanIf2BsPkmBaseTable

wmanIf2BsPkmBaseTable contains base station PKM operational parameters described in subclause 10.2 of IEEE 802.16-2004 standard.

#### 13.1.3.1.3.2 wmanIf2BsSsPkmAuthTable

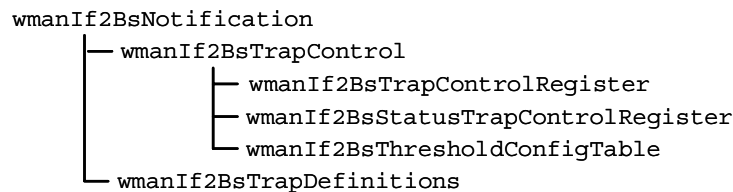
wmanIf2BsSsPkmAuthTable contains runtime subscriber station authentication and authorization parameters for each base station.

#### 13.1.3.1.3.3 wmanIf2BsPkmTekTable

wmanIf2BsPkmTekTable is double indexed by ifIndex and SAId and contains runtime Security association parameters for each base station.

#### 13.1.3.1.4 wmanIf2BsNotification

Figure 9 shows the structure of wmanIf2BsNotification subtree that contains BS traps to report fault events and exceptions, such as power status, RSSI threshold crossing.



**Figure 9—wmanIf2BsNotification structure**

#### 13.1.3.1.4.1 wmanIf2BsTrapControl

##### 13.1.3.1.4.1.1 wmanIf2BsTrapControlRegister

wmanIf2BsTrapControlRegister is used to enable or disable Base traps independently.

##### 13.1.3.1.4.1.2 wmanIf2BsStatusTrapControlRegister

wmanIf2BsStatusTrapControlRegister is used to enable or disable Base Station status notification traps.

##### 13.1.3.1.4.1.3 wmanIf2BsThresholdConfigTable

wmanIf2BsThresholdConfigTable contains threshold objects that can be set to detect the threshold crossing events.

##### 13.1.3.1.4.2 wmanIf2BsTrapDefinitions

wmanIf2BsTrapDefinitions object group defines all the traps reported by BS.

#### 13.1.3.1.5 wmanIf2BsPhy

Figure 10 shows the structure of wmanIf2BsPhy subtree that contains BS managed objects related to the Physical layer.



Figure 10—wmanIf2BsPhy structure

#### 13.1.3.1.5.1 wmanIf2BsOfdmPhy

wmanIf2BsOfdmPhy is a group containing objects specific to OFDM PHY.

##### 13.1.3.1.5.1.1 wmanIf2BsOfdmUplinkChannelTable

wmanIf2BsOfdmUplinkChannelTable contains OFDM UCD (Uplink Channel Descriptor) channel attributes, defining the transmission characteristics of uplink channels.

##### 13.1.3.1.5.1.2 wmanIf2BsOfdmDownlinkChannelTable

wmanIf2BsOfdmDownlinkChannelTable contains OFDM DCD (Downlink Channel Descriptor) channel attributes, defining the transmission characteristics of downlink channels.

##### 13.1.3.1.5.1.3 wmanIf2BsOfdmUcdBurstProfileTable

wmanIf2BsOfdmUcdBurstProfileTable contains OFDM UCD burst profiles for each uplink channel.

##### 13.1.3.1.5.1.4 wmanIf2BsOfdmDcdBurstProfileTable

wmanIf2BsOfdmDcdBurstProfileTable provides one row for each OFDM DCD burst profile.

##### 13.1.3.1.5.1.5 wmanIf2BsOfdmConfigurationTable

wmanIf2BsOfdmConfigurationTable contains BS configuration objects, specific to OFDM PHY.

#### 13.1.3.1.5.1.6 wmanIf2BsSsOfdmReqCapabilitiesTable

wmanIf2BsSsOfdmReqCapabilitiesTable contains the basic capability information, specific to OFDM Phy, of SSs that have been reported by SSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages. Entries in this table should be created when an SS registers with a BS.

#### 13.1.3.1.5.1.7 wmanIf2BsSsOfdmRspCapabilitiesTable

wmanIf2BsSsOfdmRspCapabilitiesTable contains the basic capability information, specific to OFDM Phy, of SSs that have been negotiated and agreed between BS and SS via RNG-REQ/RSP, SBC-REQ/RSP and REG-REQ/RSP messages. This table augments the wmanIf2BsRegisteredSsTable.

#### 13.1.3.1.5.1.8 wmanIf2BsOfdmCapabilitiesTable

wmanIf2BsOfdmCapabilitiesTable contains the basic capabilities, specific to OFDM Phy, of the BS as implemented in BS hardware and software.

#### 13.1.3.1.5.1.9 wmanIf2BsOfdmCapabilitiesConfigTable

wmanIf2BsOfdmCapabilitiesConfigTable contains the configuration for basic capabilities of BS, specific to OFDM Phy. The table is intended to be used to restrict the Capabilities implemented by BS.

#### 13.1.3.1.5.2 wmanIf2BsOfdmaPhy

wmanIf2BsOfdmaPhy is a group containing objects specific to OFDMA PHY.

##### 13.1.3.1.5.2.1 wmanIf2BsOfdmaUplinkChannelTable

wmanIf2BsOfdmaUplinkChannelTable contains OFDMA UCD channel attributes, defining the transmission characteristics of uplink channels.

##### 13.1.3.1.5.2.2 wmanIf2BsOfdmaDownlinkChannelTable

wmanIf2BsOfdmaDownlinkChannelTable contains OFDMA DCD channel attributes, defining the transmission characteristics of downlink channels.

##### 13.1.3.1.5.2.3 wmanIf2BsOfdmaUcdBurstProfileTable

wmanIf2BsOfdmaUcdBurstProfileTable contains OFDMA UCD burst profiles for each uplink channel.

##### 13.1.3.1.5.2.4 wmanIf2BsOfdmaDcdBurstProfileTable

wmanIf2BsOfdmaDcdBurstProfileTable provides one row for each OFDMA DCD burst profile.

##### 13.1.3.1.5.2.5 wmanIf2BsOfdmaConfigurationTable

wmanIf2BsOfdmaConfigurationTable contains BS configuration objects, specific to OFDMA PHY.

#### 13.1.3.1.5.2.6 wmanIf2BsSsOfdmaReqCapabilitiesTable

wmanIf2BsSsOfdmaReqCapabilitiesTable contains the basic capability information, specific to OFDMA Phy, of SSs or MSs that have been reported by SSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages. Entries in this table should be created when an SS registers with a BS.

#### 13.1.3.1.5.2.7 wmanIf2BsSsOfdmaRspCapabilitiesTable

wmanIf2BsSsOfdmaRspCapabilitiesTable contains the basic capability information, specific to OFDMA Phy, of SSs or MSs that have been negotiated and agreed between BS and SS via RNG-REQ/RSP, SBC-REQ/RSP and REG-REQ/RSP messages. This table augments the wmanIf2BsRegisteredSsTable.

#### 13.1.3.1.5.2.8 wmanIf2BsOfdmaCapabilitiesTable

wmanIf2BsOfdmaCapabilitiesTable contains the basic capabilities, specific to OFDMA Phy, of the BS as implemented in BS hardware and software.

#### 13.1.3.1.5.2.9 wmanIf2BsOfdmaCapabilitiesConfigTable

wmanIf2BsOfdmaCapabilitiesConfigTable contains the configuration for basic capabilities of BS, specific to OFDMA Phy. The table is intended to be used to restrict the Capabilities implemented by BS.

#### 13.1.3.1.5.2.10 wmanIf2BsOfdmaExUplinkChannelTable

wmanIf2BsOfdmaExUplinkChannelTable arguments wmanIf2BsOfdmaUplinkChannelTable to contain new UCD channel encodings that have been added to IEEE 802.16e 2005.

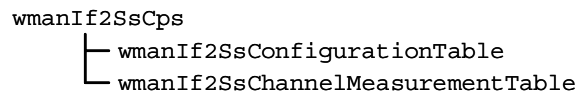
#### 13.1.3.1.5.2.11 wmanIf2BsOfdmaExDownlinkChannelTable

wmanIf2BsOfdmaExDownlinkChannelTable arguments wmanIf2BsOfdmaDownlinkChannelTable to contain new DCD channel encodings that have been added to IEEE 802.16e 2005.

### 13.1.3.2 wmanIf2SsObjects

#### 13.1.3.2.1 wmanIf2SsCps

Figure 11 shows the structure of wmanIf2SsCps subtree that contains SS managed objects related to the MAC CPS management entity layer.



**Figure 11—wmanIf2SsCps structure**

#### 13.1.3.2.1.1 wmanIf2SsConfigurationTable

wmanIf2SsConfigurationTable contains objects for SS system parameters and constants as defined in sub-clause 10.1 of IEEE 802.16-2004 standard.

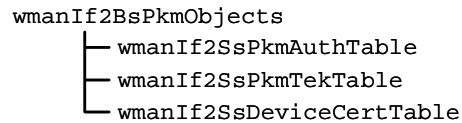
#### 13.1.3.2.1.2 wmanIf2SsChannelMeasurementTable

wmanIf2SsChannelMeasurementTable contains downlink channel measurement information for each SS.

#### 13.1.3.2.2 wmanIf2SsPkmObjects

Figure 12 shows the structure of wmanIf2SsPkmObjects subtree that contains subscriber station manageable objects related to the privacy management entity.





**Figure 12—wmanIf2SsPkmObjects structure**

#### 13.1.3.2.2.1 wmanIf2SsPkmAuthTable

wmanIf2SsPkmAuthTable contains subscriber station authentication and authorization parameters including those described in subclause 10.2 of IEEE 802.16-2004.

#### 13.1.3.2.2.2 wmanIf2SsPkmTekTable

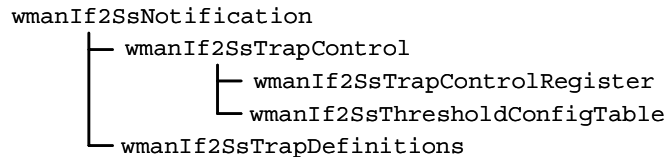
wmanIf2SsPkmTekTable contains subscriber station runtime parameters for each active security association.

#### 13.1.3.2.2.3 wmanIf2SsDeviceCertTable

wmanIf2SsDeviceCertTable describes the PKM device certificates for each SS wireless interface.

#### 13.1.3.2.3 wmanIf2SsNotification

Figure 13 shows the structure of wmanIf2SsNotification subtree that contains SS traps to report fault events and exceptions, such as RSSI threshold crossing.



**Figure 13—wmanIf2SsNotification structure**

#### 13.1.3.2.3.1 wmanIf2SsTrapControl

##### 13.1.3.2.3.1.1 wmanIf2SsTrapControlRegister

wmanIf2SsTrapControlRegister is used to enable or disable Subscriber Station traps.

##### 13.1.3.2.3.1.2 wmanIf2SsThresholdConfigTable

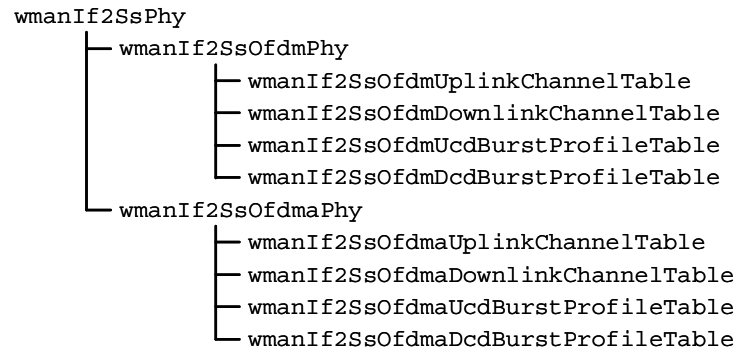
wmanIf2SsThresholdConfigTable contains threshold objects that can be set to detect the threshold crossing events.

##### 13.1.3.2.3.2 wmanIf2SsTrapDefinitions

wmanIf2SsTrapDefinitions group defines all the traps reported by SS.

### 13.1.3.2.4 wmanIf2SsPhy

Figure 14 shows the structure of wmanIf2SsPhy subtree that contains SS managed objects related to the Physical layer.



**Figure 14—wmanIf2SsPhy structure**

#### 13.1.3.2.4.1 wmanIf2SsOfdmPhy

wmanIf2SsOfdmPhy is a group containing objects specific to OFDM PHY.

##### 13.1.3.2.4.1.1 wmanIf2SsOfdmUplinkChannelTable

wmanIf2SsOfdmUplinkChannelTable contains OFDM UCD channel attributes, defining the transmission characteristics of uplink channels.

##### 13.1.3.2.4.1.2 wmanIf2SsOfdmDownlinkChannelTable

wmanIf2SsOfdmDownlinkChannelTable contains OFDM DCD channel attributes, defining the transmission characteristics of downlink channels.

##### 13.1.3.2.4.1.3 wmanIf2SsOfdmUcdBurstProfileTable

wmanIf2SsOfdmUcdBurstProfileTable contains OFDM UCD burst profiles for each uplink channel.

##### 13.1.3.2.4.1.4 wmanIf2SsOfdmDcdBurstProfileTable

wmanIf2SsOfdmDcdBurstProfileTable provides one row for each OFDM DCD burst profile.

#### 13.1.3.2.4.2 wmanIf2SsOfdmaPhy

wmanIf2SsOfdmaPhy is a group containing objects specific to OFDMA PHY.

##### 13.1.3.2.4.2.1 wmanIf2SsOfdmaUplinkChannelTable

wmanIf2SsOfdmaUplinkChannelTable contains OFDMA UCD channel attributes, defining the transmission characteristics of uplink channels.

#### 13.1.3.2.4.2.2 wmanIf2SsOfdmaDownlinkChannelTable

wmanIf2SsOfdmaDownlinkChannelTable contains OFDMA DCD channel attributes, defining the transmission characteristics of downlink channels.

#### 13.1.3.2.4.2.3 wmanIf2SsOfdmaUcdBurstProfileTable

wmanIf2SsOfdmaUcdBurstProfileTable contains OFDMA UCD burst profiles for each uplink channel.

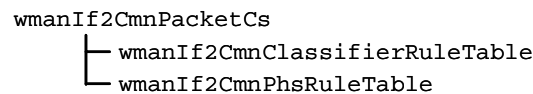
#### 13.1.3.2.4.2.4 wmanIf2SsOfdmaDcdBurstProfileTable

wmanIf2SsOfdmaDcdBurstProfileTable provides one row for each OFDMA DCD burst profile.

### 13.1.3.3 wmanIf2CommonObjects

#### 13.1.3.3.1 wmanIf2CmnPacketCs

Figure 15 shows the structure of wmanIf2CmnPacketCs subtree that contains common managed objects related to the Packet CS management entity layer.



**Figure 15—wmanIf2CmnPacketCs structure**

#### 13.1.3.3.1.1 wmanIf2CmnClassifierRuleTable

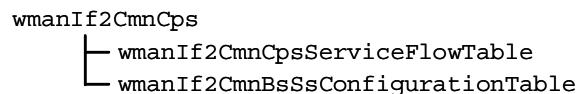
wmanIf2CmnClassifierRuleTable contains runtime classifier rules screening criteria for each service flow.

#### 13.1.3.3.1.2 wmanIf2CmnPhsRuleTable

wmanIf2CmnPhsRuleTable contains PHS rule dictionary entries. Each entry contains the data of the header to be suppressed along with its identification - PHSI.

#### 13.1.3.3.2 wmanIf2CmnCps

Figure 16 shows the structure of wmanIf2CmnCps subtree that contains common managed objects related to the MAC CPS management entity.



**Figure 16—wmanIf2CmnCps structure**

#### 13.1.3.3.2.1 wmanIf2CmnCpsServiceFlowTable

wmanIf2CmnCpsServiceFlowTable contains Service Flow managed objects that are common in BS and SS.

### 13.1.3.3.2 wmanIf2CmnBsSsConfigurationTable

wmanIf2CmnBsSsConfigurationTable provides one row for each BS sector that contains the system parameters common in both SS and BS. All SSs shall have the same parameters as the BS to which the SSs are associated.

### 13.1.3.3.3 wmanIf2CmnPkmObjects

Figure 17 shows the structure of wmanIf2CmnPkmObjects subtree that contains common PKM objects.

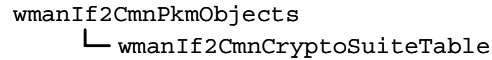


Figure 17—wmanIf2CmnPkmObjects structure

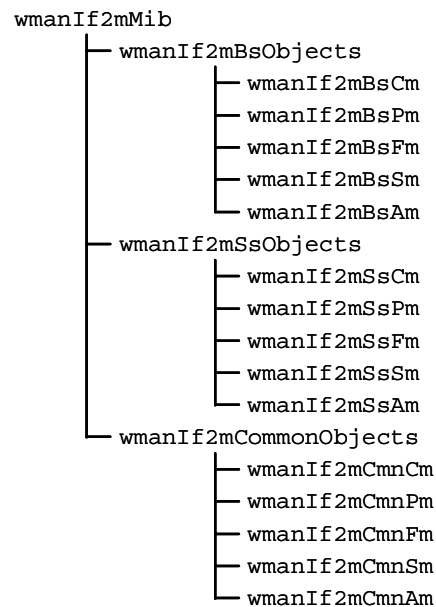
#### 13.1.3.3.3.1 wmanIf2CmnCryptoSuiteTable

wmanIf2CmnCryptoSuiteTable contains supported crypto suites for the particular SS and other crypto parameters such as key lifetimes.

### *[Add a new subclause 13.1.4]*

## 13.1.4 wmanIf2mMib

Figure 18 shows the high level MIB structure of wmanIf2mMib for IEEE 802.16e-2005. The MIB structure is organized based on the the FCAPS reference model.

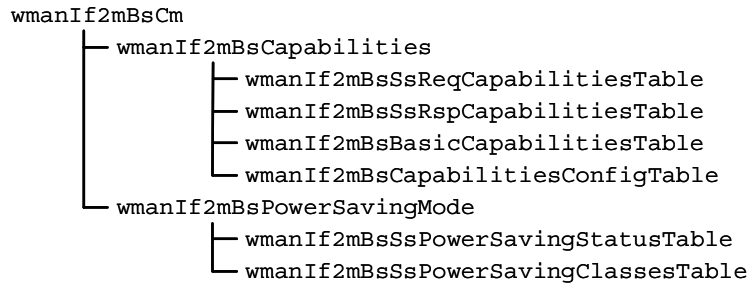


## Figure 18—wmanIf2mMib structure

### 13.1.4.1 wmanIf2mBsObjects

#### 13.1.4.1.1 wmanIf2mBsCm

Figure 19 shows the structure of wmanIf2mBsCm subtree that contains BS managed objects related to Configuration Management.



## Figure 19—wmanIf2mBsCm structure

### 13.1.4.1.1.1 wmanIf2mBsCapabilities

#### 13.1.4.1.1.1.1 wmanIf2mBsSsReqCapabilitiesTable

wmanIf2mBsSsReqCapabilitiesTable contains the basic capability information of SSs that have been reported by SSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages

#### 13.1.4.1.1.1.2 wmanIf2mBsSsRspCapabilitiesTable

wmanIf2mBsSsRspCapabilitiesTable contains the basic capability information of SSs that have been negotiated and agreed between BS and SS via RNG-REQ/RSP, SBC-REQ/RSP and REG-REQ/RSP messages

#### 13.1.4.1.1.1.3 wmanIf2mBsBasicCapabilitiesTable

wmanIf2mBsBasicCapabilitiesTable contains the basic capabilities of the BS as implemented in BS hardware and software. These capabilities along with the configuration for them (wmanIf2mBsCapabilitiesConfigTable) are used for negotiation of basic capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP messages.

#### 13.1.4.1.1.1.4 wmanIf2mBsCapabilitiesConfigTable

wmanIf2mBsCapabilitiesConfigTable contains the configuration for basic capabilities of BS. The table is intended to be used to restrict the Capabilities implemented by BS, for example in order to comply with local regulatory requirements. The BS should use the configuration along with the implemented Capabilities (wmanIf2mBsBasicCapabilitiesTable) for negotiation of basic capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP messages.

### 13.1.4.1.1.2 wmanIf2mBsPowerSavingMode

#### 13.1.4.1.1.2.1 wmanIf2mBsSsPowerSavingStatusTable

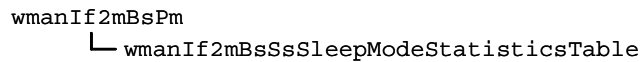
wmanIf2mBsSsPowerSavingStatusTable contains the power saving status for each CID in an SS.

#### 13.1.4.1.1.2.2 wmanIf2mBsSsPowerSavingClassesTable

wmanIf2mBsSsPowerSavingClassesTable contains the power saving classes definitions, and activation / deactivation information that are provided by MOB\_SLP-REQ and MOB\_SLP-RSP messages..

### 13.1.4.1.2 wmanIf2mBsPm

Figure 20 shows the structure of wmanIf2mBsPm subtree that contains BS managed objects related to Performance Management.



**Figure 20—wmanIf2mBsPm structure**

#### 13.1.4.1.2.1 wmanIf2mBsSsSleepModeStatisticsTable

wmanIf2mBsSsSleepModeStatisticsTable contains the sleep mode statistic for SS.

## 13.2 ASN.1 Definitions of MIB Modules

### 13.2.1 wmanIfMib

### 13.2.2 wmanDevMib

### 13.2.3 wmanIf2Mib

```

1
2
3
4 WMAN-IF2-MIB DEFINITIONS ::= BEGIN
5
6     IMPORTS
7         MODULE-IDENTITY,
8         OBJECT-TYPE,
9         NOTIFICATION-TYPE,
10        Unsigned32, Integer32, Counter32,
11        Counter64, transmission
12            FROM SNMPv2-SMI
13        SnmpAdminString
14            FROM SNMP-FRAMEWORK-MIB
15        TEXTUAL-CONVENTION,
16        MacAddress, RowStatus, TruthValue,
17        TimeStamp, DateAndTime
18            FROM SNMPv2-TC
19        InetAddressType, InetAddress
20            FROM INET-ADDRESS-MIB
21        OBJECT-GROUP,
22        MODULE-COMPLIANCE,
23        NOTIFICATION-GROUP
24            FROM SNMPv2-CONF
25        ifIndex
26            FROM IF-MIB;
27
28
29
30
31
32 wmanIf2mMib MODULE-IDENTITY
33     LAST-UPDATED      "200611280000Z" -- November 28, 2006
34     ORGANIZATION      "IEEE 802.16"
35     CONTACT-INFO
36         "WG E-mail: stds-802-16@ieee.org
37         WG Chair: Roger B. Marks
38         Postal: NextWave Broadband, Inc.
39         E-mail: r.b.marks@ieee.org
40
41         TGF Chair: Phillip Barber
42         Postal: Huawei Technologies Co., Ltd
43         E-mail: pbarber@futurewei.com
44
45         Editor: Joey Chou
46         Postal: Intel Corporation
47                 5000 W. Chandler Blvd,
48                 Chandler, AZ 85227, USA
49         E-mail: joey.chou@intel.com"
50
51
52
53     DESCRIPTION
54         "This material is from IEEE Std 802.16i
55         Copyright (c) 2006 IEEE.
56         This MIB Module defines managed objects for
57         Subscriber Station and Base Station based on IEEE Std
58         802.16-2004 and its amendment IEEE Std 802.16e-2005.
59         The MIB contains managed objects that are specific
60         to mobile Broadband Wireless Networks."
61
62
63     REVISION           "20061120000Z"
64
65     DESCRIPTION

```

```

1         "WMAN-IF2M-MIB module that is included in
2         IEEE 802.16i-06/001r5."
3     REVISION         "200605230000Z"
4     DESCRIPTION
5
6         "The first revision of WMAN-IF2-MIB module that is
7         enhanced to support IEEE 802.16e-2005 standard."
8     ::= { iso std(0) iso8802(8802) wman(16) 2 }
9
10
11     wmanIf2MibObjects    OBJECT IDENTIFIER ::= { wmanIf2Mib 1 }
12     wmanIf2BsObjects    OBJECT IDENTIFIER ::= { wmanIf2MibObjects 1 }
13     wmanIf2SsObjects    OBJECT IDENTIFIER ::= { wmanIf2MibObjects 2 }
14     wmanIf2CommonObjects OBJECT IDENTIFIER ::= { wmanIf2MibObjects 3 }
15
16
17     -- Textual Conventions
18     WmanIf2SfSchedulingType ::= TEXTUAL-CONVENTION
19         STATUS         current
20         DESCRIPTION
21
22             "The scheduling service provided by a SC for an
23             upstream service flow. If the parameter is omitted
24             from an upstream QOS Parameter Set, this object takes
25             the value of bestEffort (2). This parameter must be
26             reported as undefined (1) for downstream QOS Parameter
27             Sets."
28         SYNTAX         INTEGER {undefined(1),
29                         bestEffort(2),
30                         nonRealTimePollingService(3),
31                         realTimePollingService(4),
32                         reserved(5),
33                         unsolicitedGrantService(6)}
34
35
36
37     WmanIf2PhsRuleVerify ::= TEXTUAL-CONVENTION
38         STATUS         current
39         DESCRIPTION
40
41             "The value of this field indicates to the sending entity
42             whether or not the packet header contents are to be
43             verified prior to performing suppression. If PHSV is
44             enabled, the sender shall compare the bytes in the packet
45             header with the bytes in the PHSF that are to be
46             suppressed as indicated by the PHSM."
47         REFERENCE
48
49             "Subclause 11.13.19.3.7.5 in IEEE Std 802.16-2004"
50         SYNTAX         INTEGER {phsVerifyEnable(0),
51                         phsVerifyDisable(1)}
52
53
54     WmanIf2ClassifierBitMap ::= TEXTUAL-CONVENTION
55         STATUS         current
56         DESCRIPTION
57
58             "A bit of of this object is set to 1 if the parameter
59             indicated by the comment was present in the classifier
60             encoding, and 0 otherwise.
61             Note: that BITS are encoded most significant bit first,
62             so that if e.g. bits 6 and 7 are set, this object is
63             encoded as the octet string '030000'H."
64         REFERENCE
65

```



```

1         "Subclause 11.13.19.3.4 in IEEE Std 802.16-2004"
2     SYNTAX     BITS {priority(0),
3                 ipTos(1),
4                 ipProtocol(2),
5                 ipMaskedSrcAddr(3),
6                 ipMaskedDestAddr(4),
7                 srcPort(5),
8                 destPort(6),
9                 destMacAddr(7),
10                srcMacAddr(8),
11                ethernetProtocol(9),
12                userPriority(10),
13                vlanId(11),
14                ipv6FlowLabel(12)}
15
16 WmanIf2SfState ::= TEXTUAL-CONVENTION
17     STATUS     current
18     DESCRIPTION
19         "Defines the state of a service flow."
20     SYNTAX     INTEGER {authorized(1),
21                 admitted(2),
22                 active(3)}
23
24 WmanIf2ServClassName ::= TEXTUAL-CONVENTION
25     STATUS     current
26     DESCRIPTION
27         "Defines the type of service class name."
28     SYNTAX     OCTET STRING (SIZE(2..128))
29
30 WmanIf2CsSpecification ::= TEXTUAL-CONVENTION
31     STATUS     current
32     DESCRIPTION
33         "Defines the types of convergence sublayer."
34     REFERENCE
35         "Subclause 11.13.19.1 in IEEE Std 802.16e-2005"
36     SYNTAX     INTEGER {reserved(0),
37                 packetIPv4(1),
38                 packetIPv6(2),
39                 packet802dot3Ethernet(3),
40                 packet802dot1QVlan(4),
41                 packetIPv4Over802dot3(5),
42                 packetIPv6Over802dot3(6),
43                 packetIPv4Over802dot1Q(7),
44                 packetIPv6Over802dot1Q(8),
45                 atm(9),
46                 packet802dot3EthernetRohcHc(10),
47                 packet802dot3EthernetEcrtPc(11),
48                 packetIp2RohcHc(12),
49                 packetIp2EcrtPc(13)}
50
51 WmanIf2MacVersion ::= TEXTUAL-CONVENTION
52     STATUS     current
53     DESCRIPTION
54         "Version number of IEEE 802.16."
55
56
57
58
59
60
61
62
63
64
65

```

```

1          SYNTAX          INTEGER {ieee802Dot16Of2001(1),
2                                ieee802Dot16cOf2002(2),
3                                ieee802Dot16aOf2003(3),
4                                ieee802Dot16Of2004(4),
5                                ieee802Dot16e(5),
6                                tbd(6)}
7
8
9
10         WmanIf2CidType ::= TEXTUAL-CONVENTION
11             STATUS          current
12             DESCRIPTION
13                 "Type of CID."
14             SYNTAX          INTEGER (0 .. 65535)
15
16
17         WmanIf2DataEncryptAlgId ::= TEXTUAL-CONVENTION
18             STATUS          current
19             DESCRIPTION
20                 "Data encryption algorithm identifiers."
21             REFERENCE
22                 "Table 375 in IEEE Std 802.16-2004"
23             SYNTAX          INTEGER {none(0),
24                                   des56BitCbcMode(1),
25                                   aesCcmMode(2)}
26
27
28
29         WmanIf2DataAuthAlgId ::= TEXTUAL-CONVENTION
30             STATUS          current
31             DESCRIPTION
32                 "Data authentication algorithm identifiers."
33             REFERENCE
34                 "Table 376 in IEEE Std 802.16-2004"
35             SYNTAX          INTEGER {noDataAuthentication(0),
36                                   reserved(1)}
37
38
39
40         WmanIf2TekEncryptAlgId ::= TEXTUAL-CONVENTION
41             STATUS          current
42             DESCRIPTION
43                 "TEK encryption algorithm identifiers."
44             REFERENCE
45                 "Table 377 in IEEE Std 802.16-2004"
46             SYNTAX          INTEGER {tripleDes128BitKey(1),
47                                   rsa1024BitKey(2),
48                                   aes128BitKey(3)}
49
50
51
52         WmanIf2ChannelNumber ::= TEXTUAL-CONVENTION
53             STATUS          current
54             DESCRIPTION
55                 "Physical channel number"
56             SYNTAX          INTEGER (0 .. 199)
57
58
59         WmanIf2OfdmFecCodeType ::= TEXTUAL-CONVENTION
60             STATUS          current
61             DESCRIPTION
62                 "FEC code type and modulation type"
63             REFERENCE
64                 "Table 356 and Table 362 in IEEE Std 802.16-2004"
65

```

```

1      SYNTAX      INTEGER {bpskCc1Over2(0),
2                    qpskRsCcCc1Over2(1),
3                    qpskRsCcCc3Over4(2),
4                    sixteenQamRsCcCc1Over2(3),
5                    sixteenQamRsCcCc3Over4(4),
6                    sixtyFourQamRsCcCc2Over3(5),
7                    sixtyFourQamRsCcCc3Over4(6),
8                    qpskBtc1Over2(7),
9                    qpskBtc3Over4(8),
10                   sixteenQamBtc3Over4(9),
11                   sixteenQamBtc4Over5(10),
12                   sixtyFourQamBtc2Over3(11),
13                   sixtyFourQamBtc5Over6(12),
14                   qpskCtc1Over2(13),
15                   qpskCtc2Over3(14),
16                   qpskCtc3Over4(15),
17                   sixteenQamCtc1Over2(16),
18                   sixteenQamCtc3Over4(17),
19                   sixtyFourQamCtc2Over3(18),
20                   sixtyFourQamCtc3Over4(19)}
21
22 WmanIf2OfdmaFecCodeType ::= TEXTUAL-CONVENTION
23     STATUS      deprecated
24     DESCRIPTION
25         "FEC code type and modulation type"
26     REFERENCE
27         "Table 356 and Table 362 in IEEE Std 802.16-2004"
28     SYNTAX      INTEGER {qpskCc1Over2(0),
29                       qpskCc3Over4(1),
30                       sixteenQamCc1Over2(2),
31                       sixteenQamCc3Over4(3),
32                       sixtyFourQamCc2Over3(4),
33                       sixtyFourQamCc3Over4(5),
34                       qpskBtc1Over2(6),
35                       qpskBtc2Over3(7),
36                       sixteenQamBtc3Over5(8),
37                       sixteenQamBtc4Over5(9),
38                       sixtyFourQamBtc5Over8(10),
39                       sixtyFourQamBtc4Over5(11),
40                       qpskCtc1Over2(12),
41                       qpskCtc2Over3(13),
42                       qpskCtc3Over4(14),
43                       sixteenQamCtc1Over2(15),
44                       sixteenQamCtc3Over4(16),
45                       sixtyFourQamCtc2Over3(17),
46                       sixtyFourQamCtc3Over4(18),
47                       sixtyFourQamCtc5Over6(19),
48                       qpskZtCc1Over2(20),
49                       qpskZtCc3Over4(21),
50                       sixteenQamZtCc1Over2(22),
51                       sixteenQamZtCc3Over4(23),
52                       sixtyFourQamZtCc2Over3(24),
53                       sixtyFourQamZtCc3Over4(25)}
54
55
56
57
58
59
60
61
62
63
64
65

```

```

1  WmanIf2OfdmaUcdFecCode ::= TEXTUAL-CONVENTION
2      STATUS          current
3      DESCRIPTION
4          "UCD FEC code type and modulation type"
5      REFERENCE
6          "Table 357 in IEEE Std 802.16e-2005"
7      SYNTAX          INTEGER {qpskCc1Over2(0),
8                          qpskCc3Over4(1),
9                          sixteenQamCc1Over2(2),
10                         sixteenQamCc3Over4(3),
11                         sixtyFourQamCc1Over2(4),
12                         sixtyFourQamCc2Over3(5),
13                         sixtyFourQamCc3Over4(6),
14                         qpskBtc1Over2(7),
15                         qpskBtc3Over4(8),
16                         sixteenQamBtc3Over5(9),
17                         sixteenQamBtc4Over5(10),
18                         sixtyFourQamBtc5Over8(11),
19                         sixtyFourQamBtc4Over5(12),
20                         qpskCtc1Over2(13),
21                         reserved(14),
22                         qpskCtc3Over4(15),
23                         sixteenQamCtc1Over2(16),
24                         sixteenQamCtc3Over4(17),
25                         sixtyFourQamCtc1Over2(18),
26                         sixtyFourQamCtc2Over3(19),
27                         sixtyFourQamCtc3Over4(20),
28                         sixtyFourQamCtc5Over6(21),
29                         qpskZtCc1Over2(22),
30                         qpskZtCc3Over4(23),
31                         sixteenQamZtCc1Over2(24),
32                         sixteenQamZtCc3Over4(25),
33                         sixtyFourQamZtCc1Over2(26),
34                         sixtyFourQamZtCc2Over3(27),
35                         sixtyFourQamZtCc3Over4(28),
36                         qpskLdpc1over2(29),
37                         qpskLdpc2over3A(30),
38                         qpskLdpc3over4A(31),
39                         sixteenQamLdpc1over2(32),
40                         sixteenQamLdpc2over3A(33),
41                         sixteenQamLdpc3over4A(34),
42                         sixtyFourQamLdpc1over2(35),
43                         sixtyFourQamLdpc2over3A(36),
44                         sixtyFourQamLdpc3over4A(37),
45                         qpskLdpc2over3B(38),
46                         qpskLdpc3over4B(39),
47                         sixteenQamLdpc2over3B(40),
48                         sixteenQamLdpc3over4B(41),
49                         sixtyFourQamLdpc2over3B(42),
50                         sixtyFourQamLdpc3over4B(43),
51                         qpskCcOptIntv1over2(44),
52                         qpskCcOptIntv3over4(45),
53                         sixteenQamCcOptIntv1over2(46),
54                         sixteenQamCcOptIntv3over4(47),
55
56
57
58
59
60
61
62
63
64
65

```

```

1          sixtyFourQamCcOptIntv2over3 (48) ,
2          sixtyFourQamCcOptIntv3over4 (49) ,
3          qpskLdpc5over6 (50) ,
4          sixteenQamLdpc5over6 (51) ,
5          sixtyFourQamLdpc5over6 (52) }
6
7
8
9 WmanIf2OfdmaDcdFecCode ::= TEXTUAL-CONVENTION
10     STATUS          current
11     DESCRIPTION
12         "DCD FEC code type and modulation type"
13     REFERENCE
14         "Table 363 in IEEE Std 802.16e-2005"
15     SYNTAX          INTEGER {qpskCc1Over2 (0) ,
16                     qpskCc3Over4 (1) ,
17                     sixteenQamCc1Over2 (2) ,
18                     sixteenQamCc3Over4 (3) ,
19                     sixtyFourQamCc1Over2 (4) ,
20                     sixtyFourQamCc2Over3 (5) ,
21                     sixtyFourQamCc3Over4 (6) ,
22                     qpskBtc1Over2 (7) ,
23                     qpskBtc3Over4Or2Over3 (8) ,
24                     sixteenQamBtc3Over5 (9) ,
25                     sixteenQamBtc4Over5 (10) ,
26                     sixtyFourQamBtc2Over3Or5Over8 (11) ,
27                     sixtyFourQamBtc5Over6Or4Over5 (12) ,
28                     qpskCtc1Over2 (13) ,
29                     reserved14 (14) ,
30                     qpskCtc3Over4 (15) ,
31                     sixteenQamCtc1Over2 (16) ,
32                     sixteenQamCtc3Over4 (17) ,
33                     sixtyFourQamCtc1Over2 (18) ,
34                     sixtyFourQamCtc2Over3 (19) ,
35                     sixtyFourQamCtc3Over4 (20) ,
36                     sixtyFourQamCtc5Over6 (21) ,
37                     qpskZtCc1Over2 (22) ,
38                     qpskZtCc3Over4 (23) ,
39                     sixteenQamZtCc1Over2 (24) ,
40                     sixteenQamZtCc3Over4 (25) ,
41                     sixtyFourQamZtCc1Over2 (26) ,
42                     sixtyFourQamZtCc2Over3 (27) ,
43                     sixtyFourQamZtCc3Over4 (28) ,
44                     reserved29 (29) ,
45                     reserved30 (30) ,
46                     reserved31 (31) ,
47                     reserved32 (32) ,
48                     reserved33 (33) ,
49                     reserved34 (34) ,
50                     reserved35 (35) ,
51                     reserved36 (36) ,
52                     reserved37 (37) ,
53                     reserved38 (38) ,
54                     reserved39 (39) ,
55                     reserved40 (40) ,
56                     reserved41 (41) ,
57
58
59
60
61
62
63
64
65

```

```

1         reserved42(42),
2         reserved43(43),
3         qpskCcOptIntv1over2(44),
4         qpskCcOptIntv3over4(45),
5         sixteenQamCcOptIntv1over2(46),
6         sixteenQamCcOptIntv3over4(47),
7         sixtyFourQamCcOptIntv2over3(48),
8         sixtyFourQamCcOptIntv3over4(49) }
9
10
11
12 -- Textual convention for capabilities encodings
13 WmanIf2NumOfCid ::= TEXTUAL-CONVENTION
14     STATUS          current
15     DESCRIPTION
16         "The object of this type shows the number of CIDs that
17         SS can support."
18     REFERENCE
19         "Subclause 11.7.6 in IEEE Std 802.16e-2005"
20     SYNTAX          INTEGER (2..65535)
21
22
23
24 WmanIf2ArqSupportType ::= TEXTUAL-CONVENTION
25     STATUS          current
26     DESCRIPTION
27         "The object of this type indicates whether the SS support
28         ARQ."
29     REFERENCE
30         "Subclause 11.7.8.1 in IEEE Std 802.16-2004"
31     SYNTAX          INTEGER {arqNotSupported(0),
32                         arqSupported(1) }
33
34
35
36 WmanIf2MaxDsxFlowType ::= TEXTUAL-CONVENTION
37     STATUS          current
38     DESCRIPTION
39         "The object of this type specifies the maximum number of
40         concurrent DSA, DSC, or DSD transactions that may be
41         outstanding."
42     REFERENCE
43         "Subclause 11.7.8.2 in IEEE Std 802.16-2004"
44     SYNTAX          INTEGER (0..255)
45
46
47
48 WmanIf2MacCrcSupport ::= TEXTUAL-CONVENTION
49     STATUS          current
50     DESCRIPTION
51         "The object of this type indicates whether or not the SS
52         supports MAC level CRC."
53     REFERENCE
54         "Subclause 11.7.8.3 in IEEE Std 802.16-2004"
55     SYNTAX          INTEGER {noMacCrcSupport(0),
56                         macCrcSupport(1) }
57
58
59
60 WmanIf2MaxMcaFlowType ::= TEXTUAL-CONVENTION
61     STATUS          current
62     DESCRIPTION
63         "The object of this type specifies the maximum number of
64         concurrent MCA transactions that may be outstanding."
65

```

```

1      REFERENCE
2          "Subclause 11.7.8.4 in IEEE Std 802.16-2004"
3      SYNTAX      INTEGER (0..255)
4
5
6      WmanIf2MaxMcpGroupCid ::= TEXTUAL-CONVENTION
7          STATUS      current
8          DESCRIPTION
9              "The object of this type indicates the maximum number of
10             simultaneous Multicast Polling Groups the SS is
11             capable of belonging to."
12
13         REFERENCE
14             "Subclause 11.7.8.5 in IEEE Std 802.16-2004"
15         SYNTAX      INTEGER (0..255)
16
17
18         WmanIf2MaxPkmFlowType ::= TEXTUAL-CONVENTION
19             STATUS      current
20             DESCRIPTION
21                 "The object of this type specifies the maximum number of
22                 concurrent PKM transactions that may be outstanding."
23
24             REFERENCE
25                 "Subclause 11.7.8.6 in IEEE Std 802.16-2004"
26             SYNTAX      INTEGER (0..255)
27
28
29         WmanIf2AuthPolicyType ::= TEXTUAL-CONVENTION
30             STATUS      current
31             DESCRIPTION
32                 "The object of this type specifies authorization policy
33                 that both SS and BS need to negotiate and synchronize.
34                 A bit value of 0 = 'not supported', 1 = 'supported'. If
35                 this field is omitted, then both SS and BS shall use the
36                 IEEE 802.16 security, constituting X.509 digital
37                 certificates and the RSA public key encryption
38                 algorithm, as authorization policy."
39
40             REFERENCE
41                 "Subclause 11.7.8.7 in IEEE Std 802.16-2004"
42             SYNTAX      BITS {ieee802Dot16PrivacySupported(0),
43                             reserved1(1),
44                             reserved2(2),
45                             reserved3(3),
46                             reserved4(4),
47                             reserved5(5),
48                             reserved6(6),
49                             reserved7(7)}
50
51
52
53
54         WmanIf2MaxNumOfSaType ::= TEXTUAL-CONVENTION
55             STATUS      current
56             DESCRIPTION
57                 "This field specifies maximum number of supported
58                 security association of the SS."
59
60             REFERENCE
61                 "Subclause 11.7.8.8 in IEEE Std 802.16-2004"
62             SYNTAX      INTEGER (0..255)
63
64
65         WmanIf2IpVersionType ::= TEXTUAL-CONVENTION

```

```

1      STATUS      current
2      DESCRIPTION
3          "The object of this type indicates the version of IP used
4          on the Secondary Management Connection. The value should
5          be undefined if the 2nd management CID doesn't exist."
6      REFERENCE
7          "Subclause 11.7.4 in IEEE Std 802.16-2004"
8      SYNTAX      INTEGER {undefined(0),
9                  ipv4(1),
10                 ipv6(2)}
11
12
13
14 WmanIf2MacCsBitMap ::= TEXTUAL-CONVENTION
15     STATUS      current
16     DESCRIPTION
17         "The object of this type indicates the set of MAC
18         convergence sublayer support. When a bit is set, it
19         indicates the corresponding CS feature is supported."
20     REFERENCE
21         "Subclause 11.7.7.1 in IEEE Std 802.16e-2005"
22     SYNTAX      BITS {atm(0),
23                   packetIpv4(1),
24                   packetIpv6(2),
25                   packet802Dot3(3),
26                   packet802Dot1Q(4),
27                   packetIpv4Over802Dot3(5),
28                   packetIpv6Over802Dot3(6),
29                   packetIpv4Over802Dot1Q(7),
30                   packetIpv6Over802Dot1Q(8),
31                   packet802dot3EthernetRohcHc(9),
32                   packet802dot3EthernetEcrtPc(10),
33                   packetIpv4Orv6RohcHc(11),
34                   packetIpv4Orv6EcrtPc(12)}
35
36
37
38
39
40
41 WmanIf2MaxClassifiers ::= TEXTUAL-CONVENTION
42     STATUS      current
43     DESCRIPTION
44         "The object of this type indicates the maximum number of
45         admitted Classifiers that the SS is allowed to have."
46     REFERENCE
47         "Subclause 11.7.7.2 in IEEE Std 802.16-2004"
48     SYNTAX      INTEGER (0..65535)
49
50
51
52 WmanIf2PhsSupportType ::= TEXTUAL-CONVENTION
53     STATUS      current
54     DESCRIPTION
55         "The object of this type indicates the level
56         of PHS support."
57     REFERENCE
58         "Subclause 11.7.7.3 in IEEE Std 802.16e-2005"
59     SYNTAX      INTEGER {noPhsSupport(0),
60                   atmPhsSupport(1),
61                   packetPhsSupport(2),
62                   atmAndPacketPhsSupport(3)}
63
64
65

```



```

1  WmanIf2BwAllocSupport ::= TEXTUAL-CONVENTION
2      STATUS          current
3      DESCRIPTION
4          "This field indicates properties of the SS that the BS
5          needs to know for bandwidth allocation purposes. When
6          a bit is set, it indicates the corresponding feature
7          is supported. All unspecified and reserved bits should
8          be set to zero."
9
10     REFERENCE
11         "Subclause 11.8.1 in IEEE Std 802.16-2004"
12
13     SYNTAX          BITS {reserved(0),
14                       halfDuplexFdd(1),
15                       fullDuplexFdd(2)}
16
17
18  WmanIf2PduConstruction ::= TEXTUAL-CONVENTION
19      STATUS          current
20      DESCRIPTION
21          "Specifies capabilities for construction and transmission
22          of MAC PDUs. When piggybackedRequests bit is set, it
23          indicates that the piggybacked requests are supported. The
24          fsnValuesSize bit is coded as follows:
25          0 - only 3-bit FSN values are supported
26          1 - only 11-bit FSN values are supported
27          All unspecified and reserved bits should be set to zero."
28
29     REFERENCE
30         "Subclause 11.8.2 in IEEE Std 802.16e-2005"
31
32     SYNTAX          BITS {piggybackedRequests(0),
33                       fsnValuesSize(1)}
34
35
36  WmanIf2SsTransitionGap ::= TEXTUAL-CONVENTION
37      STATUS          current
38      DESCRIPTION
39          "This field indicates the transition speed SSTTG and SSRTG
40          for TDD and H-FDD SSs. Allowed values are:
41          OFDM mode: TDD and H-FDD 0..100
42          Other modes: TDD: 0..50; H-FDD: 0..100"
43
44     REFERENCE
45         "Subclause 11.8.3.1 in IEEE Std 802.16-2004"
46
47     SYNTAX          INTEGER (0..100)
48
49
50  WmanIf2MaxTxPowerType ::= TEXTUAL-CONVENTION
51      STATUS          current
52      DESCRIPTION
53          "This type is used to define maximum available power for
54          BPSK, QPSK, 16-QAM and 64-QAM constellations. The maximum
55          power parameters are reported in dBm and quantized in 0.5
56          dBm steps ranging from -64 dBm (encoded 0x00) to 63.5 dBm
57          (encoded 0xFF). Values outside this range shall be
58          assigned the closest extreme. SSs that do not support
59          QAM64 shall report the value of 0x00 in the maximum QAM64
60          power field."
61
62     REFERENCE
63         "Subclause 11.8.3.2 in IEEE Std 802.16-2004"
64
65     SYNTAX          INTEGER (0..255)

```

```

1
2 WmanIf2OfdmFftSizes ::= TEXTUAL-CONVENTION
3     STATUS          current
4     DESCRIPTION
5         "This field indicates the FFT sizes supported by the SS.
6         For each FFT size, a bit value of 0 indicates 'not
7         supported' while 1 indicates 'supported'."
8     REFERENCE
9         "Subclause 11.8.3.6.1 in IEEE 802.16-2004"
10    SYNTAX          BITS {fft256(0),
11                    fft2048(1)}
12
13 WmanIf2OfdmSsDeModType ::= TEXTUAL-CONVENTION
14     STATUS          current
15     DESCRIPTION
16         "This field indicates the different demodulator options
17         supported by a WirelessMAN-OFDM PHY SS for downlink. This
18         field is not used for other PHY specifications. A bit
19         value of 0 indicates 'not supported' while 1 indicates
20         'supported'."
21     REFERENCE
22         "Subclause 11.8.3.6.2 in IEEE Std 802.16e-2005"
23    SYNTAX          BITS {qam64(0),
24                    btc(1),
25                    ctc(2),
26                    stc(3),
27                    aas(4),
28                    subchannelization(5)}
29
30 WmanIf2OfdmSsModType ::= TEXTUAL-CONVENTION
31     STATUS          current
32     DESCRIPTION
33         "This field indicates the different modulator options
34         supported by a WirelessMAN-OFDM PHY SS for uplink. This
35         field is not used for other PHY specifications. A bit
36         value of 0 indicates 'not supported' while 1 indicates
37         'supported'."
38     REFERENCE
39         "Subclause 11.8.3.6.3 in IEEE Std 802.16e-2005"
40    SYNTAX          BITS {qam64(0),
41                    btc(1),
42                    ctc(2),
43                    subchanellization(3),
44                    focusedCtBwReq(4),
45                    ulCyclicDelay(5)}
46
47 WmanIf2OfdmFocusedCt ::= TEXTUAL-CONVENTION
48     STATUS          current
49     DESCRIPTION
50         "This field indicates whether the SS supports Focused
51         Contention (see 8.3.7.3.3). A bit value of 0 indicates
52         'not supported' while 1 indicates 'supported'."
53     REFERENCE
54         "Subclause 11.8.3.6.4 in IEEE Std 802.16-2004"
55
56
57
58
59
60
61
62
63
64
65

```

```

1          SYNTAX          BITS {focusedCtSupport(0)}
2
3
4 WmanIf2OfdmTcSublayer ::= TEXTUAL-CONVENTION
5     STATUS              current
6     DESCRIPTION
7         "This field indicates whether or not the SS supports the
8         TC sublayer (see 8.3.4). A bit value of 0 indicates
9         'not supported' while 1 indicates 'supported'."
10
11    REFERENCE
12        "Subclause 11.8.3.6.5 in IEEE Std 802.16-2004"
13    SYNTAX              BITS {tcSublayerSupport(0)}
14
15 WmanIf2OfdmPrivMap ::= TEXTUAL-CONVENTION
16     STATUS              current
17     DESCRIPTION
18         "This field indicates if the private map parameters
19         is supported. A bit value of 0 indicates
20         'not supported' while 1 indicates 'supported'."
21
22    REFERENCE
23        "Subclause 11.8.3.6.6 in IEEE Std 802.16e-2005"
24    SYNTAX              BITS {regularMap(0),
25                             compressedMap(1)}
26
27
28 WmanIf2OfdmUlPower ::= TEXTUAL-CONVENTION
29     STATUS              current
30     DESCRIPTION
31         "This field indicates the uplink power control options
32         supported by a WirelessMAN-OFDM PHY SS for uplink
33         transmission. A bit value of 0 indicates
34         'not supported' while 1 indicates 'supported'."
35
36    REFERENCE
37        "Subclause 11.8.3.7.10 in IEEE Std 802.16e-2005"
38    SYNTAX              BITS {ulOpenLoopPwrCntl(0),
39                             ulAasPreamblePwrCntl(1)}
40
41
42 WmanIf2BsIdType ::= TEXTUAL-CONVENTION
43     STATUS              current
44     DESCRIPTION
45         "Defines the encoding of BSID. The BSID is a 6 byte number
46         and follows the encoding rules of MacAddress textual
47         convention, i.e. as if it were transmitted
48         least-significant bit first. The value should be displayed
49         with 2 parts clearly separated by a colon e.g:
50         001DFF:00003A. The most significant part is representing
51         the Operator ID. "
52
53    SYNTAX              OCTET STRING (SIZE(6))
54
55
56 WmanIf2Ipv6FlowLabel ::= TEXTUAL-CONVENTION
57     STATUS              current
58     DESCRIPTION
59         "The value of this field specifies the matching values for
60         the IPv6 Flow label field. As the flow label field has a
61         length of 20 bits, the first 4 bits of the most
62         significant byte shall be set to 0x0 and disregarded."
63
64
65

```

```

1          SYNTAX          OCTET STRING (SIZE(3))
2
3
4 WmanIf2OfdmaFftSizes ::= TEXTUAL-CONVENTION
5     STATUS          current
6     DESCRIPTION
7         "This field indicates the FFT sizes supported by the SS/MS.
8         For each FFT size, a bit value of 0 indicates 'not
9         supported' while 1 indicates 'supported'."
10
11    REFERENCE
12        "Subclause 11.8.3.7.1 in IEEE 802.16e-2005"
13    SYNTAX          BITS {fft256(0),
14                      fft2048(1),
15                      fft128(2),
16                      fft512(3),
17                      fft1024(4)}
18
19
20 WmanIf2OfdmaMsDeModType ::= TEXTUAL-CONVENTION
21     STATUS          current
22     DESCRIPTION
23         "This field indicates the different demodulator options
24         supported by a WirelessMAN-OFDMA PHY SS for downlink.
25         A bit value of 0 indicates 'not supported' while 1
26         indicates 'supported'."
27
28    REFERENCE
29        "Subclause 11.8.3.7.2 in IEEE Std 802.16e-2005"
30    SYNTAX          BITS {qam64(0),
31                      btc(1),
32                      ctc(2),
33                      stc(3),
34                      ccWithInterleacer(4),
35                      harqChase(5),
36                      harqCtcIr(6),
37                      reserved(7),
38                      harqCcIr(8),
39                      ldpc(9),
40                      dedicatedPilots(10)}
41
42
43
44
45 WmanIf2OfdmaMsModType ::= TEXTUAL-CONVENTION
46     STATUS          current
47     DESCRIPTION
48         "This field indicates the different modulator options
49         supported by a WirelessMAN-OFDMA PHY SS for uplink. A bit
50         value of 0 indicates 'not supported' while 1 indicates
51         'supported'."
52
53    REFERENCE
54        "Subclause 11.8.3.7.3 in IEEE Std 802.16e-2005"
55    SYNTAX          BITS {qam64(0),
56                      btc(1),
57                      ctc(2),
58                      stc(3),
59                      harqChase(4),
60                      ctcIr(5),
61                      ccIr(6),
62                      ldpc(7)}
63
64
65

```

```

1
2 WmanIf2OfdmaPermutation ::= TEXTUAL-CONVENTION
3     STATUS          current
4     DESCRIPTION
5         "This field indicates the OFDMA SS Permutation support
6         A bit value of 0 indicates 'not supported' while 1
7         indicates 'supported'."
8     REFERENCE
9         "Subclause 11.8.3.7.4 in IEEE 802.16e"
10    SYNTAX          BITS {optionalPuscSupport(0),
11                    optionalFuscSupport(1),
12                    amcOneBySixSupport(2),
13                    amcTwoByThreeSupport(3),
14                    amcThreeByTwoSupport(4),
15                    amcSupportWithHarqMap(5),
16                    tusclSupport(6),
17                    tusc2Support(7)}
18
19 WmanIf2OfdmaDemMimo ::= TEXTUAL-CONVENTION
20     STATUS          current
21     DESCRIPTION
22         "This field indicates the MIMO capability of OFDMA MS
23         demodulator. A bit value of 0 indicates 'not supported'
24         while 1 indicates 'supported'."
25     REFERENCE
26         "Subclause 11.8.3.7.5 in IEEE 802.16e"
27    SYNTAX          BITS {twoAntStcMatrixA(0),
28                    twoAntStcMatrixBVCoding(1),
29                    twoAntStcMatrixBHCoding(2),
30                    fourAntStcMatrixA(3),
31                    fourAntStcMatrixBVCoding(4),
32                    fourAntStcMatrixBHCoding(5),
33                    fourAntStcMatrixCVCoding(6),
34                    fourAntStcMatrixCHCodingt(7)}
35
36 WmanIf2OfdmaMimoCap ::= TEXTUAL-CONVENTION
37     STATUS          current
38     DESCRIPTION
39         "This field indicates the MIMO capability of
40         OFDMA MS demodulator."
41     REFERENCE
42         "Subclause 11.8.3.7.5 in IEEE 802.16e"
43    SYNTAX          BITS {twoAntStcMatrixA(0),
44                    twoAntStcMatrixBVCoding(1),
45                    fourRxAntenna(2),
46                    fourAntStcMatrixA(3),
47                    fourAntStcMatrixBVCoding(4),
48                    fourAntStcMatrixBHCoding(5),
49                    fourAntStcMatrixCVCoding(6),
50                    fourAntStcMatrixCHCodingt(7),
51                    threeAntStcMatrixA(8),
52                    threeAntStcMatrixB(9),
53                    threeAntStcMatrixCVCoding(10),
54                    threeAntStcMatrixCHCodingt(11),
55
56
57
58
59
60
61
62
63
64
65

```

```

1         calculatingPrecodingWeight(12),
2         adaptiveRateControl(13),
3         calculatingChannelMatrix(14),
4         antennaGrouping(15),
5         antennaSelection(16),
6         codebookBasedPrecoding(17),
7         longTermPrecoding(18),
8         mimoMidamble(19) }
9
10
11
12 WmanIf2OfdmaUlMimo ::= TEXTUAL-CONVENTION
13     STATUS         current
14     DESCRIPTION
15         "This field indicates the different MIMO options supported
16         by a WirelessMAN-OFDMA PHY SS in the uplink.
17         A bit value of 0 indicates 'not supported' while 1
18         indicates 'supported'."
19     REFERENCE
20         "Subclause 11.8.3.7.6 in IEEE 802.16e"
21     SYNTAX         BITS {twoAntSttd(0),
22                   twoAntSmVCoding(1),
23                   oneAntCooperativeSm(2) }
24
25
26
27 WmanIf2OfdmaPrivMap ::= TEXTUAL-CONVENTION
28     STATUS         current
29     DESCRIPTION
30         "This field indicates the AAS private map parameters
31         supported by a WirelessMAN-OFDMA SS. A bit value of
32         0 indicates 'not supported' while 1 indicates
33         'supported' for most bits, except chainConcurrency0,
34         chainConcurrency1 that indicates how many parallel
35         private map chains can be supported by an SS.
36         0:    no limit
37         1..3: maximum concurrent private map chains"
38     REFERENCE
39         "Subclause 11.8.3.7.7 in IEEE Std 802.16e-2005"
40     SYNTAX         BITS {harqMap(0),
41                   privMap(1),
42                   reducedPrivMap(2),
43                   privMapChainEnable(3),
44                   privMapDlFrameOffset(4),
45                   privMapUlFrameOffset(5),
46                   chainConcurrency0(6),
47                   chainConcurrency1(7) }
48
49
50
51
52
53
54 WmanIf2OfdmaAasCap ::= TEXTUAL-CONVENTION
55     STATUS         current
56     DESCRIPTION
57         "This field indicates the different AAS options
58         supported by a WirelessMAN-OFDMA PHY SS in the
59         downlink. A bit value of 0 indicates 'not supported'
60         while 1 indicates 'supported' for most bits."
61     REFERENCE
62         "Subclause 11.8.3.7.8 in IEEE Std 802.16e-2005"
63     SYNTAX         BITS {aasZone(0),
64
65

```

```

1         aasDiversityMapScan(1),
2         aasFbckRsp(2),
3         dlAasPreamble(3),
4         ulAasPreamble(4) }
5
6
7 WmanIf2OfdmaCinrCap ::= TEXTUAL-CONVENTION
8     STATUS          current
9     DESCRIPTION
10        "This field indicates the CINR measurement capability
11        supported by a WirelessMAN-OFDMA PHY SS in the
12        downlink. A bit value of 0 indicates 'not supported'
13        while 1 indicates 'supported'."
14    REFERENCE
15        "Subclause 11.8.3.7.9 in IEEE Std 802.16e-2005"
16    SYNTAX          BITS {phyCinrPreamble(0),
17                    phyCinrPilotSubc(1),
18                    phyCinrDataSubc(2),
19                    effectiveCinrPreamble(3),
20                    effectiveCinrPilotSubc(4),
21                    effectiveCinrDataSubc(5),
22                    twoCqiChannel(6),
23                    freqSelectivityReport(7) }
24
25
26
27
28
29 WmanIf2OfdmaUlPower ::= TEXTUAL-CONVENTION
30     STATUS          current
31     DESCRIPTION
32        "This field indicates the power control options
33        supported by a WirelessMAN-OFDMA PHY SS for uplink
34        transmission. A bit value of 0 indicates
35        'not supported' while 1 indicates 'supported'."
36    REFERENCE
37        "Subclause 11.8.3.7.11 in IEEE Std 802.16e-2005"
38    SYNTAX          BITS {ulOpenLoopPwrCntl(0),
39                    ulAasPreamblePwrCntl(1) }
40
41
42
43
44 WmanIf2OfdmaMapCap ::= TEXTUAL-CONVENTION
45     STATUS          current
46     DESCRIPTION
47        "This field indicates the different MAP options supported
48        by a WirelessMAN-OFDMA PHY SS. A bit value of 0
49        indicates 'not supported' while 1 indicates 'supported'."
50    REFERENCE
51        "Subclause 11.8.3.7.12 in IEEE Std 802.16e-2005"
52    SYNTAX          BITS {harqMap(0),
53                    extendedHarqIe(1),
54                    subMapFor1stZone(2),
55                    subMapForOtherZone(3),
56                    dlRegionDefinition(4) }
57
58
59
60 WmanIf2OfdmaUlCntlCh ::= TEXTUAL-CONVENTION
61     STATUS          current
62     DESCRIPTION
63        "This field indicates different uplink control channels
64        supported by a WirelessMAN-OFDMA PHY SS. A bit value
65

```

```

1         of 0 indicates 'not supported' while 1 indicates
2         'supported'."
3     REFERENCE
4         "Subclause 11.8.3.7.13 in IEEE Std 802.16e-2005"
5     SYNTAX     BITS {threeBitMimoFastFeedback(0),
6                 enhancedFastFeedback(1),
7                 ulAck(2),
8                 reserved(3),
9                 uepFastFeedback(4),
10                fastDlMeasurementFeedback(5),
11                priSecFastFeedback(6),
12                diucCqiFastFeedback(7)}
13
14
15
16 WmanIf2OfdmaMsCistCap ::= TEXTUAL-CONVENTION
17     STATUS     current
18     DESCRIPTION
19         "This field indicates MS capability of supporting CSIT
20         (uplink sounding). A bit value of 0 indicates 'not
21         supported' while 1 indicates 'supported'."
22
23
24
25         Bits 3..5: Time needed for SS to respond to a sounding
26         command transmitted by the BS
27
28             000    0.5ms
29             001    0.75ms
30             010    1ms
31             011    1.25ms
32             100    1.5ms
33             101    min(2ms, Next Frame)
34             110    min(5ms, Next Frame)
35             111    Next Frame
36
37
38         Bits 6..9: Max number of simultaneous sounding
39         instructions (0 = unlimited)"
40
41     REFERENCE
42         "Subclause 11.8.3.7.14 in IEEE Std 802.16e-2005"
43     SYNTAX     BITS {csitTypeA(0),
44                 csitTypeB(1),
45                 powerAssignment(2),
46                 soundingRspTime0(3),
47                 soundingRspTime1(4),
48                 soundingRspTime2(5),
49                 maxSimuSoundInst0(6),
50                 maxSimuSoundInst1(7),
51                 maxSimuSoundInst2(8),
52                 maxSimuSoundInst3(9),
53                 noP9Or18ForCsitTypeA(10),
54                 csitNotSupported(11)}
55
56
57
58 WmanIf2OfdmaMaxHarq ::= TEXTUAL-CONVENTION
59     STATUS     current
60     DESCRIPTION
61         "This field indicates the maximum number of UL/DL HARQ
62         burst allocations for the SS in a single UL/DL subframe."
63
64
65

```



```

1       Bits 0..2: Maximum number of UL HARQ bursts per HARQ
2         enabled MS per frame
3         0b000 = 1 (default)
4
5
6       Bit      3: Indicates whether the maximum number of UL
7         HARQ bursts per frame in bits 0-2 includes the
8         one Non-HARQ burst.
9         0 = not included (default)
10        1 = included
11
12
13      Bits 4..7: Maximum number of DL HARQ bursts per HARQ
14        enabled MS per frame.
15        0b0000 = 1 (default)"
16
17      REFERENCE
18        "Subclause 11.8.3.7.15 in IEEE Std 802.16e-2005"
19      SYNTAX      BITS {maxUlHarqBurst0(0),
20                    maxUlHarqBurst1(1),
21                    maxUlHarqBurst2(2),
22                    nonHarqBurstInUl(3),
23                    maxDlHarqBurst0(4),
24                    maxDlHarqBurst1(5),
25                    maxDlHarqBurst2(6),
26                    maxDlHarqBurst3(7)}
27
28
29
30      WmanIf2OfdmaModMimo ::= TEXTUAL-CONVENTION
31        STATUS      current
32        DESCRIPTION
33          "This field indicates the MIMO capability of OFDMA SS
34          modulator. A bit value of 0 indicates 'not supported'
35          while 1 indicates 'supported'"
36        REFERENCE
37          "Subclause 11.8.3.7.16 in IEEE Std 802.16e-2005"
38        SYNTAX      BITS {twoTxAntenna(0),
39                    txDiversity(1),
40                    spatialMultiplexing(2),
41                    beamforming(3),
42                    adaptiveRateControl(4),
43                    singleAntenna(5),
44                    twoAntenna(6)}
45
46
47
48
49      WmanIf2SdmaPilotCap ::= TEXTUAL-CONVENTION
50        STATUS      current
51        DESCRIPTION
52          "This field indicates SDMA pilot pattern support for
53          AMC zone."
54        REFERENCE
55          "Subclause 11.8.3.7.17 in IEEE Std 802.16e-2005"
56        SYNTAX      INTEGER {noSupport(0),
57                    sdmaPilotAandB(1),
58                    allSdmaPilotPatterns(2)}
59
60
61
62      WmanIf2MultiBurst ::= TEXTUAL-CONVENTION
63        STATUS      current
64        DESCRIPTION
65

```

```

1         "This field indicates whether multiple FEC types are
2         supported in DL/UL burst profiles. A bit value of 0
3         indicates 'not supported' while 1 indicates
4         'supported'"
5
6     REFERENCE
7         "Subclause 11.8.3.7.18 in IEEE Std 802.16e-2005"
8
9     SYNTAX      INTEGER {dlWithMultiFecType(0),
10                ulWithMultiFecType(1)}
11
12 WmanIf2IncrHarqBuf ::= TEXTUAL-CONVENTION
13     STATUS      current
14
15     DESCRIPTION
16         "This field indicates the maximal number of data
17         bits the SS is able to use for buffering for NEP/NSCH
18         based incremental redundancy CTC in downlink and uplink
19         transmissions.
20
21         Bits 0..3: NEP value indicating downlink HARQ buffering
22                 capability for incremental redundancy CTC
23
24         Bit      4: Aggregation Flag for DL
25                 0 = the number of bits is counted separately
26                 for each channel
27                 1 = buffering capability may be shared between
28                 channels
29
30         Bits 5..7: reserved
31
32         Bits 8..11: NEP value indicating uplink HARQ buffering
33                  capability for incremental redundancy CTC
34
35         Bit      12: Aggregation Flag for UL
36                 0 = the number of bits is counted separately
37                 for each channel
38                 1 = buffering capability may be shared between
39                 channels"
40
41     REFERENCE
42         "Subclause 11.8.3.7.19 in IEEE Std 802.16e-2005"
43
44     SYNTAX      INTEGER {dlNep0(0),
45                dlNep1(1),
46                dlNep2(2),
47                dlNep3(3),
48                dlAggFlag(4),
49                reserved0(5),
50                reserved1(6),
51                reserved2(7),
52                ulNep0(8),
53                ulNep1(9),
54                ulNep2(10),
55                ulNep3(11),
56                ulAggFlag(12)}
57
58 WmanIf2ChaseHarqBuf ::= TEXTUAL-CONVENTION
59     STATUS      current
60
61
62
63
64
65

```

```

1      DESCRIPTION
2          "This field indicates the maximal number of data
3          bits the SS is able to use for buffering for
4          DIUC/duration based HARQ methods (Chase combining and
5          CC-IR) in downlink and uplink transmissions.
6
7
8          Bits 0..5: Downlink HARQ buffering capability for
9                      chase combining (K)
10
11
12         Bit      6: Aggregation Flag for DL
13                   0 = the number of bits is counted separately
14                   for each channel
15                   1 = buffering capability may be shared between
16                   channels
17
18
19         Bits     7: reserved
20
21
22         Bits 8..13: Uplink HARQ buffering capability for chase
23                     combining (K)
24
25         Bit     14: Aggregation Flag for UL
26                   0 = the number of bits is counted separately
27                   for each channel
28                   1 = buffering capability may be shared between
29                   channels"
30
31     REFERENCE
32         "Subclause 11.8.3.7.19 in IEEE Std 802.16e-2005"
33     SYNTAX     INTEGER {dlChaseComb0(0),
34                   dlChaseComb1(1),
35                   dlChaseComb2(2),
36                   dlChaseComb3(3),
37                   dlChaseComb4(4),
38                   dlChaseComb5(5),
39                   dlAggFlag(6),
40                   reserved(7),
41                   ulChaseComb0(8),
42                   ulChaseComb1(9),
43                   ulChaseComb2(10),
44                   ulChaseComb3(11),
45                   ulChaseComb4(12),
46                   ulChaseComb5(13),
47                   ulAggFlag(14)}
48
49
50
51
52     WmanIf2PackingSupport ::= TEXTUAL-CONVENTION
53         STATUS          current
54         DESCRIPTION
55             "Indicates the availability of MS support for Packing"
56         REFERENCE
57             "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
58         SYNTAX          INTEGER {noPackingSupport(0),
59                   packingSupported(1)}
60
61
62
63     WmanIf2ExtRtpsSupport ::= TEXTUAL-CONVENTION
64         STATUS          current
65

```

```

1      DESCRIPTION
2          "Indicates the availability of MS support for Extended
3          rtPS."
4
5      REFERENCE
6          "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
7      SYNTAX      INTEGER {noExtendedRtpsSupport(0),
8                  extendedRtpsSupported(1)}
9
10
11 WmanIf2IpAllocMethod ::= TEXTUAL-CONVENTION
12     STATUS      current
13     DESCRIPTION
14         "Indicates the method of allocating IP address for the
15         secondary management connection. A bit value of 0
16         indicates 'not supported' while 1 indicates 'supported'."
17     REFERENCE
18         "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
19     SYNTAX      BITS {dhcp(0),
20                  mobileIpv4(1),
21                  dhcpV6(2),
22                  ipv6Autoconfig(3)}
23
24
25
26 WmanIf2ArqAckType ::= TEXTUAL-CONVENTION
27     STATUS      current
28     DESCRIPTION
29         "Specifies the ARQ ACK type supported by the MS."
30     REFERENCE
31         "Subclause 11.7.23 in IEEE Std 802.16e-2005"
32     SYNTAX      BITS {selectiveAck(0),
33                  cumulativeAck(1),
34                  cumWithSelAck(2),
35                  cumWithBlockSeqAck(3)}
36
37
38
39 WmanIf2MacHeaderSupp ::= TEXTUAL-CONVENTION
40     STATUS      current
41     DESCRIPTION
42         "Indicates whether or not the MS and BS support various
43         types of MAC header and extended subheaders. A bit
44         value of 0 indicates 'not supported', while 1 indicates
45         'supported'.
46
47         Bits 8-10: parameters of SDU_SN extended subheader that
48         represent the period of SDU_SN transmission for
49         connection with ARQ disabled = once every 2^p MAC
50         PDUs."
51     REFERENCE
52         "Subclause 11.7.25 in IEEE Std 802.16e-2005"
53     SYNTAX      BITS {bwReqUltTxPowerReport(0),
54                  bwReqCinrReport(1),
55                  cqichAlloationReq(2),
56                  phyChannelReport(3),
57                  bwReqUlSleepCntl(4),
58                  snReport(5),
59                  feedbackReport(6),
60                  sduSn(7),
61
62
63
64
65

```

```

1         sdnSnPeriod0(8),
2         sdnSnPeriod1(9),
3         sdnSnPeriod2(10),
4         dlSleepControl(11),
5         feedbackRequest(12),
6         mimcModeFeedback(13),
7         ulTxPowerReport(14),
8         miniFeedback(15),
9         snRequest(16),
10        shortPduSn(17),
11        longPduSn(18) }
12
13
14
15 WmanIf2HarqAckDelay ::= TEXTUAL-CONVENTION
16     STATUS          current
17     DESCRIPTION
18         "HARQ ACK delay for UL and DL bursts
19         1 = one frame offset
20         2 = two frames offset
21         3 = three frames offset"
22     REFERENCE
23         "Table 353 in IEEE Std 802.16e-2005"
24     SYNTAX          INTEGER {oneframeoffset(1),
25                    twoframesoffset(2),
26                    threeframesoffset(3) }
27
28
29
30
31 WmanIf2AasBeamSel ::= TEXTUAL-CONVENTION
32     STATUS          current
33     DESCRIPTION
34         "Boolean to indicate whether unsolicited AAS Beam Select
35         messages (see 6.3.2.3.41 in IEEE 802.16e-2005) should be
36         sent by the MS.
37         0: MS should not send AAS Beam Select Messages
38         1: MS may send AAS Beam Select Messages"
39     REFERENCE
40         "Table 353 in IEEE Std 802.16e-2005"
41     SYNTAX          INTEGER {notAllowed(0),
42                    allowed(1) }
43
44
45
46
47 WmanIf2MaxMacLevel ::= TEXTUAL-CONVENTION
48     STATUS          current
49     DESCRIPTION
50         "maximum amount of MAC level data including MAC headers
51         and HARQ retransmission bursts the MS is capable of
52         processing in the DL/UL part of a single MAC frame."
53     REFERENCE
54         "Subclause 11.7.8.10 in IEEE Std 802.16e-2005"
55     SYNTAX          INTEGER (0..65535)
56
57
58
59 WmanIfPermutationType ::= TEXTUAL-CONVENTION
60     STATUS          current
61     DESCRIPTION
62         "Permutation type for broadcast region in HARQ zone"
63     REFERENCE
64         "Table 353 in IEEE Std 802.16e-2005"
65

```

```

1      SYNTAX      INTEGER {pusc(1),
2                    fusc(2),
3                    optionalFusc(3),
4                    amc(4)}
5
6
7      WmanIf2HoSupportType ::= TEXTUAL-CONVENTION
8          STATUS      current
9          DESCRIPTION
10             "The types of handover supported."
11          REFERENCE
12             "Table 358 in IEEE Std 802.16e-2005"
13          SYNTAX      BITS {handover(0),
14                          mdHandover(1),
15                          fbssHandover(2)}
16
17
18
19      --
20      -- BS object group - containing tables and objects to be implemented in
21      -- the Base station
22      --
23      --
24      -- wmanIf2BsPacketCs contain the Base Station Packet Convergence
25      -- Sublayer objects
26      --
27      wmanIf2BsPacketCs OBJECT IDENTIFIER ::= { wmanIf2BsObjects 1 }
28
29
30      wmanIf2BsProvisionedSfTable OBJECT-TYPE
31          SYNTAX      SEQUENCE OF WmanIf2BsProvisionedSfEntry
32          MAX-ACCESS  not-accessible
33          STATUS      current
34          DESCRIPTION
35             "This table contains service flow profiles provisioned by
36             NMS. The service flow should be created with SS(s)
37             following instruction given by wmanIf2BsSfState object.
38             1. The QoS parameters of the service flow are provisioned
39             in wmanIf2BsServiceClassTable and referenced by
40             wmanIf2BsServiceClassIndex.
41             2. The classifier rules of the service flow are provisioned
42             in wmanIf2BsClassifierRuleTable, where they refer to SF
43             via wmanIf2BsSfId.
44
45             The MAC addresses of SSs the service flow is created with
46             are provisioned in wmanIf2BsSsProvisionedForSfTable, where
47             they refer to SF via wmanIf2BsSfId."
48          REFERENCE
49             "Subclause 6.3.13 and 6.3.14 in IEEE Std 802.16-2004"
50          ::= { wmanIf2BsPacketCs 1 }
51
52
53
54      wmanIf2BsProvisionedSfEntry OBJECT-TYPE
55          SYNTAX      WmanIf2BsProvisionedSfEntry
56          MAX-ACCESS  not-accessible
57          STATUS      current
58          DESCRIPTION
59             "This table provides one row for each service flow
60             provisioned by NMS. The table is indexed by ifIndex and
61             wmanIf2BsSfId. ifIndex is associated with the BS sector."
62
63
64
65

```

```

1      INDEX { ifIndex, wmanIf2BsSfId }
2      ::= { wmanIf2BsProvisionedSfTable 1 }
3
4
5      WmanIf2BsProvisionedSfEntry ::= SEQUENCE {
6          wmanIf2BsSfId                Unsigned32,
7          wmanIf2BsSfDirection         INTEGER,
8          wmanIf2BsServiceClassIndex   INTEGER,
9          wmanIf2BsSfState              WmanIf2SfState,
10         wmanIf2BsSfProvisionedTime   TimeStamp,
11         wmanIf2BsSfCsSpecification   WmanIf2CsSpecification,
12         wmanIf2BsProvisionedSfRowStatus RowStatus}
13
14
15      wmanIf2BsSfId OBJECT-TYPE
16          SYNTAX      Unsigned32 (1 .. 4294967295)
17          MAX-ACCESS  not-accessible
18          STATUS      current
19          DESCRIPTION
20              "A 32 bit quantity that uniquely identifies a service flow
21              to both the subscriber station and base station (BS)."
```

```

22      ::= { wmanIf2BsProvisionedSfEntry 1 }
23
24
25
26      wmanIf2BsSfDirection OBJECT-TYPE
27          SYNTAX      INTEGER {downstream(1),
28                          upstream(2)}
29          MAX-ACCESS  read-create
30          STATUS      current
31          DESCRIPTION
32              "An attribute indicating the service flow is downstream or
33              upstream."
```

```

34      ::= { wmanIf2BsProvisionedSfEntry 2 }
35
36
37
38      wmanIf2BsServiceClassIndex OBJECT-TYPE
39          SYNTAX      INTEGER (1..65535)
40          MAX-ACCESS  read-create
41          STATUS      current
42          DESCRIPTION
43              "The index in wmanIf2BsServiceClassTable describing the
44              service class or QoS parameters for such service flow.
45              If no associated entry in wmanIf2BsServiceClassTable
46              exists, this object returns a value of zero."
```

```

47      ::= { wmanIf2BsProvisionedSfEntry 3 }
48
49
50
51      wmanIf2BsSfState OBJECT-TYPE
52          SYNTAX      WmanIf2SfState
53          MAX-ACCESS  read-create
54          STATUS      current
55          DESCRIPTION
56              "wmanIf2BsSfState determines the requested state of a service
57              flow.
58              - authorized state: A service flow is provisioned but
59              not resource is reserved yet
60              - admitted state: service flow has resources reserved.
61              - active state: has resources committed by the BS (e.g., is
62              actively sending maps containing unsolicited grants for a
63
64
65
```

```

1         UGS-based service flow),"
2     REFERENCE
3         "Subclause 6.3.14.6, in IEEE Std 802.16-2004"
4     ::= { wmanIf2BsProvisionedSfEntry 4 }
5
6
7     wmanIf2BsSfProvisionedTime OBJECT-TYPE
8         SYNTAX      TimeStamp
9         MAX-ACCESS  read-create
10        STATUS      current
11        DESCRIPTION
12            "Indicates the date and time when the service flow is
13             provisioned."
14        ::= { wmanIf2BsProvisionedSfEntry 5 }
15
16
17
18    wmanIf2BsSfCsSpecification OBJECT-TYPE
19        SYNTAX      WmanIf2CsSpecification
20        MAX-ACCESS  read-create
21        STATUS      current
22        DESCRIPTION
23            "This parameter specifies the convergence sublayer
24             encapsulation mode."
25        REFERENCE
26            "Subclause 11.13.19.1 in IEEE Std 802.16-2004"
27        ::= { wmanIf2BsProvisionedSfEntry 6 }
28
29
30
31    wmanIf2BsProvisionedSfRowStatus OBJECT-TYPE
32        SYNTAX      RowStatus
33        MAX-ACCESS  read-create
34        STATUS      current
35        DESCRIPTION
36            "This object is used to create a new row or modify or
37             delete an existing row in this table.
38
39             If the implementator of this MIB has choosen not
40             to implement 'dynamic assignment' of profiles, this
41             object is not useful and should return noSuchName
42             upon SNMP request."
43        ::= { wmanIf2BsProvisionedSfEntry 7 }
44
45
46
47
48    wmanIf2BsSsProvisionedForSfTable OBJECT-TYPE
49        SYNTAX      SEQUENCE OF WmanIf2BsSsProvisionedForSfEntry
50        MAX-ACCESS  not-accessible
51        STATUS      current
52        DESCRIPTION
53            "This table maps the MAC addresses of SSs to the service
54             flows provisioned in wmanIf2BsProvisionedSfTable."
55        REFERENCE
56            "Subclause 6.3.14 in IEEE Std 802.16-2004"
57        ::= { wmanIf2BsPacketCs 2 }
58
59
60
61    wmanIf2BsSsProvisionedForSfEntry OBJECT-TYPE
62        SYNTAX      WmanIf2BsSsProvisionedForSfEntry
63        MAX-ACCESS  not-accessible
64        STATUS      current
65

```



```

1      DESCRIPTION
2          "This table is indexed by wmanIf2BsSsProvMacAddress and
3          wmanIf2BsProvSfId."
4      INDEX { wmanIf2BsSsProvMacAddress, wmanIf2BsProvSfId }
5      ::= { wmanIf2BsSsProvisionedForSfTable 1 }
6
7
8      WmanIf2BsSsProvisionedForSfEntry ::= SEQUENCE {
9          wmanIf2BsSsProvMacAddress      MacAddress,
10         wmanIf2BsProvSfId              Unsigned32,
11         wmanIf2BsSsProvisionedForSfRowStatus  RowStatus}
12
13
14     wmanIf2BsSsProvMacAddress OBJECT-TYPE
15         SYNTAX      MacAddress
16         MAX-ACCESS  not-accessible
17         STATUS      current
18         DESCRIPTION
19             "The MAC address of the SS, the service flow is created
20             with."
21         ::= { wmanIf2BsSsProvisionedForSfEntry 1 }
22
23
24
25     wmanIf2BsProvSfId OBJECT-TYPE
26         SYNTAX      Unsigned32 (1 .. 4294967295)
27         MAX-ACCESS  not-accessible
28         STATUS      current
29         DESCRIPTION
30             "A 32 bit quantity that uniquely identifies a service flow.
31             The value of this object can be used by BS to index the
32             wmanIf2BsProvisionedSfTable."
33         ::= { wmanIf2BsSsProvisionedForSfEntry 2 }
34
35
36
37     wmanIf2BsSsProvisionedForSfRowStatus OBJECT-TYPE
38         SYNTAX      RowStatus
39         MAX-ACCESS  read-create
40         STATUS      current
41         DESCRIPTION
42             "This object is used to ensure that the write, create,
43             delete operation to multiple columns is guaranteed to
44             be treated as atomic operation by agent."
45         ::= { wmanIf2BsSsProvisionedForSfEntry 3 }
46
47
48
49     wmanIf2BsServiceClassTable OBJECT-TYPE
50         SYNTAX      SEQUENCE OF WmanIf2BsServiceClassEntry
51         MAX-ACCESS  not-accessible
52         STATUS      current
53         DESCRIPTION
54             "This table is provisioned and is indexed by
55             wmanIf2BsQoSProfileIndex. Each entry of the table contains
56             corresponding service flow characteristic attributes
57             (e.g. QoS parameter set). The value of
58             wmanIf2BsQoSProfileIndex is obtained from
59             wmanIf2BsServiceClassIndex in wmanIf2BsProvisionedSfTable"
60         REFERENCE
61             "Subclause 6.3.14.4 in IEEE Std 802.16-2004"
62         ::= { wmanIf2BsPacketCs 3 }
63
64
65

```

```

1
2 wmanIf2BsServiceClassEntry OBJECT-TYPE
3     SYNTAX      WmanIf2BsServiceClassEntry
4     MAX-ACCESS  not-accessible
5     STATUS      current
6     DESCRIPTION
7         "This table provides one row for each service class"
8     INDEX { ifIndex, wmanIf2BsQoSProfileIndex }
9     ::= { wmanIf2BsServiceClassTable 1 }
10
11
12
13 WmanIf2BsServiceClassEntry ::= SEQUENCE {
14     wmanIf2BsQoSProfileIndex          INTEGER,
15     wmanIf2BsQoSServiceClassName     WmanIf2ServClassName,
16     wmanIf2BsQoSSTrafficPriority      INTEGER,
17     wmanIf2BsQoSMaxSustainedRate     Unsigned32,
18     wmanIf2BsQoSMaxTrafficBurst      Unsigned32,
19     wmanIf2BsQoSMinReservedRate      Unsigned32,
20     wmanIf2BsQOSToleratedJitter      Unsigned32,
21     wmanIf2BsQoSMaxLatency           Unsigned32,
22     wmanIf2BsQoSFixedVsVariableSduInd INTEGER,
23     wmanIf2BsQoSsSduSize             Unsigned32,
24     wmanIf2BsQoSsScSchedulingType    WmanIf2SfSchedulingType,
25     wmanIf2BsQoSsScArqEnable         TruthValue,
26     wmanIf2BsQoSsScArqWindowSize    INTEGER,
27     wmanIf2BsQoSsScArqBlockLifetime  INTEGER,
28     wmanIf2BsQoSsScArqSyncLossTimeout INTEGER,
29     wmanIf2BsQoSsScArqDeliverInOrder TruthValue,
30     wmanIf2BsQoSsScArqRxPurgeTimeout INTEGER,
31     wmanIf2BsQoSsScArqBlockSize      INTEGER,
32     wmanIf2BsQoSSCMinRsvdTolerableRate Unsigned32,
33     wmanIf2BsQoSsReqTxPolicy         BITS,
34     wmanIf2BsQoSsServiceClassRowStatus RowStatus }
35
36
37
38
39
40
41 wmanIf2BsQoSProfileIndex OBJECT-TYPE
42     SYNTAX      INTEGER (1 .. 65535)
43     MAX-ACCESS  not-accessible
44     STATUS      current
45     DESCRIPTION
46         "The index value which uniquely identifies an entry
47         in the wmanIf2BsServiceClassTable"
48     ::= { wmanIf2BsServiceClassEntry 1 }
49
50
51
52 wmanIf2BsQoSServiceClassName OBJECT-TYPE
53     SYNTAX      WmanIf2ServClassName
54     MAX-ACCESS  read-create
55     STATUS      current
56     DESCRIPTION
57         "Refers to the Service Class Name"
58     REFERENCE
59         "Subclause 11.13.3 in IEEE Std 802.16-2004"
60     ::= { wmanIf2BsServiceClassEntry 2 }
61
62
63
64 wmanIf2BsQoSSTrafficPriority OBJECT-TYPE
65     SYNTAX      INTEGER (0..7)

```

```

1      MAX-ACCESS  read-create
2      STATUS      current
3      DESCRIPTION
4          "The value of this parameter specifies the priority
5           assigned to a service flow. For uplink service flows,
6           the BS should use this parameter when determining
7           precedence in request service and grant generation,
8           and the SS shall preferentially select contention
9           Request opportunities for Priority Request CIDs
10          based on this priority. Higher numbers indicate higher
11          priority"
12      REFERENCE
13          "Subclause 11.13.5 in IEEE Std 802.16-2004"
14      ::= { wmanIf2BsServiceClassEntry 3 }
15
16  wmanIf2BsQoSMaxSustainedRate OBJECT-TYPE
17      SYNTAX      Unsigned32
18      UNITS       "b/s"
19      MAX-ACCESS  read-create
20      STATUS      current
21      DESCRIPTION
22          "This parameter defines the peak information rate
23           of the service. The rate is expressed in bits per
24           second and pertains to the SDUs at the input to
25           the system."
26      REFERENCE
27          "Subclause 11.13.6 in IEEE Std 802.16-2004"
28      ::= { wmanIf2BsServiceClassEntry 4 }
29
30  wmanIf2BsQoSMaxTrafficBurst OBJECT-TYPE
31      SYNTAX      Unsigned32
32      UNITS       "byte"
33      MAX-ACCESS  read-create
34      STATUS      current
35      DESCRIPTION
36          "This parameter defines the maximum burst size that
37           must be accommodated for the service."
38      REFERENCE
39          "Subclause 11.13.7 in IEEE Std 802.16-2004"
40      ::= { wmanIf2BsServiceClassEntry 5 }
41
42  wmanIf2BsQoSMinReservedRate OBJECT-TYPE
43      SYNTAX      Unsigned32
44      UNITS       "b/s"
45      MAX-ACCESS  read-create
46      STATUS      current
47      DESCRIPTION
48          "This parameter specifies the minimum rate reserved
49           for this service flow."
50      REFERENCE
51          "Subclause 11.13.8 in IEEE Std 802.16-2004"
52      ::= { wmanIf2BsServiceClassEntry 6 }
53
54  wmanIf2BsQoSSToleratedJitter OBJECT-TYPE
55
56
57
58
59
60
61
62
63
64
65

```

```

1      SYNTAX      Unsigned32
2      UNITS       "millisecond"
3      MAX-ACCESS  read-create
4      STATUS      current
5      DESCRIPTION
6          "This parameter defines the Maximum delay
7          variation (jitter) for the connection."
8      REFERENCE
9          "Subclause 11.13.13 in IEEE Std 802.16-2004"
10     ::= { wmanIf2BsServiceClassEntry 7 }
11
12
13
14     wmanIf2BsQoSMaxLatency OBJECT-TYPE
15         SYNTAX      Unsigned32
16         UNITS       "millisecond"
17         MAX-ACCESS  read-create
18         STATUS      current
19         DESCRIPTION
20             "The value of this parameter specifies the maximum
21             latency between the reception of a packet by the BS
22             or SS on its network interface and the forwarding
23             of the packet to its RF Interface."
24         REFERENCE
25             "Subclause 11.13.14 in IEEE Std 802.16-2004"
26         ::= { wmanIf2BsServiceClassEntry 8 }
27
28
29
30
31     wmanIf2BsQoSFixedVsVariableSduInd OBJECT-TYPE
32         SYNTAX      INTEGER { variableLength(0),
33                       fixedLength(1) }
34         MAX-ACCESS  read-create
35         STATUS      current
36         DESCRIPTION
37             "The value of this parameter specifies whether the SDUs
38             on the service flow are variable-length (0) or
39             fixed-length (1). The parameter is used only if
40             packing is on for the service flow. The default value
41             is 0, i.e., variable-length SDUs."
42         REFERENCE
43             "Subclause 11.13.15 in IEEE Std 802.16-2004"
44         DEFVAL      { variableLength }
45         ::= { wmanIf2BsServiceClassEntry 9 }
46
47
48
49
50     wmanIf2BsQoSsduSize OBJECT-TYPE
51         SYNTAX      Unsigned32
52         UNITS       "byte"
53         MAX-ACCESS  read-create
54         STATUS      current
55         DESCRIPTION
56             "The value of this parameter specifies the length of the
57             SDU for a fixed-length SDU service flow. This parameter
58             is used only if packing is on and the service flow is
59             indicated as carrying fixed-length SDUs. The default
60             value is 49 bytes, i.e., VC-switched ATM cells with PHS.
61             The parameter is relevant for both ATM and Packet
62             Convergence Sublayers."
63
64
65

```

```

1      REFERENCE
2          "Subclause 11.13.16 in IEEE Std 802.16-2004"
3      DEFVAL      { 49 }
4      ::= { wmanIf2BsServiceClassEntry 10 }
5
6
7      wmanIf2BsQosScSchedulingType OBJECT-TYPE
8          SYNTAX      WmanIf2SfSchedulingType
9          MAX-ACCESS  read-create
10         STATUS      current
11         DESCRIPTION
12             "Specifies the upstream scheduling service used for
13             upstream service flow. If the referenced parameter
14             is not present in the corresponding 802.16 QOS
15             Parameter Set of an upstream service flow, the
16             default value of this object is bestEffort(2)."

```

```

1      REFERENCE
2          "Subclause 11.13.18 in IEEE Std 802.16-2004"
3      DEFVAL    {0}
4      ::= { wmanIf2BsServiceClassEntry 14 }
5
6
7      wmanIf2BsQosScArqSyncLossTimeout OBJECT-TYPE
8          SYNTAX      INTEGER (0 .. 65535 )
9          UNITS        "10 us"
10         MAX-ACCESS  read-create
11         STATUS      current
12         DESCRIPTION
13             "The maximum interval before declaring a loss
14             of synchronization of the sender and receiver
15             state machines. A value of 0 means Infinite."
16         REFERENCE
17             "Subclause 11.13.18 in IEEE Std 802.16-2004"
18         DEFVAL    {0}
19         ::= { wmanIf2BsServiceClassEntry 15 }
20
21
22
23
24         wmanIf2BsQosScArqDeliverInOrder OBJECT-TYPE
25             SYNTAX      TruthValue
26             MAX-ACCESS  read-create
27             STATUS      current
28             DESCRIPTION
29                 "Indicates whether or not data is to be delivered
30                 by the receiving MAC to its client application
31                 in the order in which data was handed off to the
32                 originating MAC."
33             REFERENCE
34                 "Subclause 11.13.18 in IEEE Std 802.16-2004"
35             DEFVAL    {0}
36             ::= { wmanIf2BsServiceClassEntry 16 }
37
38
39
40         wmanIf2BsQosScArqRxPurgeTimeout OBJECT-TYPE
41             SYNTAX      INTEGER (0 .. 65535)
42             UNITS        "10 us"
43             MAX-ACCESS  read-create
44             STATUS      current
45             DESCRIPTION
46                 "Indicates the time interval the ARQ window is advanced
47                 after a fragment is received. A value of 0 means
48                 Infinite."
49             REFERENCE
50                 "Subclause 11.13.18 in IEEE Std 802.16-2004"
51             DEFVAL    {0}
52             ::= { wmanIf2BsServiceClassEntry 17 }
53
54
55
56         wmanIf2BsQosScArqBlockSize OBJECT-TYPE
57             SYNTAX      INTEGER (1..2040)
58             UNITS        "byte"
59             MAX-ACCESS  read-create
60             STATUS      current
61             DESCRIPTION
62                 "The value of this parameter specifies the size of an
63                 ARQ block. This parameter shall be established by
64
65

```

```

1         negotiation during the connection creation dialog."
2     REFERENCE
3         "Subclause 11.13.18.8 in IEEE Std 802.16-2004"
4     ::= { wmanIf2BsServiceClassEntry 18 }
5
6
7     wmanIf2BsQoSsCMinRsvdTolerableRate OBJECT-TYPE
8         SYNTAX      Unsigned32
9         UNITS       "b/s"
10        MAX-ACCESS  read-create
11        STATUS      current
12        DESCRIPTION
13            "Minimum Tolerable Traffic Rate = R (bits/sec) with
14            time base T(sec) means the following. Let S denote
15            additional demand accumulated at the MAC SAP of the
16            transmitter during an arbitrary time interval of the
17            length T. Then the amount of data forwarded at the
18            receiver to CS (in bits) during this interval should
19            be not less than min {S, R * T}."
20        REFERENCE
21            "Subclause 11.13.9 in IEEE Std 802.16-2004"
22        ::= { wmanIf2BsServiceClassEntry 19 }
23
24
25     wmanIf2BsQoSReqTxPolicy OBJECT-TYPE
26         SYNTAX      BITS {noBroadcastBwReq(0),
27             reserved1(1),
28             noPiggybackReq(2),
29             noFragmentData(3),
30             noPHS(4),
31             noSduPacking(5),
32             noCrc(6),
33             reserved2(7)}
34         MAX-ACCESS  read-create
35         STATUS      current
36         DESCRIPTION
37             "The value of this parameter provides the capability to
38             specify certain attributes for the associated service
39             flow. An attribute is enabled by setting the
40             corresponding bit position to 1."
41         REFERENCE
42             "Subclause 11.13.12 in IEEE Std 802.16-2004"
43         ::= { wmanIf2BsServiceClassEntry 20 }
44
45
46     wmanIf2BsQoSServiceClassRowStatus OBJECT-TYPE
47         SYNTAX      RowStatus
48         MAX-ACCESS  read-create
49         STATUS      current
50         DESCRIPTION
51             "This object is used to create a new row or modify or
52             delete an existing row in this table.
53
54             If the implementator of this MIB has choosen not
55             to implement 'dynamic assignment' of profiles, this
56             object is not useful and should return noSuchName
57             upon SNMP request."
58         ::= { wmanIf2BsServiceClassEntry 21 }
59
60
61
62
63
64
65

```

```

1
2
3 wmanIf2BsClassifierRuleTable OBJECT-TYPE
4     SYNTAX      SEQUENCE OF WmanIf2BsClassifierRuleEntry
5     MAX-ACCESS  not-accessible
6     STATUS      current
7     DESCRIPTION
8         "This table contains packet classifier rules associated
9         with service flows."
10
11     REFERENCE
12         "Subclause 11.13.19.3.4 in IEEE Std 802.16-2004"
13     ::= { wmanIf2BsPacketCs 4 }
14
15 wmanIf2BsClassifierRuleEntry OBJECT-TYPE
16     SYNTAX      WmanIf2BsClassifierRuleEntry
17     MAX-ACCESS  not-accessible
18     STATUS      current
19     DESCRIPTION
20
21         "This table provides one row for each packet classifier
22         rule, and is indexed by ifIndex, wmanIf2BsSfId, and
23         wmanIf2BsClassifierRuleIndex. IfIndex is associated with
24         the BS sector. wmanIf2BsSfId identifies the service flow,
25         while wmanIf2BsClassifierRuleIndex identifies the packet
26         classifier rule."
27
28     INDEX { ifIndex, wmanIf2BsSfId, wmanIf2BsClassifierRuleIndex }
29     ::= { wmanIf2BsClassifierRuleTable 1 }
30
31
32 WmanIf2BsClassifierRuleEntry ::= SEQUENCE {
33     wmanIf2BsClassifierRuleIndex      Unsigned32,
34     wmanIf2BsClassifierRulePriority    INTEGER,
35     wmanIf2BsClassifierRuleIpTosLow   INTEGER,
36     wmanIf2BsClassifierRuleIpTosHigh  INTEGER,
37     wmanIf2BsClassifierRuleIpTosMask  INTEGER,
38     wmanIf2BsClassifierRuleIpProtocol Integer32,
39     wmanIf2BsClassifierRuleIpSourceAddr InetAddress,
40     wmanIf2BsClassifierRuleIpSourceMask InetAddress,
41     wmanIf2BsClassifierRuleIpDestAddr  InetAddress,
42     wmanIf2BsClassifierRuleIpDestMask  InetAddress,
43     wmanIf2BsClassifierRuleSourcePortStart Integer32,
44     wmanIf2BsClassifierRuleSourcePortEnd Integer32,
45     wmanIf2BsClassifierRuleDestPortStart Integer32,
46     wmanIf2BsClassifierRuleDestPortEnd Integer32,
47     wmanIf2BsClassifierRuleDestMacAddr  MacAddress,
48     wmanIf2BsClassifierRuleDestMacMask  MacAddress,
49     wmanIf2BsClassifierRuleSourceMacAddr MacAddress,
50     wmanIf2BsClassifierRuleSourceMacMask MacAddress,
51     wmanIf2BsClassifierRuleEnetProtocolType INTEGER,
52     wmanIf2BsClassifierRuleEnetProtocol Integer32,
53     wmanIf2BsClassifierRuleUserPriLow   Integer32,
54     wmanIf2BsClassifierRuleUserPriHigh  Integer32,
55     wmanIf2BsClassifierRuleVlanId       Integer32,
56     wmanIf2BsClassifierRuleState        INTEGER,
57     wmanIf2BsClassifierRulePhsSize       Integer32,
58     wmanIf2BsClassifierRulePhsMask      OCTET STRING,
59     wmanIf2BsClassifierRulePhsVerify    WmanIf2PhsRuleVerify,
60
61
62
63
64
65

```



```

1      wmanIf2BsClassifierRuleIpv6FlowLabel      WmanIf2Ipv6FlowLabel,
2      wmanIf2BsClassifierRuleBitMap            WmanIf2ClassifierBitMap,
3      wmanIf2BsClassifierRuleRowStatus         RowStatus}
4
5
6      wmanIf2BsClassifierRuleIndex OBJECT-TYPE
7          SYNTAX      Unsigned32 (1..4294967295)
8          MAX-ACCESS  not-accessible
9          STATUS      current
10         DESCRIPTION
11             "An index is assigned to a classifier in BS classifiers
12              table"
13         ::= { wmanIf2BsClassifierRuleEntry 1 }
14
15
16      wmanIf2BsClassifierRulePriority OBJECT-TYPE
17          SYNTAX      INTEGER (0..255)
18          MAX-ACCESS  read-create
19          STATUS      current
20          DESCRIPTION
21             "The value specifies the priority for the Classifier, which
22              is used for determining the order of the Classifier. A
23              higher value indicates higher priority. Classifiers may
24              have priorities in the range 0..255."
25          REFERENCE
26             "Subclause 11.13.19.3.4.1 in IEEE Std 802.16-2004"
27          DEFVAL      { 0 }
28          ::= { wmanIf2BsClassifierRuleEntry 2 }
29
30
31      wmanIf2BsClassifierRuleIpTosLow OBJECT-TYPE
32          SYNTAX      INTEGER (0..255)
33          MAX-ACCESS  read-create
34          STATUS      current
35          DESCRIPTION
36             "The low value of a range of TOS byte values. If the
37              referenced parameter is not present in a classifier, this
38              object reports the value of 0."
39          REFERENCE
40             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
41          ::= { wmanIf2BsClassifierRuleEntry 3 }
42
43
44      wmanIf2BsClassifierRuleIpTosHigh OBJECT-TYPE
45          SYNTAX      INTEGER (0..255)
46          MAX-ACCESS  read-create
47          STATUS      current
48          DESCRIPTION
49             "The 8-bit high value of a range of TOS byte values.
50              If the referenced parameter is not present in a classifier,
51              this object reports the value of 0."
52          REFERENCE
53             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
54          ::= { wmanIf2BsClassifierRuleEntry 4 }
55
56
57      wmanIf2BsClassifierRuleIpTosMask OBJECT-TYPE
58          SYNTAX      INTEGER (0..255)
59          MAX-ACCESS  read-create
60
61
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object specifies the matching parameter
4          for the IP type of service/DSCP [IETF RFC 2474] byte mask.
5          An IP packet with IP type of service (ToS) byte value
6          ip-tos matches this parameter if tos-low less than or
7          equal (ip-tos AND tos-mask) less than or equal tos-high."
8
9      REFERENCE
10         "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
11         ::= { wmanIf2BsClassifierRuleEntry 5 }
12
13
14 wmanIf2BsClassifierRuleIpProtocol OBJECT-TYPE
15     SYNTAX      Integer32 (0..255)
16     MAX-ACCESS  read-create
17     STATUS      current
18     DESCRIPTION
19         "This object indicates the value of the IP Protocol field
20         required for IP packets to match this rule. If the
21         referenced parameter is not present in a classifier, this
22         object reports the value of 0."
23
24     REFERENCE
25         "Subclause 11.13.19.3.4.3 in IEEE Std 802.16-2004"
26         ::= { wmanIf2BsClassifierRuleEntry 6 }
27
28
29
30 wmanIf2BsClassifierRuleIpSourceAddr OBJECT-TYPE
31     SYNTAX      InetAddress
32     MAX-ACCESS  read-create
33     STATUS      current
34     DESCRIPTION
35         "This object specifies the value of the IP Source Address
36         required for packets to match this rule. An IP packet
37         matches the rule when the packet ip source address bitwise
38         ANDed with the wmanIf2BsClassifierRuleIpSourceMask value
39         equals the wmanIf2BsClassifierRuleIpSourceAddr value.
40         If the referenced parameter is not present in a classifier,
41         this object reports the value of 0.0.0.0."
42
43     REFERENCE
44         "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
45         ::= { wmanIf2BsClassifierRuleEntry 7 }
46
47
48
49 wmanIf2BsClassifierRuleIpSourceMask OBJECT-TYPE
50     SYNTAX      InetAddress
51     MAX-ACCESS  read-create
52     STATUS      current
53     DESCRIPTION
54         "This object specifies which bits of a packet's IP Source
55         Address that are compared to match this rule. An IP packet
56         matches the rule when the packet source address bitwise
57         ANDed with the
58         wmanIf2BsClassifierRuleIpSourceMask value equals the
59         wmanIf2BsClassifierRuleIpSourceAddr value.
60         If the referenced parameter is not present in a classifier,
61         this object reports the value of 0.0.0.0."
62
63     REFERENCE
64
65

```

```

1      "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
2      ::= { wmanIf2BsClassifierRuleEntry 8 }
3
4
5  wmanIf2BsClassifierRuleIpDestAddr OBJECT-TYPE
6      SYNTAX      InetAddress
7      MAX-ACCESS  read-create
8      STATUS      current
9      DESCRIPTION
10         "This object specifies the value of the IP Destination
11         Address required for packets to match this rule. An IP
12         packet matches the rule when the packet IP destination
13         address bitwise ANDed with the
14         wmanIf2BsClassifierRuleIpDestMask value equals the
15         wmanIf2BsClassifierRuleIpDestAddr value.
16         If the referenced parameter is not present in a
17         classifier, this object reports the value of 0.0.0.0."
18     REFERENCE
19         "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
20     ::= { wmanIf2BsClassifierRuleEntry 9 }
21
22
23  wmanIf2BsClassifierRuleIpDestMask OBJECT-TYPE
24
25      SYNTAX      InetAddress
26      MAX-ACCESS  read-create
27      STATUS      current
28      DESCRIPTION
29         "This object specifies which bits of a packet's IP
30         Destination Address that are compared to match this rule.
31         An IP packet matches the rule when the packet destination
32         address bitwise ANDed with the
33         wmanIf2BsClassifierRuleIpDestMask value equals the
34         wmanIf2BsClassifierRuleIpDestAddr value.
35         If the referenced parameter is not present in a classifier
36         , this object reports the value of 0.0.0.0."
37     REFERENCE
38         "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
39     ::= { wmanIf2BsClassifierRuleEntry 10 }
40
41
42  wmanIf2BsClassifierRuleSourcePortStart OBJECT-TYPE
43
44      SYNTAX      Integer32 (0..65535)
45      MAX-ACCESS  read-create
46      STATUS      current
47      DESCRIPTION
48         "This object specifies the low end inclusive range of
49         TCP/UDP source port numbers to which a packet is compared.
50         This object is irrelevant for non-TCP/UDP IP packets.
51         If the referenced parameter is not present in a
52         classifier, this object reports the value of 0."
53     REFERENCE
54         "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
55     ::= { wmanIf2BsClassifierRuleEntry 11 }
56
57
58  wmanIf2BsClassifierRuleSourcePortEnd OBJECT-TYPE
59
60      SYNTAX      Integer32 (0..65535)
61      MAX-ACCESS  read-create
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object specifies the high end inclusive range of
4          TCP/UDP source port numbers to which a packet is compared.
5          This object is irrelevant for non-TCP/UDP IP packets.
6          If the referenced parameter is not present in a classifier,
7          this object reports the value of 65535."
8
9      REFERENCE
10         "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
11         ::= { wmanIf2BsClassifierRuleEntry 12 }
12
13
14 wmanIf2BsClassifierRuleDestPortStart OBJECT-TYPE
15     SYNTAX      Integer32 (0..65535)
16     MAX-ACCESS  read-create
17     STATUS      current
18     DESCRIPTION
19         "This object specifies the low end inclusive range of
20         TCP/UDP destination port numbers to which a packet is
21         compared. If the referenced parameter is not present
22         in a classifier, this object reports the value of 0."
23
24     REFERENCE
25         "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
26         ::= { wmanIf2BsClassifierRuleEntry 13 }
27
28
29
30 wmanIf2BsClassifierRuleDestPortEnd OBJECT-TYPE
31     SYNTAX      Integer32 (0..65535)
32     MAX-ACCESS  read-create
33     STATUS      current
34     DESCRIPTION
35         "This object specifies the high end inclusive range of
36         TCP/UDP destination port numbers to which a packet is
37         compared. If the referenced parameter is not present
38         in a classifier, this object reports the value of
39         65535."
40
41     REFERENCE
42         "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
43         ::= { wmanIf2BsClassifierRuleEntry 14 }
44
45
46
47 wmanIf2BsClassifierRuleDestMacAddr OBJECT-TYPE
48     SYNTAX      MacAddress
49     MAX-ACCESS  read-create
50     STATUS      current
51     DESCRIPTION
52         "An Ethernet packet matches an entry when its destination
53         MAC address bitwise ANDed with
54         wmanIf2BsClassifierRuleDestMacMask equals the value of
55         wmanIf2BsClassifierRuleDestMacAddr. If the referenced
56         parameter is not present in a classifier, this object
57         reports the value of '000000000000'H."
58
59     REFERENCE
60         "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
61         ::= { wmanIf2BsClassifierRuleEntry 15 }
62
63
64
65 wmanIf2BsClassifierRuleDestMacMask OBJECT-TYPE

```

```

1      SYNTAX      MacAddress
2      MAX-ACCESS  read-create
3      STATUS      current
4      DESCRIPTION
5          "An Ethernet packet matches an entry when its destination
6          MAC address bitwise ANDed with
7          wmanIf2BsClassifierRuleDestMacMask equals the value of
8          wmanIf2BsClassifierRuleDestMacAddr. If the referenced
9          parameter is not present in a classifier, this object
10         reports the value of '000000000000'H."
11
12     REFERENCE
13         "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
14     ::= { wmanIf2BsClassifierRuleEntry 16 }
15
16
17
18 wmanIf2BsClassifierRuleSourceMacAddr OBJECT-TYPE
19     SYNTAX      MacAddress
20     MAX-ACCESS  read-create
21     STATUS      current
22     DESCRIPTION
23         "An Ethernet packet matches this entry when its source
24         MAC address bitwise ANDed with
25         wmanIf2BsClassifierRuleSourceMacMask equals the value
26         of wmanIf2BsClassifierRuleSourceMacAddr. If the
27         referenced parameter is not present in a classifier,
28         this object reports the value of '000000000000'H."
29
30     REFERENCE
31         "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
32     ::= { wmanIf2BsClassifierRuleEntry 17 }
33
34
35
36 wmanIf2BsClassifierRuleSourceMacMask OBJECT-TYPE
37     SYNTAX      MacAddress
38     MAX-ACCESS  read-create
39     STATUS      current
40     DESCRIPTION
41         "An Ethernet packet matches an entry when its source
42         MAC address bitwise ANDed with
43         wmanIf2BsClassifierRuleSourceMacMask equals the value of
44         wmanIf2BsClassifierRuleSourceMacAddr. If the referenced
45         parameter is not present in a classifier, this object
46         reports the value of '000000000000'H."
47
48     REFERENCE
49         "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
50     ::= { wmanIf2BsClassifierRuleEntry 18 }
51
52
53
54 wmanIf2BsClassifierRuleEnetProtocolType OBJECT-TYPE
55     SYNTAX      INTEGER { none(0),
56                  ethertype(1),
57                  dsap(2) }
58     MAX-ACCESS  read-create
59     STATUS      current
60     DESCRIPTION
61         "This object indicates the format of the layer 3 protocol
62         id in the Ethernet packet. A value of none(0) means that
63         the rule does not use the layer 3 protocol type as a
64
65

```

1 matching criteria. A value of etherType(1) means that the  
 2 rule applies only to frames which contains an EtherType  
 3 value. EtherType values are contained in packets using  
 4 the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042  
 5 Sub-Network Access Protocol (SNAP) encapsulation formats.  
 6 A value of dsap(2) means that the rule applies only to  
 7 frames using the IEEE802.3 encapsulation format with a  
 8 Destination Service Access Point (DSAP) other than 0xAA  
 9 (which is reserved for SNAP). If the Ethernet frame  
 10 contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),  
 11 this object applies to the embedded EtherType field within  
 12 the 802.1P/Q header. If the referenced parameter is not  
 13 present in a classifier, this object reports the value of  
 14 0."  
 15  
 16  
 17

18 REFERENCE

19 "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"  
 20 ::= { wmanIf2BsClassifierRuleEntry 19 }  
 21

22 wmanIf2BsClassifierRuleEnetProtocol OBJECT-TYPE

23 SYNTAX Integer32 (0..65535)

24 MAX-ACCESS read-create

25 STATUS current

26 DESCRIPTION

27 "If wmanIf2BsClassifierRuleEnetProtocolType is none(0),  
 28 this object is ignored when considering whether a packet  
 29 matches the current rule.  
 30 If wmanIf2BsClassifierRuleEnetProtocolType is etherType(1),  
 31 this object gives the 16-bit value of the EtherType that  
 32 the packet must match in order to match the rule.  
 33 If wmanIf2BsClassifierRuleEnetProtocolType is dsap(2), the  
 34 lower 8 bits of this object's value must match the DSAP  
 35 byte of the packet in order to match the rule.  
 36 If the Ethernet frame contains an 802.1P/Q Tag header  
 37 (i.e. EtherType 0x8100), this object applies to the  
 38 embedded EtherType field within the 802.1P/Q header.  
 39 If the referenced parameter is not present in the  
 40 classifier, the value of this object is reported as 0."  
 41  
 42  
 43  
 44

45 REFERENCE

46 "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"  
 47 ::= { wmanIf2BsClassifierRuleEntry 20 }  
 48  
 49

50 wmanIf2BsClassifierRuleUserPriLow OBJECT-TYPE

51 SYNTAX Integer32 (0..7)

52 MAX-ACCESS read-create

53 STATUS current

54 DESCRIPTION

55 "This object applies only to Ethernet frames using the  
 56 802.1P/Q tag header (indicated with EtherType 0x8100).  
 57 Such frames include a 16-bit Tag that contains a 3 bit  
 58 Priority field and a 12 bit VLAN number.  
 59 Tagged Ethernet packets must have a 3-bit Priority field  
 60 within the range of wmanIf2BsClassifierRuleUserPriLow and  
 61 wmanIf2BsClassifierRuleUserPriHigh in order to match this  
 62 rule.  
 63  
 64  
 65

```

1         If the referenced parameter is not present in the
2         classifier, the value of this object is reported as 0."
3     REFERENCE
4         "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
5     ::= { wmanIf2BsClassifierRuleEntry 21 }
6
7
8     wmanIf2BsClassifierRuleUserPriHigh OBJECT-TYPE
9     SYNTAX      Integer32 (0..7)
10    MAX-ACCESS  read-create
11    STATUS      current
12    DESCRIPTION
13        "This object applies only to Ethernet frames using the
14        802.1P/Q tag header (indicated with EtherType 0x8100).
15        Such frames include a 16-bit Tag that contains a 3 bit
16        Priority field and a 12 bit VLAN number.
17        Tagged Ethernet packets must have a 3-bit Priority
18        field within the range of wmanIf2BsClassifierRuleUserPriLow
19        and wmanIf2BsClassifierRuleUserPriHigh in order to match
20        this rule.
21        If the referenced parameter is not present in the
22        classifier, the value of this object is reported as 7."
23    REFERENCE
24        "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
25    ::= { wmanIf2BsClassifierRuleEntry 22 }
26
27
28    wmanIf2BsClassifierRuleVlanId OBJECT-TYPE
29    SYNTAX      Integer32 (0..4095)
30    MAX-ACCESS  read-create
31    STATUS      current
32    DESCRIPTION
33        "This object applies only to Ethernet frames using the
34        802.1P/Q tag header.
35        If this object's value is nonzero, tagged packets must
36        have a VLAN Identifier that matches the value in order
37        to match the rule.
38        Only the least significant 12 bits of this object's
39        value are valid.
40        If the referenced parameter is not present in the
41        classifier, the value of this object is reported as 0."
42    REFERENCE
43        "Subclause 11.13.19.3.4.12 in IEEE Std 802.16-2004"
44    ::= { wmanIf2BsClassifierRuleEntry 23 }
45
46
47    wmanIf2BsClassifierRuleState OBJECT-TYPE
48    SYNTAX      INTEGER {active(1),
49                  inactive(2)}
50    MAX-ACCESS  read-create
51    STATUS      deprecated
52    DESCRIPTION
53        "This object indicates whether or not the classifier is
54        enabled to classify packets to a Service Flow.
55        If the referenced parameter is not present in the
56        classifier, the value of this object is reported
57        as active(1)."
```

```

1      ::= { wmanIf2BsClassifierRuleEntry 24 }
2
3
4  wmanIf2BsClassifierRulePhsSize OBJECT-TYPE
5      SYNTAX      Integer32 (0..255)
6      UNITS       "byte"
7      MAX-ACCESS  read-create
8      STATUS      current
9      DESCRIPTION
10         "This object is used to configure the PHS rule for this
11         classifier. The value of this field - PHSS is the total
12         number of bytes in the header to be suppressed and then
13         restored in a service flow that uses PHS. If the value of
14         this field is 0 bytes then PHS is disabled for this
15         classifier. If flag phsMask in wmanIf2BsClassifierRuleBitMap
16         is set to 0 and flag phsSize in
17         wmanIf2BsClassifierRuleBitMap is set to 0, then BS can still
18         create PHS rules using its own custom mask (i.e. the rule
19         is not configured by NMS)."
```

REFERENCE

```

24         "Subclause 11.13.19.3.7.4 in IEEE Std 802.16-2004"
25     DEFVAL      {0}
26     ::= { wmanIf2BsClassifierRuleEntry 25 }
27
28
29  wmanIf2BsClassifierRulePhsMask OBJECT-TYPE
30      SYNTAX      OCTET STRING (SIZE(0..65535))
31      MAX-ACCESS  read-create
32      STATUS      current
33      DESCRIPTION
34         "This object is used to configure the PHS rule for this
35         classifier. It is encoded as follows:
36         bit 0:
37             0 = don't suppress the 1st byte of the suppression field
38             1 = suppress first byte of the suppression field
39         bit 1:
40             0 = don't suppress the 2nd byte of the suppression field
41             1 = suppress second byte of the suppression field
42         bit x:
43             0 = don't suppress the (x+1) byte of the suppression
44             field
45             1 = suppress (x+1) byte of the suppression field
46         where the length of the octet string is ceiling
47         (wmanIf2BsClassifierRulePhsSize/8). BS should use this value
48         to create a new PHS rule index (PHSI) and field (PHSF) as
49         defined in the standard. If flag phsMask in
50         wmanIf2BsClassifierRuleBitMap is set to 0 and flag phsSize
51         in wmanIf2BsClassifierRuleBitMap is set to 0, then BS can
52         still create PHS rules using its own custom mask (i.e. the
53         rule is not configured by NMS)."
```

REFERENCE

```

59         "Subclause 11.13.19.3.7.3 in IEEE Std 802.16-2004"
60     ::= { wmanIf2BsClassifierRuleEntry 26 }
61
62
63  wmanIf2BsClassifierRulePhsVerify OBJECT-TYPE
64      SYNTAX      WmanIf2PhsRuleVerify
65
```



```

1      MAX-ACCESS read-create
2      STATUS current
3      DESCRIPTION
4          "The value of this field indicates to the sending entity
5          whether or not the packet header contents are to be
6          verified prior to performing suppression."
7      DEFVAL { phsVerifyEnable }
8      ::= { wmanIf2BsClassifierRuleEntry 27 }
9
10
11
12 wmanIf2BsClassifierRuleIpv6FlowLabel OBJECT-TYPE
13     SYNTAX WmanIf2Ipv6FlowLabel
14     MAX-ACCESS read-create
15     STATUS current
16     DESCRIPTION
17         "The value of this field specifies the matching values for
18         the IPv6 Flow label field."
19     ::= { wmanIf2BsClassifierRuleEntry 28 }
20
21
22
23 wmanIf2BsClassifierRuleBitMap OBJECT-TYPE
24     SYNTAX WmanIf2ClassifierBitMap
25     MAX-ACCESS read-create
26     STATUS current
27     DESCRIPTION
28         "This object indicates which parameter encodings were
29         actually present in the entry. A bit set to '1' indicates
30         the corresponding classifier encoding is present, and '0'
31         means otherwise"
32     ::= { wmanIf2BsClassifierRuleEntry 29 }
33
34
35
36 wmanIf2BsClassifierRuleRowStatus OBJECT-TYPE
37     SYNTAX RowStatus
38     MAX-ACCESS read-create
39     STATUS current
40     DESCRIPTION
41         "This object is used to create a new row or modify or
42         delete an existing row in this table.
43
44         If the implementator of this MIB has choosen not
45         to implement 'dynamic assignment' of profiles, this
46         object is not useful and should return noSuchName
47         upon SNMP request."
48     ::= { wmanIf2BsClassifierRuleEntry 30 }
49
50
51
52
53 wmanIf2BsSsPacketCounterTable OBJECT-TYPE
54     SYNTAX SEQUENCE OF WmanIf2BsSsPacketCounterEntry
55     MAX-ACCESS not-accessible
56     STATUS current
57     DESCRIPTION
58         "This table contains counters to keep track of the number
59         of packets and octets that have been received or
60         transmitted on the per service flow basis."
61     ::= { wmanIf2BsPacketCs 5 }
62
63
64
65 wmanIf2BsSsPacketCounterEntry OBJECT-TYPE

```

```

1      SYNTAX      WmanIf2BsSsPacketCounterEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table provides one row for each service flow, and
6          is indexed by ifIndex, wmanIf2CmnCpsSfMacAddress, and
7          wmanIf2CmnCpsSfId."
8      INDEX { ifIndex, wmanIf2CmnCpsSfMacAddress,
9              wmanIf2CmnCpsSfId }
10     ::= { wmanIf2BsSsPacketCounterTable 1 }
11
12
13
14     WmanIf2BsSsPacketCounterEntry ::= SEQUENCE {
15         wmanIf2BsSsMacSduCount          Counter64,
16         wmanIf2BsSsOctetCount          Counter64,
17         wmanIf2BsSsResetCounter        INTEGER,
18         wmanIf2BsSsResetCounterTime    TimeStamp}
19
20
21     wmanIf2BsSsMacSduCount OBJECT-TYPE
22         SYNTAX      Counter64
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "This object counts the number of MAC SDUs that have
27             been transmitted or received."
28         ::= { wmanIf2BsSsPacketCounterEntry 1 }
29
30
31
32     wmanIf2BsSsOctetCount OBJECT-TYPE
33         SYNTAX      Counter64
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "This object counts the number of octets of MAC SDUs
38             that have been transmitted or received."
39         ::= { wmanIf2BsSsPacketCounterEntry 2 }
40
41
42
43     wmanIf2BsSsResetCounter OBJECT-TYPE
44         SYNTAX      INTEGER {null(0),
45                             resetCounter(1)}
46         MAX-ACCESS  read-write
47         STATUS      current
48         DESCRIPTION
49             "When this attribute is SET to resetCounter(1), the
50             corresponding entry of packet counters will be reset.
51             A GET operation performed on this object will always
52             return null(0). The counter is normally reset after
53             the packet count information is retrieved. "
54         ::= { wmanIf2BsSsPacketCounterEntry 3 }
55
56
57
58     wmanIf2BsSsResetCounterTime OBJECT-TYPE
59         SYNTAX      TimeStamp
60         MAX-ACCESS  read-only
61         STATUS      current
62         DESCRIPTION
63             "Indicates the date and time when the counter is
64
65

```

```

1         reset."
2         ::= { wmanIf2BsSsPacketCounterEntry 4 }
3
4
5         --
6         -- wmanIf2BsCps contain the Base Station Common Part Sublayer objects
7         --
8         wmanIf2BsCps OBJECT IDENTIFIER ::= { wmanIf2BsObjects 2 }
9
10
11 wmanIf2BsRegisteredSsTable OBJECT-TYPE
12     SYNTAX      SEQUENCE OF WmanIf2BsRegisteredSsEntry
13     MAX-ACCESS  not-accessible
14     STATUS      current
15     DESCRIPTION
16         "This table contains the basic capability information
17         of SSs that have been negotiated and agreed between
18         BS and SS via REG-REQ and REG-RSP messages. An entry
19         in this table indicates the SS has entered and registered
20         into the BS."
21     REFERENCE
22         "Subclause 6.3.2.3.7 in IEEE Std 802.16-2004"
23     ::= { wmanIf2BsCps 1 }
24
25
26
27 wmanIf2BsRegisteredSsEntry OBJECT-TYPE
28     SYNTAX      WmanIf2BsRegisteredSsEntry
29     MAX-ACCESS  not-accessible
30     STATUS      current
31     DESCRIPTION
32         "This table provides one row for each SS that has been
33         registered in the BS, and is indexed by
34         wmanIf2BsSsMacAddress. The primary index is the ifIndex
35         with an ifType of ieee80216WMAN, indicating the BS sector
36         with which the SS is associated. wmanIf2BsSsMacAddress
37         identifies the SS being registered."
38     INDEX { ifIndex, wmanIf2BsSsMacAddress }
39     ::= { wmanIf2BsRegisteredSsTable 1 }
40
41
42
43
44 WmanIf2BsRegisteredSsEntry ::= SEQUENCE {
45     wmanIf2BsSsMacAddress      MacAddress,
46     wmanIf2BsSsBasicCid       WmanIf2CidType,
47     wmanIf2BsSsPrimaryCid     WmanIf2CidType,
48     wmanIf2BsSsSecondaryCid   WmanIf2CidType,
49     wmanIf2BsSsManagementSupport  INTEGER,
50     wmanIf2BsSsIpManagementMode  INTEGER,
51     wmanIf2BsSs2ndMgmtArqEnable  TruthValue,
52     wmanIf2BsSs2ndMgmtArqWindowSize  INTEGER,
53     wmanIf2BsSs2ndMgmtArqDnLinkTxDelay  INTEGER,
54     wmanIf2BsSs2ndMgmtArqUpLinkTxDelay  INTEGER,
55     wmanIf2BsSs2ndMgmtArqDnLinkRxDelay  INTEGER,
56     wmanIf2BsSs2ndMgmtArqUpLinkRxDelay  INTEGER,
57     wmanIf2BsSs2ndMgmtArqBlockLifetime  INTEGER,
58     wmanIf2BsSs2ndMgmtArqSyncLossTimeout  INTEGER,
59     wmanIf2BsSs2ndMgmtArqDeliverInOrder  TruthValue,
60     wmanIf2BsSs2ndMgmtArqRxPurgeTimeout  INTEGER,
61     wmanIf2BsSs2ndMgmtArqBlockSize     INTEGER,
62
63
64
65

```

```

1      wmanIf2BsSsVendorIdEncoding          OCTET STRING,
2      wmanIf2BsSsAasBroadcastPermission    INTEGER,
3      wmanIf2BsSsMaxTxPowerBpsk           WmanIf2MaxTxPowerType,
4      wmanIf2BsSsMaxTxPowerQpsk          WmanIf2MaxTxPowerType,
5      wmanIf2BsSsMaxTxPower16Qam         WmanIf2MaxTxPowerType,
6      wmanIf2BsSsMaxTxPower64Qam        WmanIf2MaxTxPowerType,
7      wmanIf2BsSsMacVersion              WmanIf2MacVersion}
8
9
10     wmanIf2BsSsMacAddress OBJECT-TYPE
11     SYNTAX      MacAddress
12     MAX-ACCESS  not-accessible
13     STATUS      current
14     DESCRIPTION
15     "The MAC address of SS is received from the RNG-REQ
16     message. When SS registers, this MAC address is entered
17     into the table, and used as the identifier to the SS."
18     REFERENCE
19     "Subclause 6.3.2.3.5 in IEEE Std 802.16-2004"
20     ::= { wmanIf2BsRegisteredSsEntry 1 }
21
22
23     wmanIf2BsSsBasicCid OBJECT-TYPE
24     SYNTAX      WmanIf2CidType
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28     "The value of this object indicates the SS's basic CID
29     that was sent in the RNG-RSP message."
30     REFERENCE
31     "Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
32     ::= { wmanIf2BsRegisteredSsEntry 2 }
33
34
35     wmanIf2BsSsPrimaryCid OBJECT-TYPE
36     SYNTAX      WmanIf2CidType
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40     "The value of this object indicates the primary CID of the
41     SS received from the RNG-RSP message."
42     REFERENCE
43     "Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
44     ::= { wmanIf2BsRegisteredSsEntry 3 }
45
46
47     wmanIf2BsSsSecondaryCid OBJECT-TYPE
48     SYNTAX      WmanIf2CidType
49     MAX-ACCESS  read-only
50     STATUS      current
51     DESCRIPTION
52     "The value of this object indicates the secondary
53     management CID present in the REG-RSP message. The value
54     should be null if the 2nd management connection is not
55     available."
56     REFERENCE
57     "Subclause 6.4.2.3.8 in IEEE Std 802.16-2004"
58     ::= { wmanIf2BsRegisteredSsEntry 4 }
59
60
61
62
63
64
65

```

```

1
2
3 wmanIf2BsSsManagementSupport OBJECT-TYPE
4     SYNTAX      INTEGER {unmanagedSs(0),
5                   managedSs(1)}
6     MAX-ACCESS  read-only
7     STATUS      current
8     DESCRIPTION
9         "This object indicates whether or not the SS is managed."
10
11    REFERENCE
12        "Subclause 11.7.2 in IEEE Std 802.16-2004"
13    ::= { wmanIf2BsRegisteredSsEntry 5 }
14
15 wmanIf2BsSsIpManagementMode OBJECT-TYPE
16     SYNTAX      INTEGER {unmanaged(0),
17                   ipManaged(1)}
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21         "The IP management mode parameter dictates whether
22         the provider intends to manage the SS on an ongoing
23         basis via IP-based mechanisms."
24     REFERENCE
25         "Subclause 11.7.3 in IEEE Std 802.16-2004"
26     ::= { wmanIf2BsRegisteredSsEntry 6 }
27
28
29
30
31 wmanIf2BsSs2ndMgmtArqEnable OBJECT-TYPE
32     SYNTAX      TruthValue
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36         "True(1) ARQ enabling is requested for the 2nd
37         management channel."
38     REFERENCE
39         "Subclause 11.13.18.1 in IEEE Std 802.16-2004"
40     ::= { wmanIf2BsRegisteredSsEntry 7 }
41
42
43
44 wmanIf2BsSs2ndMgmtArqWindowSize OBJECT-TYPE
45     SYNTAX      INTEGER (1 .. 1024)
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "Indicates the maximum number of unacknowledged
50         fragments at any time for 2nd management connection."
51     REFERENCE
52         "Subclause 11.13.18.2 in IEEE Std 802.16-2004"
53     ::= { wmanIf2BsRegisteredSsEntry 8 }
54
55
56
57 wmanIf2BsSs2ndMgmtArqDnLinkTxDelay OBJECT-TYPE
58     SYNTAX      INTEGER (0 .. 65535)
59     UNITS       "us"
60     MAX-ACCESS  read-only
61     STATUS      current
62     DESCRIPTION
63         "The object defines the ARQ transmitter delay for
64
65

```

```

1         downlink transmission."
2     REFERENCE
3         "Subclause 11.13.18.3 in IEEE Std 802.16-2004"
4     ::= { wmanIf2BsRegisteredSsEntry 9 }
5
6
7     wmanIf2BsSs2ndMgmtArqUpLinkTxDelay OBJECT-TYPE
8         SYNTAX      INTEGER (0 .. 65535)
9         UNITS       "us"
10        MAX-ACCESS  read-only
11        STATUS      current
12        DESCRIPTION
13            "The object defines the ARQ transmitter delay for
14            uplink transmission."
15        REFERENCE
16            "Subclause 11.13.18.3 in IEEE Std 802.16-2004"
17        ::= { wmanIf2BsRegisteredSsEntry 10 }
18
19
20
21    wmanIf2BsSs2ndMgmtArqDnLinkRxDelay OBJECT-TYPE
22        SYNTAX      INTEGER (0 .. 65535)
23        UNITS       "us"
24        MAX-ACCESS  read-only
25        STATUS      current
26        DESCRIPTION
27            "The object defines the ARQ receiver delay for
28            downlink transmission."
29        REFERENCE
30            "Subclause 11.13.18.3 in IEEE Std 802.16-2004"
31        ::= { wmanIf2BsRegisteredSsEntry 11 }
32
33
34
35
36    wmanIf2BsSs2ndMgmtArqUpLinkRxDelay OBJECT-TYPE
37        SYNTAX      INTEGER (0 .. 65535)
38        UNITS       "us"
39        MAX-ACCESS  read-only
40        STATUS      current
41        DESCRIPTION
42            "The object defines the ARQ receiver delay for
43            uplink transmission."
44        REFERENCE
45            "Subclause 11.13.18.3 in IEEE Std 802.16-2004"
46        ::= { wmanIf2BsRegisteredSsEntry 12 }
47
48
49
50    wmanIf2BsSs2ndMgmtArqBlockLifetime OBJECT-TYPE
51        SYNTAX      INTEGER (0 .. 65535)
52        UNITS       "10 us"
53        MAX-ACCESS  read-only
54        STATUS      current
55        DESCRIPTION
56            "The maximum time interval an ARQ fragment will be
57            managed by the transmitter ARQ machine, once
58            initial transmission of the fragment has occurred.
59            If transmission or retransmission of the fragment
60            is not acknowledged by the receiver before the
61            time limit is reached, the fragment is discarded.
62            A value of 0 means Infinite."
63
64
65

```

```

1      REFERENCE
2          "Subclause 11.13.18.4 in IEEE Std 802.16-2004"
3      DEFVAL    {0}
4      ::= { wmanIf2BsRegisteredSsEntry 13 }
5
6
7      wmanIf2BsSs2ndMgmtArqSyncLossTimeout OBJECT-TYPE
8          SYNTAX      INTEGER (0 .. 65535)
9          UNITS        "10 us"
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "The maximum interval before declaring a loss
14             of synchronization of the sender and receiver
15             state machines. A value of 0 means Infinite."
16         REFERENCE
17             "Subclause 11.13.18.5 in IEEE Std 802.16-2004"
18         DEFVAL    {0}
19         ::= { wmanIf2BsRegisteredSsEntry 14 }
20
21
22
23
24         wmanIf2BsSs2ndMgmtArqDeliverInOrder OBJECT-TYPE
25             SYNTAX      TruthValue
26             MAX-ACCESS  read-only
27             STATUS      current
28             DESCRIPTION
29                 "Indicates whether or not data is to be delivered
30                 by the receiving MAC to its client application
31                 in the order in which data was handed off to the
32                 originating MAC."
33             REFERENCE
34                 "Subclause 11.13.18.6 in IEEE Std 802.16-2004"
35             DEFVAL    {0}
36             ::= { wmanIf2BsRegisteredSsEntry 15 }
37
38
39
40         wmanIf2BsSs2ndMgmtArqRxPurgeTimeout OBJECT-TYPE
41             SYNTAX      INTEGER (0 .. 65535)
42             UNITS        "10 us"
43             MAX-ACCESS  read-only
44             STATUS      current
45             DESCRIPTION
46                 "Indicates the time interval the ARQ window is advanced
47                 after a fragment is received. A value of 0 means Infinite."
48             REFERENCE
49                 "Subclause 11.13.18.7 in IEEE Std 802.16-2004"
50             DEFVAL    {0}
51             ::= { wmanIf2BsRegisteredSsEntry 16 }
52
53
54
55         wmanIf2BsSs2ndMgmtArqBlockSize OBJECT-TYPE
56             SYNTAX      INTEGER (1 .. 2040)
57             MAX-ACCESS  read-only
58             STATUS      current
59             DESCRIPTION
60                 "This parameter specifies the size of a ARQ block. This
61                 parameter shall be established by negotiation during the
62                 connection setup. The requester includes its desired
63                 setting in the REQ message. The receiver of the REQ
64
65

```





```

1      "The maximum available power for QPSK. The maximum power
2      parameters are reported in dBm and quantized in 0.5 dBm
3      steps ranging from -64 dBm (encoded 0x00) to 63.5 dBm
4      (encoded 0xFF). Values outside this range shall be assigned
5      to closest extreme. This parameter is only applicable to
6      systems supporting the SCa, OFDM or OFDMA PHY."
7
8      REFERENCE
9      "Subclause 11.8.3.2 in IEEE Std 802.16-2004"
10     ::= { wmanIf2BsRegisteredSsEntry 21 }
11
12
13     wmanIf2BsSsMaxTxPower16Qam OBJECT-TYPE
14         SYNTAX      WmanIf2MaxTxPowerType
15         MAX-ACCESS  read-only
16         STATUS      current
17         DESCRIPTION
18             "The maximum available power for 16-QAM constellations.
19             The maximum power parameters are reported in dBm and
20             quantized in 0.5 dBm steps ranging from -64 dBm (encoded
21             0x00) to 63.5 dBm (encoded 0xFF). Values outside this
22             range shall be assigned the closest extreme. This parameter
23             is only applicable to systems supporting the SCa, OFDM or
24             OFDMA PHY."
25
26         REFERENCE
27             "Subclause 11.8.3.2 in IEEE Std 802.16-2004"
28         ::= { wmanIf2BsRegisteredSsEntry 22 }
29
30
31
32     wmanIf2BsSsMaxTxPower64Qam OBJECT-TYPE
33         SYNTAX      WmanIf2MaxTxPowerType
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "The maximum available power for 64-QAM constellations.
38             The maximum power parameters are reported in dBm and
39             quantized in 0.5 dBm steps ranging from -64 dBm (encoded
40             0x00) to 63.5 dBm (encoded 0xFF). Values outside this
41             range shall be assigned the closest extreme. Ss that do
42             not support QAM64 shall report the value of 0x00. This
43             parameter is only applicable to systems supporting the SCa,
44             OFDM or OFDMA PHY."
45
46         REFERENCE
47             "Subclause 11.8.3.2 in IEEE Std 802.16-2004"
48         ::= { wmanIf2BsRegisteredSsEntry 23 }
49
50
51
52     wmanIf2BsSsMacVersion OBJECT-TYPE
53         SYNTAX      WmanIf2MacVersion
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "This parameter specifies the version of 802.16 to which the
58             message originator conforms."
59
60         REFERENCE
61             "Subclause 11.1.3 in IEEE Std 802.16-2004"
62         ::= { wmanIf2BsRegisteredSsEntry 24 }
63
64
65

```

```

1  --
2  -- wmanIf2BsConfigurationTable contains global parameters common in BS
3  --
4
5  wmanIf2BsConfigurationTable OBJECT-TYPE
6      SYNTAX      SEQUENCE OF WmanIf2BsConfigurationEntry
7      MAX-ACCESS  not-accessible
8      STATUS      current
9      DESCRIPTION
10         "This table provides one row for each BS sector that
11         contains the BS system parameters as defined in Subclause
12         10.1 of [3]. The objects in this table define the default
13         behaviour of the BS for 2nd Management connection
14         scheduling and SFID allocation as well as configuration
15         parameters of the CPS scheduler and AAS system."
16
17     REFERENCE
18         "Subclause 10.1 in IEEE Std 802.16-2004"
19     ::= { wmanIf2BsCps 2 }
20
21
22
23  wmanIf2BsConfigurationEntry OBJECT-TYPE
24      SYNTAX      WmanIf2BsConfigurationEntry
25      MAX-ACCESS  not-accessible
26      STATUS      current
27      DESCRIPTION
28         "This table is indexed by ifIndex with an ifType of
29         ieee80216WMAN."
30     INDEX { ifIndex }
31     ::= { wmanIf2BsConfigurationTable 1 }
32
33
34
35  WmanIf2BsConfigurationEntry ::= SEQUENCE {
36      wmanIf2BsDcdInterval          INTEGER,
37      wmanIf2BsUcdInterval          INTEGER,
38      wmanIf2BsUcdTransition        INTEGER,
39      wmanIf2BsDcdTransition        INTEGER,
40      wmanIf2BsInitialRangingInterval  INTEGER,
41      wmanIf2BsSsULMapProcTime      Unsigned32,
42      wmanIf2BsSsRangRespProcTime    Unsigned32,
43      wmanIf2BsT5Timeout            INTEGER,
44      wmanIf2BsT9Timeout            INTEGER,
45      wmanIf2BsT13Timeout           INTEGER,
46      wmanIf2BsT15Timeout           INTEGER,
47      wmanIf2BsT17Timeout           INTEGER,
48      wmanIf2BsT27IdleTimer         Unsigned32,
49      wmanIf2BsT27ActiveTimer       Unsigned32,
50      wmanIf2Bs2ndMgmtDlQoSProfileIndex  INTEGER,
51      wmanIf2Bs2ndMgmtUlQoSProfileIndex  INTEGER,
52      wmanIf2BsAutoSfidEnabled       INTEGER,
53      wmanIf2BsAutoSfidRangeMin      Unsigned32,
54      wmanIf2BsAutoSfidRangeMax      Unsigned32,
55      wmanIf2BsAasChanFbckReqFreq    INTEGER,
56      wmanIf2BsAasBeamSelectFreq     INTEGER,
57      wmanIf2BsAasChanFbckReqResolution  INTEGER,
58      wmanIf2BsAasBeamReqResolution  INTEGER,
59      wmanIf2BsAasNumOptDiversityZones  INTEGER,
60      wmanIf2BsResetSector           INTEGER}
61
62
63
64
65

```

```

1
2
3 wmanIf2BsDcdInterval OBJECT-TYPE
4     SYNTAX      INTEGER (0..10000)
5     UNITS       "milliseconds"
6     MAX-ACCESS  read-write
7     STATUS      current
8     DESCRIPTION
9         "Time between transmission of DCD messages in ms."
10    ::= { wmanIf2BsConfigurationEntry 1 }
11
12
13 wmanIf2BsUcdInterval OBJECT-TYPE
14     SYNTAX      INTEGER (0..10000)
15     UNITS       "milliseconds"
16     MAX-ACCESS  read-write
17     STATUS      current
18     DESCRIPTION
19         "Time between transmission of UCD messages in ms."
20    ::= { wmanIf2BsConfigurationEntry 2 }
21
22
23
24 wmanIf2BsUcdTransition OBJECT-TYPE
25     SYNTAX      INTEGER (2..65535)
26     UNITS       "Number of MAC Frames"
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "The time the BS shall wait after transmitting a UCD message
31         with an incremented Configuration Change Count before
32         issuing a UL-MAP message referring to
33         Uplink_Burst_Profiles defined in that UCD message."
34    ::= { wmanIf2BsConfigurationEntry 3 }
35
36
37
38 wmanIf2BsDcdTransition OBJECT-TYPE
39     SYNTAX      INTEGER (2..65535)
40     UNITS       "Number of MAC Frames"
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "The time the BS shall wait after transmitting a DCD message
45         with an incremented Configuration Change Count before
46         issuing a DL-MAP message referring to
47         Downlink_Burst_Profiles defined in that DCD message."
48    ::= { wmanIf2BsConfigurationEntry 4 }
49
50
51
52
53 wmanIf2BsInitialRangingInterval OBJECT-TYPE
54     SYNTAX      INTEGER(0..2000)
55     UNITS       "milliseconds"
56     MAX-ACCESS  read-write
57     STATUS      current
58     DESCRIPTION
59         "Time between Initial Ranging regions assigned by the BS
60         in ms."
61    ::= { wmanIf2BsConfigurationEntry 5 }
62
63
64
65 wmanIf2BsSsULMapProcTime OBJECT-TYPE

```

```

1      SYNTAX      Unsigned32 (200 .. 4294967295)
2      UNITS       "micro seconds"
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "Time provided between arrival of the last bit of a UL-MAP
7          at an SS and effectiveness of that map in us."
8      ::= { wmanIf2BsConfigurationEntry 6 }
9
10
11
12 wmanIf2BsSsRangRespProcTime OBJECT-TYPE
13     SYNTAX      Unsigned32 (10000 .. 4294967295)
14     UNITS       "micro seconds"
15     MAX-ACCESS  read-write
16     STATUS      current
17     DESCRIPTION
18         "Time allowed for an SS following receipt of a ranging
19         response before it is expected to reply to an invited
20         ranging request in us."
21     ::= { wmanIf2BsConfigurationEntry 7 }
22
23
24
25 wmanIf2BsT5Timeout OBJECT-TYPE
26     SYNTAX      INTEGER (0 .. 2000)
27     UNITS       "milliseconds"
28     MAX-ACCESS  read-write
29     STATUS      current
30     DESCRIPTION
31         "Wait for Uplink Channel Change Response in ms."
32     ::= { wmanIf2BsConfigurationEntry 8 }
33
34
35
36 wmanIf2BsT9Timeout OBJECT-TYPE
37     SYNTAX      INTEGER (300 .. 65535)
38     UNITS       "milliseconds"
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "Registration Timeout, the time allowed between the BS
43         sending a RNG-RSP (success) to an SS, and receiving a
44         SBC-REQ from that same SS in ms."
45     ::= { wmanIf2BsConfigurationEntry 9 }
46
47
48
49 wmanIf2BsT13Timeout OBJECT-TYPE
50     SYNTAX      INTEGER (15 .. 65535)
51     UNITS       "minutes"
52     MAX-ACCESS  read-write
53     STATUS      current
54     DESCRIPTION
55         "The time allowed for an SS, following receipt of a
56         REG-RSP message to send a TFTP-CPLT message to the BS
57         in min."
58     ::= { wmanIf2BsConfigurationEntry 10 }
59
60
61
62 wmanIf2BsT15Timeout OBJECT-TYPE
63     SYNTAX      INTEGER (20 .. 65535)
64     UNITS       "milliseconds"
65

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "Wait for MCA-RSP in ms."
5      ::= { wmanIf2BsConfigurationEntry 11 }
6
7
8      wmanIf2BsT17Timeout OBJECT-TYPE
9          SYNTAX      INTEGER (5 .. 65535)
10         UNITS        "minutes"
11         MAX-ACCESS  read-write
12         STATUS      current
13         DESCRIPTION
14             "Time allowed for SS to complete SS Authorization and
15             Key Exchange in minutes."
16         ::= { wmanIf2BsConfigurationEntry 12 }
17
18
19
20         wmanIf2BsT27IdleTimer OBJECT-TYPE
21             SYNTAX      Unsigned32 (10000 .. 4294967295)
22             UNITS        "us"
23             MAX-ACCESS  read-write
24             STATUS      current
25             DESCRIPTION
26                 "Maximum time between unicast grants to SS when BS believes
27                 SS uplink transmission quality is good enough."
28             ::= { wmanIf2BsConfigurationEntry 13 }
29
30
31
32         wmanIf2BsT27ActiveTimer OBJECT-TYPE
33             SYNTAX      Unsigned32 (10000 .. 4294967295)
34             UNITS        "us"
35             MAX-ACCESS  read-write
36             STATUS      current
37             DESCRIPTION
38                 "Maximum time between unicast grants to SS when BS believes
39                 SS uplink transmission quality is not good enough."
40             ::= { wmanIf2BsConfigurationEntry 14 }
41
42
43
44
45         wmanIf2Bs2ndMgmtDlQoSProfileIndex OBJECT-TYPE
46             SYNTAX      INTEGER (1..65535)
47             MAX-ACCESS  read-write
48             STATUS      current
49             DESCRIPTION
50                 "This object defines the index of a row in
51                 wmanIf2BsServiceClassTable which is used to obtain all QoS
52                 parameters required for the BS downlink scheduler to
53                 properly allocate and manage the bandwidth and schedule
54                 the 2nd Management Connection traffic. The 2nd Management
55                 Connection traffic doesn't differ from Traffic Connection
56                 traffic in the area of QoS management."
57             ::= { wmanIf2BsConfigurationEntry 15 }
58
59
60
61
62         wmanIf2Bs2ndMgmtUlQoSProfileIndex OBJECT-TYPE
63             SYNTAX      INTEGER (1..65535)
64             MAX-ACCESS  read-write
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object defines the index of a row in
4          wmanIf2BsServiceClassTable which is used to obtain all QoS
5          parameters required for the BS uplink scheduler to
6          properly allocate and manage the bandwidth and schedule
7          the 2nd Management Connection traffic. The 2nd Management
8          Connection traffic doesn't differ from Traffic Connection
9          traffic in the area of QoS management."
10     ::= { wmanIf2BsConfigurationEntry 16 }
11
12
13
14 wmanIf2BsAutoSfidEnabled OBJECT-TYPE
15     SYNTAX      INTEGER {autoSfidDisabled(0),
16                 autoSfidEnabled(1)}
17     MAX-ACCESS  read-write
18     STATUS      current
19     DESCRIPTION
20         "This object defines whether the BS is allowed to
21         autonomously allocate SFIDs. When the object is set to
22         autoSfidEnabled, the BS is allowed to autonomously allocate
23         SFIDs from the range of allowed values defined by
24         wmanIf2BsConfigExtAutoSfidRangeMin and
25         wmanIf2BsConfigExtAutoSfidRangeMax. A SF is created
26         autonomously when it has not been provisioned in the
27         wmanIf2BsProvisionedSfTable and may be initiated by either
28         the SS or BS. The BS should always initiate SF creation
29         based on the provisioned Service flows configured in
30         wmanIf2BsProvisionedSfTable."
31     REFERENCE
32         "Subclause 11.13.1 in IEEE Std 802.16-2004"
33     ::= { wmanIf2BsConfigurationEntry 17 }
34
35
36
37
38
39 wmanIf2BsAutoSfidRangeMin OBJECT-TYPE
40     SYNTAX      Unsigned32 ( 1 .. 4294967295)
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "This object defines the minimum value of the range of SFID
45         values allocated for the BS sector for the purpose of
46         autonomous creation of service flows. This value is used
47         when the object wmanIf2BsAutoSfidEnabled allows
48         autonomous creation of SFIDs."
49     REFERENCE
50         "Subclause 11.13.1 in IEEE Std 802.16-2004"
51     ::= { wmanIf2BsConfigurationEntry 18 }
52
53
54
55
56
57 wmanIf2BsAutoSfidRangeMax OBJECT-TYPE
58     SYNTAX      Unsigned32 ( 1 .. 4294967295)
59     MAX-ACCESS  read-write
60     STATUS      current
61     DESCRIPTION
62         "This object defines the maximum value of the range of SFID
63         values allocated for the BS sector for the purpose of
64         autonomous creation of the service flows. This value is
65

```

```

1         used when the object wmanIf2BsAutoSfidEnabled allows
2         autonomous creation of SFIDs."
3
4     REFERENCE
5         "Subclause 11.13.1 in IEEE Std 802.16-2004"
6     ::= { wmanIf2BsConfigurationEntry 19 }
7
8
9     wmanIf2BsAasChanFbckReqFreq OBJECT-TYPE
10        SYNTAX      INTEGER (5..10000)
11        UNITS        "ms"
12        MAX-ACCESS   read-write
13        STATUS       current
14        DESCRIPTION
15            "This object defines AAS channel feedback request frequency.
16             It controls the frequency of downlink beam measurements.
17             The relevant MAC messages are AAS-FBCK-REQ/RSP"
18        REFERENCE
19            "Subclause 6.3.2.3.40 in IEEE Std 802.16-2004"
20        ::= { wmanIf2BsConfigurationEntry 20 }
21
22
23
24     wmanIf2BsAasBeamSelectFreq OBJECT-TYPE
25        SYNTAX      INTEGER (5..10000)
26        UNITS        "ms"
27        MAX-ACCESS   read-write
28        STATUS       current
29        DESCRIPTION
30            "This object defines AAS beam select frequency.
31             It controls how often SS issues beam select messages.
32             The relevant MAC message is AAS_Beam_Select"
33        REFERENCE
34            "Subclause 6.3.2.3.41 in IEEE Std 802.16-2004"
35        ::= { wmanIf2BsConfigurationEntry 21 }
36
37
38
39     wmanIf2BsAasChanFbckReqResolution OBJECT-TYPE
40        SYNTAX      INTEGER { aasChanFbckRes00(0),
41                               aasChanFbckRes01(1),
42                               aasChanFbckRes10(2),
43                               aasChanFbckRes11(3) }
44        MAX-ACCESS   read-write
45        STATUS       current
46        DESCRIPTION
47            "This object defines AAS feedback request frequency
48             measurements resolution. It is coded as follows:
49             aasChanFbckRes00 - every 4th carrier
50                               (-100, -96, -92, .., 100)
51             aasChanFbckRes01 - every 8th carrier
52                               (-100, -92, -84, .., 100)
53             aasChanFbckRes10 - every 16th carrier
54                               (-100, -84, -68, .., 100)
55             aasChanFbckRes11 - every 32th carrier
56                               (-100, -68, -36, .., 100)"
57        REFERENCE
58            "Subclause 8.3.6.4 in IEEE Std 802.16-2004"
59        ::= { wmanIf2BsConfigurationEntry 22 }
60
61
62
63
64
65

```

```

1  wmanIf2BsAasBeamReqResolution OBJECT-TYPE
2      SYNTAX      INTEGER {aasBeamReqRes000(0),
3                      aasBeamReqRes001(1),
4                      aasBeamReqRes010(2),
5                      aasBeamReqRes011(3),
6                      aasBeamReqRes100(4)}
7
8      MAX-ACCESS  read-write
9      STATUS      current
10     DESCRIPTION
11         "This object defines AAS beam select request resolution
12         parameter. It is coded as follows:
13         aasBeamReqRes000 - every 4th carrier
14         aasBeamReqRes001 - every 8th carrier
15         aasBeamReqRes010 - every 16th carrier
16         aasBeamReqRes011 - every 32th carrier
17         aasBeamReqRes100 - every 64th carrier"
18     REFERENCE
19         "Subclause 8.3.6.5 in IEEE Std 802.16-2004"
20     ::= { wmanIf2BsConfigurationEntry 23 }
21
22
23
24
25  wmanIf2BsAasNumOptDiversityZones OBJECT-TYPE
26      SYNTAX      INTEGER (0..65535)
27      MAX-ACCESS  read-write
28      STATUS      current
29      DESCRIPTION
30         "This object defines the number of optional diversity zones
31         transmitted in downlink."
32     REFERENCE
33         "Figure 209 in IEEE Std 802.16-2004"
34     ::= { wmanIf2BsConfigurationEntry 24 }
35
36
37
38
39  wmanIf2BsResetSector OBJECT-TYPE
40      SYNTAX      INTEGER {actionResetSectorNoAction(0),
41                      actionResetSector(1)}
42      MAX-ACCESS  read-write
43      STATUS      current
44      DESCRIPTION
45         "This object should be implemented as follows:
46         - When set to actionsResetSector value, instructs BS to
47         reset the sector identified by ifIndex. As a result of
48         this action the Phy and Mac of this sector should be
49         reinitialised.
50         - When set to value different than actionsResetSector it
51         should be ignored
52         - When read it should return actionsResetSectorNoAction"
53     ::= { wmanIf2BsConfigurationEntry 25 }
54
55
56
57
58  --
59  -- Base Station Channel Measurement Table
60  --
61  wmanIf2BsChannelMeasurementTable OBJECT-TYPE
62      SYNTAX      SEQUENCE OF WmanIf2BsChannelMeasurementEntry
63      MAX-ACCESS  not-accessible
64      STATUS      current
65

```



```

1      DESCRIPTION
2          "This table contains channel measurement information as
3          derived from BS measurement of uplink signal from SS,
4          and the downlink signal as reported from SS using
5          REP-REQ/RSP messages. The table shall be maintained as
6          FIFO to store measurement samples that can be used to
7          create RSSI and CINR histogram report. When the
8          measurement entry for a SS reaches the limit, the oldest
9          entry shall be deleted as the new entry is added to the
10         table."
11
12     REFERENCE
13         "6.3.2.3.33 in IEEE Std 802.16-2004"
14     ::= { wmanIf2BsCps 3 }
15
16 wmanIf2BsChannelMeasurementEntry OBJECT-TYPE
17     SYNTAX      WmanIf2BsChannelMeasurementEntry
18     MAX-ACCESS  not-accessible
19     STATUS      current
20     DESCRIPTION
21         "Each entry in the table contains RSSI and CINR
22         signal quality measurement on signal received from the SS.
23         The primary index is the ifIndex with ifType of
24         ieee80216WMAN identifying the BS sector.
25         wmanIf2BsSsMacAddress identifies the SS from which the
26         signal was received. wmanIf2BsChannelDirection is the
27         index to the direction of the channel.
28         wmanIf2BsHistogramIndex is the index to histogram samples.
29         Since there is no time stamp in the table,
30         wmanIf2BsHistogramIndex should be increased monotonically,
31         and wraps around when it reaches the implementation
32         specific limit."
33     INDEX      { ifIndex,
34                 wmanIf2BsSsMacAddress,
35                 wmanIf2BsChannelDirection,
36                 wmanIf2BsHistogramIndex }
37     ::= { wmanIf2BsChannelMeasurementTable 1 }
38
39 WmanIf2BsChannelMeasurementEntry ::= SEQUENCE {
40     wmanIf2BsChannelDirection      INTEGER,
41     wmanIf2BsHistogramIndex        Unsigned32,
42     wmanIf2BsChannelNumber         WmanIf2ChannelNumber,
43     wmanIf2BsStartFrame            INTEGER,
44     wmanIf2BsDuration               INTEGER,
45     wmanIf2BsBasicReport            BITS,
46     wmanIf2BsMeanCinrReport         INTEGER,
47     wmanIf2BsMeanRssiReport         INTEGER,
48     wmanIf2BsStdDeviationCinrReport INTEGER,
49     wmanIf2BsStdDeviationRssiReport INTEGER }
50
51 wmanIf2BsChannelDirection OBJECT-TYPE
52     SYNTAX      INTEGER { downstream(1),
53                        upstream(2) }
54     MAX-ACCESS  not-accessible
55     STATUS      current
56
57
58
59
60
61
62
63
64
65

```

```

1      DESCRIPTION
2          "wmanIf2BsChannelDirection identifies the direction of a
3          a channel where the measurement takes place."
4      ::= { wmanIf2BsChannelMeasurementEntry 1 }
5
6
7      wmanIf2BsHistogramIndex OBJECT-TYPE
8          SYNTAX      Unsigned32 (1 .. 4294967295)
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "wmanIf2BsHistogramIndex identifies the histogram samples
13             in the table for each subscriber station."
14         ::= { wmanIf2BsChannelMeasurementEntry 2 }
15
16
17         wmanIf2BsChannelNumber OBJECT-TYPE
18             SYNTAX      WmanIf2ChannelNumber
19             MAX-ACCESS  read-only
20             STATUS      current
21             DESCRIPTION
22                 "Physical channel number to be reported on is only
23                 applicable to license exempt band. For licensed band,
24                 this parameter should be null."
25             REFERENCE
26                 "Subclause 11.12 in IEEE Std 802.16-2004"
27             ::= { wmanIf2BsChannelMeasurementEntry 3 }
28
29
30         wmanIf2BsStartFrame OBJECT-TYPE
31             SYNTAX      INTEGER (0..65535)
32             MAX-ACCESS  read-only
33             STATUS      current
34             DESCRIPTION
35                 "Frame number in which measurement for this channel
36                 started."
37             REFERENCE
38                 "Subclause 11.12 in IEEE Std 802.16-2004"
39             ::= { wmanIf2BsChannelMeasurementEntry 4 }
40
41
42         wmanIf2BsDuration OBJECT-TYPE
43             SYNTAX      INTEGER (0 .. 16777215)
44             MAX-ACCESS  read-only
45             STATUS      current
46             DESCRIPTION
47                 "Cumulative measurement duration on the channel in
48                 multiples of Ts. For any value exceeding 0xFFFFF,
49                 report 0xFFFFF."
50             REFERENCE
51                 "Subclause 11.12 in IEEE Std 802.16-2004"
52             ::= { wmanIf2BsChannelMeasurementEntry 5 }
53
54
55         wmanIf2BsBasicReport OBJECT-TYPE
56             SYNTAX      BITS { wirelessHuman(0),
57                         unknownTransmission(1),
58                         primaryUser(2),
59                         channelNotMeasured(3) }
60
61
62
63
64
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "Bit #0: WirelessHUMAN detected on the channel
5          Bit #1: Unknown transmissions detected on the channel
6          Bit #2: Primary User detected on the channel
7          Bit #3: Unmeasured. Channel not measured"
8      REFERENCE
9          "Subclause 11.12 in IEEE Std 802.16-2004"
10     ::= { wmanIf2BsChannelMeasurementEntry 6 }
11
12
13
14     wmanIf2BsMeanCinrReport OBJECT-TYPE
15         SYNTAX      INTEGER (-20 .. 37)
16         UNITS        "dB"
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "Mean CINR report."
21         REFERENCE
22             "Subclause 8.2.2, 8.3.9 in IEEE Std 802.16-2004"
23         ::= { wmanIf2BsChannelMeasurementEntry 7 }
24
25
26
27     wmanIf2BsMeanRssiReport OBJECT-TYPE
28         SYNTAX      INTEGER (-123 .. -40)
29         UNITS        "dBm"
30         MAX-ACCESS  read-only
31         STATUS      current
32         DESCRIPTION
33             "Mean RSSI report."
34         REFERENCE
35             "Subclause 8.2.2, 8.3.9 in IEEE Std 802.16-2004"
36         ::= { wmanIf2BsChannelMeasurementEntry 8 }
37
38
39
40
41     wmanIf2BsStdDeviationCinrReport OBJECT-TYPE
42         SYNTAX      INTEGER (0 .. 29)
43         UNITS        "dB"
44         MAX-ACCESS  read-only
45         STATUS      current
46         DESCRIPTION
47             "Standard deviation CINR report."
48         REFERENCE
49             "Subclause 8.2.2, 8.3.9 in IEEE Std 802.16-2004"
50         ::= { wmanIf2BsChannelMeasurementEntry 9 }
51
52
53
54     wmanIf2BsStdDeviationRssiReport OBJECT-TYPE
55         SYNTAX      INTEGER (0 .. 42)
56         UNITS        "dB"
57         MAX-ACCESS  read-only
58         STATUS      current
59         DESCRIPTION
60             "Standard deviation RSSI report."
61         REFERENCE
62             "Subclause 8.2.2, 8.3.9 in IEEE Std 802.16-2004"
63         ::= { wmanIf2BsChannelMeasurementEntry 10 }
64
65

```

```

1
2
3 --
4 -- Base Station capabilities
5 --
6 wmanIf2BsCapabilities OBJECT IDENTIFIER ::= { wmanIf2BsCps 4 }
7
8
9 wmanIf2BsSsReqCapabilitiesTable OBJECT-TYPE
10     SYNTAX      SEQUENCE OF WmanIf2BsSsReqCapabilitiesEntry
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "This table contains the basic capability information of SSs
15         that have been reported by SSs to BS using RNG-REQ, SBC-REQ
16         and REG-REQ messages. Entries in this table should be
17         created when an SS registers with a BS."
18     ::= { wmanIf2BsCapabilities 1 }
19
20
21 wmanIf2BsSsReqCapabilitiesEntry OBJECT-TYPE
22     SYNTAX      WmanIf2BsSsReqCapabilitiesEntry
23     MAX-ACCESS  not-accessible
24     STATUS      current
25     DESCRIPTION
26         "This table provides one row for each SS that has been
27         registered in the BS. This table augments the table
28         wmanIf2BsRegisteredSsTable."
29     AUGMENTS { wmanIf2BsRegisteredSsEntry }
30     ::= { wmanIf2BsSsReqCapabilitiesTable 1 }
31
32
33
34
35 WmanIf2BsSsReqCapabilitiesEntry ::= SEQUENCE {
36     wmanIf2BsSsReqCapUplinkCidSupport      WmanIf2NumOfCid,
37     wmanIf2BsSsReqCapArqSupport            WmanIf2ArqSupportType,
38     wmanIf2BsSsReqCapDsxFlowControl        WmanIf2MaxDsxFlowType,
39     wmanIf2BsSsReqCapMacCrcSupport         WmanIf2MacCrcSupport,
40     wmanIf2BsSsReqCapMcaFlowControl        WmanIf2MaxMcaFlowType,
41     wmanIf2BsSsReqCapMcpGroupCidSupport    WmanIf2MaxMcpGroupCid,
42     wmanIf2BsSsReqCapPkmFlowControl        WmanIf2MaxPkmFlowType,
43     wmanIf2BsSsReqCapAuthPolicyControl     WmanIf2AuthPolicyType,
44     wmanIf2BsSsReqCapMaxNumOfSupportedSA  WmanIf2MaxNumOfSaType,
45     wmanIf2BsSsReqCapIpVersion            WmanIf2IpVersionType,
46     wmanIf2BsSsReqCapMacCsSupportBitMap   WmanIf2MacCsBitMap,
47     wmanIf2BsSsReqCapMaxNumOfClassifier   WmanIf2MaxClassifiers,
48     wmanIf2BsSsReqCapPhsSupport           WmanIf2PhsSupportType,
49     wmanIf2BsSsReqCapBandwidthAllocSupport WmanIf2BwAllocSupport,
50     wmanIf2BsSsReqCapPduConstruction      WmanIf2PduConstruction,
51     wmanIf2BsSsReqCapTtgTransitionGap      WmanIf2SsTransitionGap,
52     wmanIf2BsSsReqCapRtgTransitionGap      WmanIf2SsTransitionGap,
53     wmanIf2BsSsReqCapDownlinkCidSupport    WmanIf2NumOfCid,
54     wmanIf2BsSsReqCapPackingSupport        WmanIf2PackingSupport,
55     wmanIf2BsSsReqCapExtendedRtpsSupport  WmanIf2ExtRtpsSupport,
56     wmanIf2BsSsReqCapMaxNumBurstToMs      INTEGER,
57     wmanIf2BsSsReqCapIpAddrAllocMethod     WmanIf2IpAllocMethod,
58     wmanIf2BsSsReqCapArqAckType           WmanIf2ArqAckType,
59     wmanIf2BsSsReqCapMacHeader            WmanIf2MacHeaderSupp,
60     wmanIf2BsSsReqCapMaxMacLevelDlFrame   WmanIf2MaxMacLevel,
61
62
63
64
65

```

```

1         wmanIf2BsSsReqCapMaxMacLevelUlFrame      WmanIf2MaxMacLevel}
2
3
4 wmanIf2BsSsReqCapUplinkCidSupport OBJECT-TYPE
5     SYNTAX      WmanIf2NumOfCid
6     MAX-ACCESS  read-only
7     STATUS      current
8     DESCRIPTION
9         "This object shows the number of Uplink CIDs the SS can
10        support."
11
12     ::= { wmanIf2BsSsReqCapabilitiesEntry 1 }
13
14 wmanIf2BsSsReqCapArqSupport OBJECT-TYPE
15     SYNTAX      WmanIf2ArqSupportType
16     MAX-ACCESS  read-only
17     STATUS      current
18     DESCRIPTION
19         "This object indicates whether the SS supports ARQ."
20
21     ::= { wmanIf2BsSsReqCapabilitiesEntry 2 }
22
23
24 wmanIf2BsSsReqCapDsxFlowControl OBJECT-TYPE
25     SYNTAX      WmanIf2MaxDsxFlowType
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "This object specifies the maximum number of concurrent
30         DSA, DSC, or DSD transactions that SS is capable of having
31         outstanding."
32
33     DEFVAL      { 0 }
34
35     ::= { wmanIf2BsSsReqCapabilitiesEntry 3 }
36
37 wmanIf2BsSsReqCapMacCrcSupport OBJECT-TYPE
38     SYNTAX      WmanIf2MacCrcSupport
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "This object indicates whether or not the SS supports MAC
43         level CRC."
44
45     DEFVAL      { macCrcSupport }
46
47     ::= { wmanIf2BsSsReqCapabilitiesEntry 4 }
48
49 wmanIf2BsSsReqCapMcaFlowControl OBJECT-TYPE
50     SYNTAX      WmanIf2MaxMcaFlowType
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "This object specifies the maximum number of concurrent MCA
55         transactions that SS is capable of having outstanding."
56
57     DEFVAL      { 0 }
58
59     ::= { wmanIf2BsSsReqCapabilitiesEntry 5 }
60
61 wmanIf2BsSsReqCapMcpGroupCidSupport OBJECT-TYPE
62     SYNTAX      WmanIf2MaxMcpGroupCid
63     MAX-ACCESS  read-only
64     STATUS      current
65

```

```

1      DESCRIPTION
2          "This object indicates the maximum number of
3          simultaneous Multicast Polling Groups the SS is
4          capable of belonging to."
5
6      DEFVAL    { 0 }
7      ::= { wmanIf2BsSsReqCapabilitiesEntry 6 }
8
9
10     wmanIf2BsSsReqCapPkmFlowControl OBJECT-TYPE
11         SYNTAX      WmanIf2MaxPkmFlowType
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "This object specifies the maximum number of concurrent PKM
16             transactions that SS is capable of having outstanding."
17
18         DEFVAL    { 0 }
19         ::= { wmanIf2BsSsReqCapabilitiesEntry 7 }
20
21
22     wmanIf2BsSsReqCapAuthPolicyControl OBJECT-TYPE
23         SYNTAX      WmanIf2AuthPolicyType
24         MAX-ACCESS  read-only
25         STATUS      current
26         DESCRIPTION
27             "This object specifies authorization policy that SS is
28             capable of. A bit value of 0 = 'not supported', 1 =
29             'supported'. If this field is omitted, then both SS and
30             BS shall use the IEEE 802.16 security, constituting X.509
31             digital certificates and the RSA public key encryption
32             algorithm, as authorization policy."
33
34         ::= { wmanIf2BsSsReqCapabilitiesEntry 8 }
35
36
37     wmanIf2BsSsReqCapMaxNumOfSupportedSA OBJECT-TYPE
38         SYNTAX      WmanIf2MaxNumOfSaType
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "This field specifies the maximum number of supported
43             security associations of the SS."
44
45         DEFVAL    { 1 }
46         ::= { wmanIf2BsSsReqCapabilitiesEntry 9 }
47
48
49     wmanIf2BsSsReqCapIpVersion OBJECT-TYPE
50         SYNTAX      WmanIf2IpVersionType
51         MAX-ACCESS  read-only
52         STATUS      current
53         DESCRIPTION
54             "This object indicates the version of IP used on the 2nd
55             Management Connection. The value should be undefined
56             if the 2nd management CID doesn't exist."
57
58         ::= { wmanIf2BsSsReqCapabilitiesEntry 10 }
59
60
61     wmanIf2BsSsReqCapMacCsSupportBitMap OBJECT-TYPE
62         SYNTAX      WmanIf2MacCsBitMap
63         MAX-ACCESS  read-only
64         STATUS      current
65

```

```

1      DESCRIPTION
2          "This object indicates SS reported set of MAC convergence
3          sublayer support. When a bit is set, it indicates
4          the corresponding CS feature is supported."
5      ::= { wmanIf2BsSsReqCapabilitiesEntry 11 }
6
7
8      wmanIf2BsSsReqCapMaxNumOfClassifier OBJECT-TYPE
9          SYNTAX      WmanIf2MaxClassifiers
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "This object indicates the maximum number of admitted
14             Classifiers that the SS can support."
15         DEFVAL      { 0 }
16         ::= { wmanIf2BsSsReqCapabilitiesEntry 12 }
17
18
19
20         wmanIf2BsSsReqCapPhsSupport OBJECT-TYPE
21             SYNTAX      WmanIf2PhsSupportType
22             MAX-ACCESS  read-only
23             STATUS      current
24             DESCRIPTION
25                 "This object indicates indicates the level of SS support
26                 for PHS."
27             DEFVAL      { noPhsSupport }
28             ::= { wmanIf2BsSsReqCapabilitiesEntry 13 }
29
30
31
32         wmanIf2BsSsReqCapBandwidthAllocSupport OBJECT-TYPE
33             SYNTAX      WmanIf2BwAllocSupport
34             MAX-ACCESS  read-only
35             STATUS      current
36             DESCRIPTION
37                 "This field indicates the bandwidth allocation
38                 capabilities of the SS. The usage is defined by
39                 WmanIf2BwAllocSupport."
40             ::= { wmanIf2BsSsReqCapabilitiesEntry 14 }
41
42
43
44         wmanIf2BsSsReqCapPduConstruction OBJECT-TYPE
45             SYNTAX      WmanIf2PduConstruction
46             MAX-ACCESS  read-only
47             STATUS      current
48             DESCRIPTION
49                 "This field indicates the SS's capabilities for
50                 construction and transmission of MAC PDUs. The usage
51                 is defined by WmanIf2PduConstruction."
52             ::= { wmanIf2BsSsReqCapabilitiesEntry 15 }
53
54
55
56         wmanIf2BsSsReqCapTtgTransitionGap OBJECT-TYPE
57             SYNTAX      WmanIf2SsTransitionGap
58             UNITS        "us"
59             MAX-ACCESS  read-only
60             STATUS      current
61             DESCRIPTION
62                 "This field indicates the SS's transition speed SSTTG
63                 for TDD and H-FDD SSs. The usage is defined by
64
65

```

```

1         WmanIf2SsTransitionGap."
2         ::= { wmanIf2BsSsReqCapabilitiesEntry 16 }
3
4
5 wmanIf2BsSsReqCapRtgTransitionGap OBJECT-TYPE
6     SYNTAX      WmanIf2SsTransitionGap
7     UNITS       "us"
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11        "This field indicates the SS's transition speed SSRTG
12        for TDD and H-FDD SSs. The usage is defined by
13        WmanIf2SsTransitionGap."
14    ::= { wmanIf2BsSsReqCapabilitiesEntry 17 }
15
16
17
18 wmanIf2BsSsReqCapDownlinkCidSupport OBJECT-TYPE
19     SYNTAX      WmanIf2NumOfCid
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23        "This object shows the number of Downlink CIDs the SS can
24        support."
25    ::= { wmanIf2BsSsReqCapabilitiesEntry 18 }
26
27
28
29 wmanIf2BsSsReqCapPackingSupport OBJECT-TYPE
30     SYNTAX      WmanIf2PackingSupport
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34        "Indicates the availability of MS support for Packing."
35     REFERENCE
36        "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
37    ::= { wmanIf2BsSsReqCapabilitiesEntry 19 }
38
39
40
41 wmanIf2BsSsReqCapExtendedRtptsSupport OBJECT-TYPE
42     SYNTAX      WmanIf2ExtRtptsSupport
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46        "Indicates the availability of SS support for extended
47        rtPs."
48     REFERENCE
49        "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
50    ::= { wmanIf2BsSsReqCapabilitiesEntry 20 }
51
52
53
54 wmanIf2BsSsReqCapMaxNumBurstToMs OBJECT-TYPE
55     SYNTAX      INTEGER (1..16)
56     MAX-ACCESS  read-only
57     STATUS      current
58     DESCRIPTION
59        "Maximum number of bursts transmitted concurrently to the MS
60        , including all bursts without CID or with CIDs matching
61        the MS's CIDs."
62     REFERENCE
63        "Subclause 11.7.8.13 in IEEE Std 802.16e-2005"
64
65

```



```

1      ::= { wmanIf2BsSsReqCapabilitiesEntry 21 }
2
3
4  wmanIf2BsSsReqCapIpAddrAllocMethod OBJECT-TYPE
5      SYNTAX      WmanIf2IpAllocMethod
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "Indicates the method of allocating IP address for the
10         secondary management connection."
11
12     REFERENCE
13         "Subclause 11.7.11 in IEEE Std 802.16e-2005"
14     ::= { wmanIf2BsSsReqCapabilitiesEntry 22 }
15
16
17  wmanIf2BsSsReqCapArqAckType OBJECT-TYPE
18     SYNTAX      WmanIf2ArqAckType
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "The value of this parameter specifies the ARQ ACK type
23         supported by the MS."
24
25     REFERENCE
26         "Subclause 11.7.23 in IEEE Std 802.16e-2005"
27     ::= { wmanIf2BsSsReqCapabilitiesEntry 23 }
28
29
30  wmanIf2BsSsReqCapMacHeader OBJECT-TYPE
31     SYNTAX      WmanIf2MacHeaderSupp
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "Indicates whether or not the MS and BS support various
36         types of MAC header and extended subheaders."
37
38     REFERENCE
39         "Subclause 11.7.25 in IEEE Std 802.16e-2005"
40     ::= { wmanIf2BsSsReqCapabilitiesEntry 24 }
41
42
43  wmanIf2BsSsReqCapMaxMacLevelDlFrame OBJECT-TYPE
44     SYNTAX      WmanIf2MaxMacLevel
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "Maximum amount of MAC level data the MS is capable of
49         processing per DL frame."
50
51     REFERENCE
52         "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
53     DEFVAL      { 0 }
54     ::= { wmanIf2BsSsReqCapabilitiesEntry 25 }
55
56
57  wmanIf2BsSsReqCapMaxMacLevelUlFrame OBJECT-TYPE
58     SYNTAX      WmanIf2MaxMacLevel
59     MAX-ACCESS  read-only
60     STATUS      current
61     DESCRIPTION
62         "Maximum amount of MAC level data the MS is capable of
63         processing per UL frame."
64
65

```

```

1      REFERENCE
2          "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
3      DEFVAL      { 0 }
4      ::= { wmanIf2BsSsReqCapabilitiesEntry 26 }
5
6
7      wmanIf2BsSsRspCapabilitiesTable OBJECT-TYPE
8          SYNTAX      SEQUENCE OF WmanIf2BsSsRspCapabilitiesEntry
9          MAX-ACCESS  not-accessible
10         STATUS      current
11         DESCRIPTION
12             "This table contains the basic capability information of SSS
13             that have been negotiated and agreed between BS and SS via
14             RNG-REQ/RSP, SBC-REQ/RSP and REG-REQ/RSP messages.
15             This table augments the wmanIf2BsRegisteredSsTable."
16         REFERENCE
17             "Subclause 6.3.2.3.7 in IEEE Std 802.16-2004"
18         ::= { wmanIf2BsCapabilities 2 }
19
20
21
22
23     wmanIf2BsSsRspCapabilitiesEntry OBJECT-TYPE
24         SYNTAX      WmanIf2BsSsRspCapabilitiesEntry
25         MAX-ACCESS  not-accessible
26         STATUS      current
27         DESCRIPTION
28             "This table provides one row for each SS that has been
29             registered in the BS. This table augments the
30             wmanIf2BsRegisteredSsTable. "
31         AUGMENTS { wmanIf2BsRegisteredSsEntry }
32         ::= { wmanIf2BsSsRspCapabilitiesTable 1 }
33
34
35
36     WmanIf2BsSsRspCapabilitiesEntry ::= SEQUENCE {
37         wmanIf2BsSsRspCapUplinkCidSupport      WmanIf2NumOfCid,
38         wmanIf2BsSsRspCapArqSupport           WmanIf2ArqSupportType,
39         wmanIf2BsSsRspCapDsxFlowControl       WmanIf2MaxDsxFlowType,
40         wmanIf2BsSsRspCapMacCrcSupport        WmanIf2MacCrcSupport,
41         wmanIf2BsSsRspCapMcaFlowControl       WmanIf2MaxMcaFlowType,
42         wmanIf2BsSsRspCapMcpGroupCidSupport   WmanIf2MaxMcpGroupCid,
43         wmanIf2BsSsRspCapPkmFlowControl       WmanIf2MaxPkmFlowType,
44         wmanIf2BsSsRspCapAuthPolicyControl    WmanIf2AuthPolicyType,
45         wmanIf2BsSsRspCapMaxNumOfSupportedSA  WmanIf2MaxNumOfSaType,
46         wmanIf2BsSsRspCapIpVersion           WmanIf2IpVersionType,
47         wmanIf2BsSsRspCapMacCsSupportBitMap   WmanIf2MacCsBitMap,
48         wmanIf2BsSsRspCapMaxNumOfClassifier   WmanIf2MaxClassifiers,
49         wmanIf2BsSsRspCapPhsSupport           WmanIf2PhsSupportType,
50         wmanIf2BsSsRspCapBandwidthAllocSupport WmanIf2BwAllocSupport,
51         wmanIf2BsSsRspCapPduConstruction      WmanIf2PduConstruction,
52         wmanIf2BsSsRspCapTtgTransitionGap     WmanIf2SsTransitionGap,
53         wmanIf2BsSsRspCapRtgTransitionGap     WmanIf2SsTransitionGap,
54         wmanIf2BsSsRspCapDownlinkCidSupport   WmanIf2NumOfCid,
55         wmanIf2BsSsRspCapPackingSupport       WmanIf2PackingSupport,
56         wmanIf2BsSsRspCapExtendedRtpsSupport  WmanIf2ExtRtpsSupport,
57         wmanIf2BsSsRspCapMaxNumBurstToMs     INTEGER,
58         wmanIf2BsSsRspCapIpAddrAllocMethod    WmanIf2IpAllocMethod,
59         wmanIf2BsSsRspCapArqAckType          WmanIf2ArqAckType,
60         wmanIf2BsSsRspCapMacHeader           WmanIf2MacHeaderSupp,

```

```

1      wmanIf2BsSsRspCapMaxMacLevelDlFrame      WmanIf2MaxMacLevel,
2      wmanIf2BsSsRspCapMaxMacLevelUlFrame     WmanIf2MaxMacLevel,
3      wmanIf2BsSsRspCapNumOfProvisionedSf     Unsigned32}
4
5
6      wmanIf2BsSsRspCapUplinkCidSupport OBJECT-TYPE
7          SYNTAX      WmanIf2NumOfCid
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "Negotiated number of Uplink CIDs the SS can support."
12             ::= { wmanIf2BsSsRspCapabilitiesEntry 1 }
13
14
15     wmanIf2BsSsRspCapArqSupport OBJECT-TYPE
16         SYNTAX      WmanIf2ArqSupportType
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "This object indicates whether the SS is allowed to use ARQ
21             as a result of the capabilities negotiation."
22             ::= { wmanIf2BsSsRspCapabilitiesEntry 2 }
23
24
25
26     wmanIf2BsSsRspCapDsxFowControl OBJECT-TYPE
27         SYNTAX      WmanIf2MaxDsxFowType
28         MAX-ACCESS  read-only
29         STATUS      current
30         DESCRIPTION
31             "Negotiated maximum number of concurrent DSA, DSC, or DSD
32             transactions that may be outstanding."
33             ::= { wmanIf2BsSsRspCapabilitiesEntry 3 }
34
35
36
37     wmanIf2BsSsRspCapMacCrcSupport OBJECT-TYPE
38         SYNTAX      WmanIf2MacCrcSupport
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "This object indicates whether or not the SS is allowed to
43             use MAC level CRC as a result of the capabilities
44             negotiation."
45             DEFVAL   { macCrcSupport }
46             ::= { wmanIf2BsSsRspCapabilitiesEntry 4 }
47
48
49
50     wmanIf2BsSsRspCapMcaFlowControl OBJECT-TYPE
51         SYNTAX      WmanIf2MaxMcaFlowType
52         MAX-ACCESS  read-only
53         STATUS      current
54         DESCRIPTION
55             "Negotiated maximum number of concurrent
56             MCA transactions that may be outstanding."
57             DEFVAL   { 0 }
58             ::= { wmanIf2BsSsRspCapabilitiesEntry 5 }
59
60
61
62     wmanIf2BsSsRspCapMcpGroupCidSupport OBJECT-TYPE
63         SYNTAX      WmanIf2MaxMcpGroupCid
64         MAX-ACCESS  read-only
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "Negotiated maximum number of simultaneous Multicast
4          Polling Groups the SS is capable of belonging to."
5      DEFVAL      { 0 }
6      ::= { wmanIf2BsSsRspCapabilitiesEntry 6 }
7
8
9
10     wmanIf2BsSsRspCapPkmFlowControl OBJECT-TYPE
11         SYNTAX      WmanIf2MaxPkmFlowType
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "Negotiated maximum number of concurrent PKM
16             transactions that may be outstanding."
17         DEFVAL      { 0 }
18         ::= { wmanIf2BsSsRspCapabilitiesEntry 7 }
19
20
21
22     wmanIf2BsSsRspCapAuthPolicyControl OBJECT-TYPE
23         SYNTAX      WmanIf2AuthPolicyType
24         MAX-ACCESS  read-only
25         STATUS      current
26         DESCRIPTION
27             "This object specifies negotiated authorization policy.
28             A bit value of 0 = 'not supported', 1 = 'supported'. If
29             this field is omitted, then both SS and BS shall use the
30             IEEE 802.16 security, constituting X.509 digital
31             certificates and the RSA public key encryption
32             algorithm, as authorization policy."
33         ::= { wmanIf2BsSsRspCapabilitiesEntry 8 }
34
35
36
37     wmanIf2BsSsRspCapMaxNumOfSupportedSA OBJECT-TYPE
38         SYNTAX      WmanIf2MaxNumOfSaType
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "Negotiated maximum number of supported security
43             association of the SS."
44         DEFVAL      { 1 }
45         ::= { wmanIf2BsSsRspCapabilitiesEntry 9 }
46
47
48
49     wmanIf2BsSsRspCapIpVersion OBJECT-TYPE
50         SYNTAX      WmanIf2IpVersionType
51         MAX-ACCESS  read-only
52         STATUS      current
53         DESCRIPTION
54             "Negotiated version of IP used on the 2nd Management
55             Connection. The value should be undefined if the 2nd
56             management CID doesn't exist."
57         ::= { wmanIf2BsSsRspCapabilitiesEntry 10 }
58
59
60
61     wmanIf2BsSsRspCapMacCsSupportBitMap OBJECT-TYPE
62         SYNTAX      WmanIf2MacCsBitMap
63         MAX-ACCESS  read-only
64         STATUS      current
65

```

```

1      DESCRIPTION
2          "Negotiated set of MAC convergence sublayer support.
3          When a bit is set, it indicates the corresponding CS
4          feature is supported."
5      ::= { wmanIf2BsSsRspCapabilitiesEntry 11 }
6
7
8      wmanIf2BsSsRspCapMaxNumOfClassifier OBJECT-TYPE
9          SYNTAX      WmanIf2MaxClassifiers
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "Negotiated maximum number of admitted Classifiers
14             that the SS is allowed to have."
15         DEFVAL      { 0 }
16         ::= { wmanIf2BsSsRspCapabilitiesEntry 12 }
17
18
19
20     wmanIf2BsSsRspCapPhsSupport OBJECT-TYPE
21         SYNTAX      WmanIf2PhsSupportType
22         MAX-ACCESS  read-only
23         STATUS      current
24         DESCRIPTION
25             "This object indicates the negotiated level of PHS
26             support."
27         DEFVAL      { noPhsSupport }
28         ::= { wmanIf2BsSsRspCapabilitiesEntry 13 }
29
30
31
32     wmanIf2BsSsRspCapBandwidthAllocSupport OBJECT-TYPE
33         SYNTAX      WmanIf2BwAllocSupport
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "This field indicates negotiated properties of the SS
38             for bandwidth allocation purposes. The usage is defined
39             by WmanIf2BwAllocSupport."
40         ::= { wmanIf2BsSsRspCapabilitiesEntry 14 }
41
42
43
44     wmanIf2BsSsRspCapPduConstruction OBJECT-TYPE
45         SYNTAX      WmanIf2PduConstruction
46         MAX-ACCESS  read-only
47         STATUS      current
48         DESCRIPTION
49             "Specifies negotiated capabilities for construction and
50             transmission of MAC PDUs. The usage is defined by
51             WmanIf2PduConstruction."
52         ::= { wmanIf2BsSsRspCapabilitiesEntry 15 }
53
54
55
56     wmanIf2BsSsRspCapTtgTransitionGap OBJECT-TYPE
57         SYNTAX      WmanIf2SsTransitionGap
58         UNITS        "us"
59         MAX-ACCESS  read-only
60         STATUS      current
61         DESCRIPTION
62             "This field indicates the negotiated transition speed
63             SSTTG for TDD and H-FDD SSs. The usage is defined by
64
65

```

```

1         WmanIf2SsTransitionGap."
2         ::= { wmanIf2BsSsRspCapabilitiesEntry 16 }
3
4
5 wmanIf2BsSsRspCapRtgTransitionGap OBJECT-TYPE
6     SYNTAX      WmanIf2SsTransitionGap
7     UNITS       "us"
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11        "This field indicates the negotiated transition speed
12         SSRTG for TDD and H-FDD SSs. The usage is defined by
13         WmanIf2SsTransitionGap."
14    ::= { wmanIf2BsSsRspCapabilitiesEntry 17 }
15
16
17
18 wmanIf2BsSsRspCapDownlinkCidSupport OBJECT-TYPE
19     SYNTAX      WmanIf2NumOfCid
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23        "This object shows the number of Downlink CIDs the SS can
24         support."
25    ::= { wmanIf2BsSsRspCapabilitiesEntry 18 }
26
27
28
29 wmanIf2BsSsRspCapPackingSupport OBJECT-TYPE
30     SYNTAX      WmanIf2PackingSupport
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34        "Indicates the availability of MS support for Packing."
35     REFERENCE
36        "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
37    ::= { wmanIf2BsSsRspCapabilitiesEntry 19 }
38
39
40
41 wmanIf2BsSsRspCapExtendedRtPsSupport OBJECT-TYPE
42     SYNTAX      WmanIf2ExtRtPsSupport
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46        "Indicates the availability of SS support for extended
47         rtPs."
48     REFERENCE
49        "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
50    ::= { wmanIf2BsSsRspCapabilitiesEntry 20 }
51
52
53
54 wmanIf2BsSsRspCapMaxNumBurstToMs OBJECT-TYPE
55     SYNTAX      INTEGER (1..16)
56     MAX-ACCESS  read-only
57     STATUS      current
58     DESCRIPTION
59        "Maximum number of bursts transmitted concurrently to the MS
60         , including all bursts without CID or with CIDs matching
61         the MS CIDs."
62     REFERENCE
63        "Subclause 11.7.8.13 in IEEE Std 802.16e-2005"
64
65

```

```

1      ::= { wmanIf2BsSsRspCapabilitiesEntry 21 }
2
3
4      wmanIf2BsSsRspCapIpAddrAllocMethod OBJECT-TYPE
5          SYNTAX      WmanIf2IpAllocMethod
6          MAX-ACCESS  read-only
7          STATUS      current
8          DESCRIPTION
9              "Indicates the method of allocating IP address for the
10             secondary management connection."
11
12         REFERENCE
13             "Subclause 11.7.11 in IEEE Std 802.16e-2005"
14         ::= { wmanIf2BsSsRspCapabilitiesEntry 22 }
15
16
17         wmanIf2BsSsRspCapArqAckType OBJECT-TYPE
18             SYNTAX      WmanIf2ArqAckType
19             MAX-ACCESS  read-only
20             STATUS      current
21             DESCRIPTION
22                 "The value of this parameter specifies the ARQ ACK type
23                 supported by the MS."
24
25             REFERENCE
26                 "Subclause 11.7.23 in IEEE Std 802.16e-2005"
27             ::= { wmanIf2BsSsRspCapabilitiesEntry 23 }
28
29
30         wmanIf2BsSsRspCapMacHeader OBJECT-TYPE
31             SYNTAX      WmanIf2MacHeaderSupp
32             MAX-ACCESS  read-only
33             STATUS      current
34             DESCRIPTION
35                 "Indicates whether or not the MS and BS support various
36                 types of MAC header and extended subheaders."
37
38             REFERENCE
39                 "Subclause 11.7.25 in IEEE Std 802.16e-2005"
40             ::= { wmanIf2BsSsRspCapabilitiesEntry 24 }
41
42
43         wmanIf2BsSsRspCapMaxMacLevelDlFrame OBJECT-TYPE
44             SYNTAX      WmanIf2MaxMacLevel
45             MAX-ACCESS  read-only
46             STATUS      current
47             DESCRIPTION
48                 "Maximum amount of MAC level data the MS is capable of
49                 processing per DL frame. A value of 0 indicates such
50                 limitation does not exist, except the limitation of
51                 the physical medium"
52
53             REFERENCE
54                 "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
55             DEFVAL      { 0 }
56             ::= { wmanIf2BsSsRspCapabilitiesEntry 25 }
57
58
59         wmanIf2BsSsRspCapMaxMacLevelUlFrame OBJECT-TYPE
60             SYNTAX      WmanIf2MaxMacLevel
61             MAX-ACCESS  read-only
62             STATUS      current
63             DESCRIPTION
64
65

```

```

1         "Maximum amount of MAC level data the MS is capable of
2         processing per UL frame. A value of 0 indicates such
3         limitation does not exist, except the limitation of
4         the physical medium"
5
6     REFERENCE
7         "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
8
9     DEFVAL      { 0 }
10    ::= { wmanIf2BsSsRspCapabilitiesEntry 26 }
11
12    wmanIf2BsSsRspCapNumOfProvisionedSf OBJECT-TYPE
13        SYNTAX      Unsigned32 (0 .. 255)
14        MAX-ACCESS  read-only
15        STATUS      current
16        DESCRIPTION
17            "When a BS is to transmit multiple DSA transactions for
18            provisioned service flows, this object indicates how many
19            DSA transactions with provisioned service flows will be
20            transmitted."
21
22        REFERENCE
23            "Subclause 11.7.19 in IEEE Std 802.16e-2005"
24
25        ::= { wmanIf2BsSsRspCapabilitiesEntry 27 }
26
27
28    wmanIf2BsBasicCapabilitiesTable OBJECT-TYPE
29        SYNTAX      SEQUENCE OF WmanIf2BsBasicCapabilitiesEntry
30        MAX-ACCESS  not-accessible
31        STATUS      current
32        DESCRIPTION
33            "This table contains the basic capabilities of the BS as
34            implemented in BS hardware and software. These capabilities
35            along with the configuration for them
36            (wmanIf2BsCapabilitiesConfigTable) are used for negotiation
37            of basic capabilities with SS using RNG-RSP, SBC-RSP and
38            REG-RSP messages. The negotiated capabilities are obtained
39            by interSubclause of SS raw reported capabilities, BS raw
40            capabilities and BS configured capabilities. The objects in
41            the table have read-only access. The table is maintained
42            by BS."
43
44        ::= { wmanIf2BsCapabilities 3 }
45
46
47
48    wmanIf2BsBasicCapabilitiesEntry OBJECT-TYPE
49        SYNTAX      WmanIf2BsBasicCapabilitiesEntry
50        MAX-ACCESS  not-accessible
51        STATUS      current
52        DESCRIPTION
53            "This table provides one row for each BS sector and is
54            indexed by ifIndex."
55
56        INDEX { ifIndex }
57
58        ::= { wmanIf2BsBasicCapabilitiesTable 1 }
59
60    WmanIf2BsBasicCapabilitiesEntry ::= SEQUENCE {
61        wmanIf2BsCapUplinkCidSupport          WmanIf2NumOfCid,
62        wmanIf2BsCapArqSupport               WmanIf2ArqSupportType,
63        wmanIf2BsCapDsxFlowControl           WmanIf2MaxDsxFlowType,
64        wmanIf2BsCapMacCrcSupport            WmanIf2MacCrcSupport,
65

```



```

1      wmanIf2BsCapMcaFlowControl          WmanIf2MaxMcaFlowType,
2      wmanIf2BsCapMcpGroupCidSupport      WmanIf2MaxMcpGroupCid,
3      wmanIf2BsCapPkmFlowControl          WmanIf2MaxPkmFlowType,
4      wmanIf2BsCapAuthPolicyControl       WmanIf2AuthPolicyType,
5      wmanIf2BsCapMaxNumOfSupportedSA     WmanIf2MaxNumOfSaType,
6      wmanIf2BsCapIpVersion               WmanIf2IpVersionType,
7      wmanIf2BsCapMacCsSupportBitMap      WmanIf2MacCsBitMap,
8      wmanIf2BsCapMaxNumOfClassifier      WmanIf2MaxClassifiers,
9      wmanIf2BsCapPhsSupport              WmanIf2PhsSupportType,
10     wmanIf2BsCapBandwidthAllocSupport    WmanIf2BwAllocSupport,
11     wmanIf2BsCapPduConstruction          WmanIf2PduConstruction,
12     wmanIf2BsCapTtgTransitionGap         WmanIf2SsTransitionGap,
13     wmanIf2BsCapRtgTransitionGap         WmanIf2SsTransitionGap,
14     wmanIf2BsCapDownlinkCidSupport       WmanIf2NumOfCid,
15     wmanIf2BsCapPackingSupport           WmanIf2PackingSupport,
16     wmanIf2BsCapExtendedRtpsSupport      WmanIf2ExtRtpsSupport,
17     wmanIf2BsCapMaxNumBurstToMs         INTEGER,
18     wmanIf2BsCapIpAddrAllocMethod        WmanIf2IpAllocMethod,
19     wmanIf2BsCapArqAckType               WmanIf2ArqAckType,
20     wmanIf2BsCapMacHeader                 WmanIf2MacHeaderSupp,
21     wmanIf2BsSsCapMaxMacLevelDlFrame     WmanIf2MaxMacLevel,
22     wmanIf2BsSsCapMaxMacLevelUlFrame     WmanIf2MaxMacLevel,
23     wmanIf2BsCapNumOfProvisionedSf       Unsigned32}
24
25
26
27
28
29
30     wmanIf2BsCapUplinkCidSupport OBJECT-TYPE
31         SYNTAX          WmanIf2NumOfCid
32         MAX-ACCESS      read-only
33         STATUS          current
34         DESCRIPTION
35             "This object shows the number of Uplink CIDs the BS can
36             support per SS."
37         ::= { wmanIf2BsBasicCapabilitiesEntry 1 }
38
39
40
41     wmanIf2BsCapArqSupport OBJECT-TYPE
42         SYNTAX          WmanIf2ArqSupportType
43         MAX-ACCESS      read-only
44         STATUS          current
45         DESCRIPTION
46             "This object indicates whether the BS supports ARQ."
47         ::= { wmanIf2BsBasicCapabilitiesEntry 2 }
48
49
50
51     wmanIf2BsCapDsxFowControl OBJECT-TYPE
52         SYNTAX          WmanIf2MaxDsxFowType
53         MAX-ACCESS      read-only
54         STATUS          current
55         DESCRIPTION
56             "This object specifies the maximum number of concurrent
57             DSA, DSC, or DSD transactions that BS allows each SS to
58             have outstanding."
59         DEFVAL          { 0 }
60         ::= { wmanIf2BsBasicCapabilitiesEntry 3 }
61
62
63
64     wmanIf2BsCapMacCrcSupport OBJECT-TYPE
65         SYNTAX          WmanIf2MacCrcSupport

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This object indicates whether or not the BS supports MAC
5          level CRC."
6      DEFVAL      { macCrcSupport }
7      ::= { wmanIf2BsBasicCapabilitiesEntry 4 }
8
9
10     wmanIf2BsCapMcaFlowControl OBJECT-TYPE
11     SYNTAX      WmanIf2MaxMcaFlowType
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "This object specifies the maximum number of concurrent
16         MCA transactions that BS allows each SS to have."
17     DEFVAL      { 0 }
18     ::= { wmanIf2BsBasicCapabilitiesEntry 5 }
19
20
21     wmanIf2BsCapMcpGroupCidSupport OBJECT-TYPE
22     SYNTAX      WmanIf2MaxMcpGroupCid
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "This object indicates the maximum number of simultaneous
27         Multicast Polling Groups the BS allows each SS to belong
28         to."
29     DEFVAL      { 0 }
30     ::= { wmanIf2BsBasicCapabilitiesEntry 6 }
31
32
33     wmanIf2BsCapPkmFlowControl OBJECT-TYPE
34     SYNTAX      WmanIf2MaxPkmFlowType
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38         "This object specifies the maximum number of concurrent
39         PKM transactions that BS allows each SS to have."
40     DEFVAL      { 0 }
41     ::= { wmanIf2BsBasicCapabilitiesEntry 7 }
42
43
44     wmanIf2BsCapAuthPolicyControl OBJECT-TYPE
45     SYNTAX      WmanIf2AuthPolicyType
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "This object specifies authorization policy that BS is
50         capable of. A bit value of 0 = 'not supported', 1 =
51         'supported'. If this field is omitted, then both SS and
52         BS shall use the IEEE 802.16 security, constituting X.509
53         digital certificates and the RSA public key encryption
54         algorithm, as authorization policy."
55     ::= { wmanIf2BsBasicCapabilitiesEntry 8 }
56
57
58     wmanIf2BsCapMaxNumOfSupportedSA OBJECT-TYPE
59     SYNTAX      WmanIf2MaxNumOfSaType
60
61
62
63
64
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This field specifies maximum number of supported security
5          associations per SS that the BS allows."
6      DEFVAL      { 1 }
7      ::= { wmanIf2BsBasicCapabilitiesEntry 9 }
8
9
10     wmanIf2BsCapIpVersion OBJECT-TYPE
11         SYNTAX      WmanIf2IpVersionType
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "This object indicates the version of IP BS allows each SS
16             to use on the 2nd Management Connection. The value
17             'undefined' should not be used for this field."
18         REFERENCE
19             "Subclause 11.7.4 in IEEE Std 802.16-2004"
20         ::= { wmanIf2BsBasicCapabilitiesEntry 10 }
21
22
23     wmanIf2BsCapMacCsSupportBitMap OBJECT-TYPE
24         SYNTAX      WmanIf2MacCsBitMap
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "This object indicates BS set of MAC convergence
29             sublayer support. When a bit is set, it indicates
30             the corresponding CS feature is supported."
31         ::= { wmanIf2BsBasicCapabilitiesEntry 11 }
32
33
34     wmanIf2BsCapMaxNumOfClassifier OBJECT-TYPE
35         SYNTAX      WmanIf2MaxClassifiers
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "This object indicates the maximum number of admitted
40             Classifiers per SS that the BS allows."
41         DEFVAL      { 0 }
42         ::= { wmanIf2BsBasicCapabilitiesEntry 12 }
43
44
45     wmanIf2BsCapPhsSupport OBJECT-TYPE
46         SYNTAX      WmanIf2PhsSupportType
47         MAX-ACCESS  read-only
48         STATUS      current
49         DESCRIPTION
50             "This object indicates the level of BS support for PHS.
51             The usage is defined by WmanIf2PhsSupportType."
52         DEFVAL      { noPhsSupport }
53         ::= { wmanIf2BsBasicCapabilitiesEntry 13 }
54
55
56     wmanIf2BsCapBandwidthAllocSupport OBJECT-TYPE
57         SYNTAX      WmanIf2BwAllocSupport
58         MAX-ACCESS  read-only
59         STATUS      current
60
61
62
63
64
65

```

```

1      DESCRIPTION
2          "This field indicates the bandwidth allocation properties
3          that the BS permits SSs to use. The usage is defined by
4          WmanIf2BwAllocSupport."
5          ::= { wmanIf2BsBasicCapabilitiesEntry 14 }
6
7
8      wmanIf2BsCapPduConstruction OBJECT-TYPE
9          SYNTAX      WmanIf2PduConstruction
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "Specifies the capabilities for construction and
14             transmission of MAC PDUs allowed by the BS. The usage is
15             defined by WmanIf2PduConstruction."
16             ::= { wmanIf2BsBasicCapabilitiesEntry 15 }
17
18
19
20     wmanIf2BsCapTtgTransitionGap OBJECT-TYPE
21         SYNTAX      WmanIf2SsTransitionGap
22         UNITS       "us"
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "This field indicates the transition speed SSTTG for TDD
27             and H-FDD SSs allowed by the BS. The usage is defined by
28             WmanIf2SsTransitionGap."
29             ::= { wmanIf2BsBasicCapabilitiesEntry 16 }
30
31
32
33     wmanIf2BsCapRtgTransitionGap OBJECT-TYPE
34         SYNTAX      WmanIf2SsTransitionGap
35         UNITS       "us"
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "This field indicates the transition speed SSRTG for TDD
40             and H-FDD SSs allowed by the BS. The usage is defined
41             by WmanIf2SsTransitionGap."
42             ::= { wmanIf2BsBasicCapabilitiesEntry 17 }
43
44
45
46     wmanIf2BsCapDownlinkCidSupport OBJECT-TYPE
47         SYNTAX      WmanIf2NumOfCid
48         MAX-ACCESS  read-only
49         STATUS      current
50         DESCRIPTION
51             "This object shows the number of Downlink CIDs the SS can
52             support."
53             ::= { wmanIf2BsBasicCapabilitiesEntry 18 }
54
55
56
57     wmanIf2BsCapPackingSupport OBJECT-TYPE
58         SYNTAX      WmanIf2PackingSupport
59         MAX-ACCESS  read-only
60         STATUS      current
61         DESCRIPTION
62             "Indicates the availability of MS support for Packing."
63         REFERENCE
64
65

```

```

1      "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
2      ::= { wmanIf2BsBasicCapabilitiesEntry 19 }
3
4
5  wmanIf2BsCapExtendedRtptsSupport OBJECT-TYPE
6      SYNTAX      WmanIf2ExtRtptsSupport
7      MAX-ACCESS  read-only
8      STATUS      current
9      DESCRIPTION
10         "Indicates the availability of SS support for extended
11         rtPs."
12
13     REFERENCE
14         "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
15     ::= { wmanIf2BsBasicCapabilitiesEntry 20 }
16
17
18  wmanIf2BsCapMaxNumBurstToMs OBJECT-TYPE
19      SYNTAX      INTEGER (1..16)
20      MAX-ACCESS  read-only
21      STATUS      current
22      DESCRIPTION
23         "Maximum number of bursts transmitted concurrently to the MS
24         , including all bursts without CID or with CIDs matching
25         the MS CIDs."
26
27     REFERENCE
28         "Subclause 11.7.8.13 in IEEE Std 802.16e-2005"
29     ::= { wmanIf2BsBasicCapabilitiesEntry 21 }
30
31
32  wmanIf2BsCapIpAddrAllocMethod OBJECT-TYPE
33      SYNTAX      WmanIf2IpAllocMethod
34      MAX-ACCESS  read-only
35      STATUS      current
36      DESCRIPTION
37         "Indicates the method of allocating IP address for the
38         secondary management connection."
39
40     REFERENCE
41         "Subclause 11.7.11 in IEEE Std 802.16e-2005"
42     ::= { wmanIf2BsBasicCapabilitiesEntry 22 }
43
44
45  wmanIf2BsCapArqAckType OBJECT-TYPE
46      SYNTAX      WmanIf2ArqAckType
47      MAX-ACCESS  read-only
48      STATUS      current
49      DESCRIPTION
50         "The value of this parameter specifies the ARQ ACK type
51         supported by the MS."
52
53     REFERENCE
54         "Subclause 11.7.23 in IEEE Std 802.16e-2005"
55     ::= { wmanIf2BsBasicCapabilitiesEntry 23 }
56
57
58  wmanIf2BsCapMacHeader OBJECT-TYPE
59      SYNTAX      WmanIf2MacHeaderSupp
60      MAX-ACCESS  read-only
61      STATUS      current
62      DESCRIPTION
63         "Indicates whether or not the MS and BS support various
64
65

```

```

1         types of MAC header and extended subheaders."
2     REFERENCE
3         "Subclause 11.7.25 in IEEE Std 802.16e-2005"
4     ::= { wmanIf2BsBasicCapabilitiesEntry 24 }
5
6
7     wmanIf2BsSsCapMaxMacLevelDlFrame OBJECT-TYPE
8         SYNTAX      WmanIf2MaxMacLevel
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            "Maximum amount of MAC level data the MS is capable of
13             processing per DL frame. A value of 0 indicates such
14             limitation does not exist, except the limitation of
15             the physical medium"
16        REFERENCE
17            "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
18        DEFVAL      { 0 }
19        ::= { wmanIf2BsBasicCapabilitiesEntry 25 }
20
21
22
23
24     wmanIf2BsSsCapMaxMacLevelUlFrame OBJECT-TYPE
25         SYNTAX      WmanIf2MaxMacLevel
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "Maximum amount of MAC level data the MS is capable of
30             processing per UL frame. A value of 0 indicates such
31             limitation does not exist, except the limitation of
32             the physical medium"
33         REFERENCE
34             "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
35         DEFVAL      { 0 }
36         ::= { wmanIf2BsBasicCapabilitiesEntry 26 }
37
38
39
40
41     wmanIf2BsCapNumOfProvisionedSf OBJECT-TYPE
42         SYNTAX      Unsigned32 (0 .. 255)
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "When a BS is to transmit multiple DSA transactions for
47             provisioned service flows, this object indicates how many
48             DSA transactions with provisioned service flows will be
49             transmitted."
50         REFERENCE
51             "Subclause 11.7.19 in IEEE Std 802.16e-2005"
52         ::= { wmanIf2BsBasicCapabilitiesEntry 27 }
53
54
55
56     wmanIf2BsCapabilitiesConfigTable OBJECT-TYPE
57         SYNTAX      SEQUENCE OF WmanIf2BsCapabilitiesConfigEntry
58         MAX-ACCESS  not-accessible
59         STATUS      current
60         DESCRIPTION
61             "This table contains the configuration for basic
62             capabilities of BS. The table is intended to be used to
63             restrict the Capabilities implemented by BS, for example in
64
65

```

```

1      order to comply with local regulatory requirements. The BS
2      should use the configuration along with the implemented
3      Capabilities (wmanIf2BsBasicCapabilitiesTable) for
4      negotiation of basic capabilities with SS using RNG-RSP,
5      SBC-RSP and REG-RSP messages. The negotiated capabilities
6      are obtained by interSubclause of SS reported capabilities,
7      BS raw capabilities and BS configured capabilities. The
8      objects in the table have read-write access. The rows are
9      created by BS as a copy of wmanIf2BsBasicCapabilitiesTable
10     and can be modified by NMS."
11
12     ::= { wmanIf2BsCapabilities 4 }
13
14
15     wmanIf2BsCapabilitiesConfigEntry OBJECT-TYPE
16         SYNTAX      WmanIf2BsCapabilitiesConfigEntry
17         MAX-ACCESS  not-accessible
18         STATUS      current
19         DESCRIPTION
20             "This table provides one row for each BS sector and is
21             indexed by ifIndex."
22         INDEX { ifIndex }
23         ::= { wmanIf2BsCapabilitiesConfigTable 1 }
24
25
26
27     WmanIf2BsCapabilitiesConfigEntry ::= SEQUENCE {
28         wmanIf2BsCapCfgUplinkCidSupport      WmanIf2NumOfCid,
29         wmanIf2BsCapCfgArqSupport           WmanIf2ArqSupportType,
30         wmanIf2BsCapCfgDsxFowControl        WmanIf2MaxDsxFowType,
31         wmanIf2BsCapCfgMacCrcSupport        WmanIf2MacCrcSupport,
32         wmanIf2BsCapCfgMcaFlowControl       WmanIf2MaxMcaFlowType,
33         wmanIf2BsCapCfgMcpGroupCidSupport   WmanIf2MaxMcpGroupCid,
34         wmanIf2BsCapCfgPkmFlowControl       WmanIf2MaxPkmFlowType,
35         wmanIf2BsCapCfgAuthPolicyControl    WmanIf2AuthPolicyType,
36         wmanIf2BsCapCfgMaxNumOfSupportedSA WmanIf2MaxNumOfSaType,
37         wmanIf2BsCapCfgIpVersion           WmanIf2IpVersionType,
38         wmanIf2BsCapCfgMacCsSupportBitMap  WmanIf2MacCsBitMap,
39         wmanIf2BsCapCfgMaxNumOfClassifier  WmanIf2MaxClassifiers,
40         wmanIf2BsCapCfgPhsSupport          WmanIf2PhsSupportType,
41         wmanIf2BsCapCfgBandwidthAllocSupport WmanIf2BwAllocSupport,
42         wmanIf2BsCapCfgPduConstruction     WmanIf2PduConstruction,
43         wmanIf2BsCapCfgTtgTransitionGap    WmanIf2SsTransitionGap,
44         wmanIf2BsCapCfgRtgTransitionGap    WmanIf2SsTransitionGap,
45         wmanIf2BsCapCfgDownlinkCidSupport  WmanIf2NumOfCid,
46         wmanIf2BsCapCfgPackingSupport      WmanIf2PackingSupport,
47         wmanIf2BsCapCfgExtendedRtpsSupport WmanIf2ExtRtpsSupport,
48         wmanIf2BsCapCfgMaxNumBurstToMs    INTEGER,
49         wmanIf2BsCapCfgIpAddrAllocMethod   WmanIf2IpAllocMethod,
50         wmanIf2BsCapCfgArqAckType         WmanIf2ArqAckType,
51         wmanIf2BsCapCfgMacHeader          WmanIf2MacHeaderSupp,
52         wmanIf2BsSsCapCfgMaxMacLevelDlFrame WmanIf2MaxMacLevel,
53         wmanIf2BsSsCapCfgMaxMacLevelUlFrame WmanIf2MaxMacLevel,
54         wmanIf2BsCapCfgNumOfProvisionedSf Unsigned32 }
55
56
57     wmanIf2BsCapCfgUplinkCidSupport OBJECT-TYPE
58         SYNTAX      WmanIf2NumOfCid
59         MAX-ACCESS  read-write
60
61
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object shows the configured number of Uplink CIDs the
4          BS can support per SS."
5          ::= { wmanIf2BsCapabilitiesConfigEntry 1 }
6
7
8      wmanIf2BsCapCfgArqSupport OBJECT-TYPE
9          SYNTAX      WmanIf2ArqSupportType
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "This object indicates whether the BS is configured to
14             support ARQ."
15             ::= { wmanIf2BsCapabilitiesConfigEntry 2 }
16
17
18         wmanIf2BsCapCfgDsxFowControl OBJECT-TYPE
19             SYNTAX      WmanIf2MaxDsxFowType
20             MAX-ACCESS  read-write
21             STATUS      current
22             DESCRIPTION
23                 "This object specifies the configured maximum number of
24                 concurrent DSA, DSC, or DSD transactions that BS allows
25                 each SS to have outstanding."
26                 DEFVAL  { 0 }
27                 ::= { wmanIf2BsCapabilitiesConfigEntry 3 }
28
29
30         wmanIf2BsCapCfgMacCrcSupport OBJECT-TYPE
31             SYNTAX      WmanIf2MacCrcSupport
32             MAX-ACCESS  read-write
33             STATUS      current
34             DESCRIPTION
35                 "This object indicates whether BS is configured to support
36                 MAC level CRC."
37                 DEFVAL  { macCrcSupport }
38                 ::= { wmanIf2BsCapabilitiesConfigEntry 4 }
39
40
41         wmanIf2BsCapCfgMcaFlowControl OBJECT-TYPE
42             SYNTAX      WmanIf2MaxMcaFlowType
43             MAX-ACCESS  read-write
44             STATUS      current
45             DESCRIPTION
46                 "This object specifies the maximum number of concurrent
47                 MCA transactions that BS is configured to allow each SS to
48                 have."
49                 DEFVAL  { 0 }
50                 ::= { wmanIf2BsCapabilitiesConfigEntry 5 }
51
52
53         wmanIf2BsCapCfgMcpGroupCidSupport OBJECT-TYPE
54             SYNTAX      WmanIf2MaxMcpGroupCid
55             MAX-ACCESS  read-write
56             STATUS      current
57             DESCRIPTION
58                 "This object indicates the maximum number of simultaneous
59                 Multicast Polling Groups the BS is configured to allow
60
61
62
63
64
65

```



```

1         each SS to belong to."
2     DEFVAL      { 0 }
3     ::= { wmanIf2BsCapabilitiesConfigEntry 6 }
4
5
6 wmanIf2BsCapCfgPkmFlowControl OBJECT-TYPE
7     SYNTAX      WmanIf2MaxPkmFlowType
8     MAX-ACCESS  read-write
9     STATUS      current
10    DESCRIPTION
11        "This object specifies the maximum number of concurrent
12         PKM transactions that BS is configured to allow each SS
13         to have."
14    DEFVAL      { 0 }
15    ::= { wmanIf2BsCapabilitiesConfigEntry 7 }
16
17
18
19 wmanIf2BsCapCfgAuthPolicyControl OBJECT-TYPE
20     SYNTAX      WmanIf2AuthPolicyType
21     MAX-ACCESS  read-write
22     STATUS      current
23     DESCRIPTION
24        "This object specifies authorization policy that BS is
25         configured to be capable of. A bit value of 0 = 'not
26         supported', 1 = 'supported'. If this field is omitted,
27         then both SS and BS shall use the IEEE 802.16 security,
28         constituting X.509 digital certificates and the RSA
29         public key encryption algorithm, as authorization policy."
30    ::= { wmanIf2BsCapabilitiesConfigEntry 8 }
31
32
33
34
35 wmanIf2BsCapCfgMaxNumOfSupportedSA OBJECT-TYPE
36     SYNTAX      WmanIf2MaxNumOfSaType
37     MAX-ACCESS  read-write
38     STATUS      current
39     DESCRIPTION
40        "This field specifies configured maximum number of supported
41         security association per SS."
42    DEFVAL      { 1 }
43    ::= { wmanIf2BsCapabilitiesConfigEntry 9 }
44
45
46
47 wmanIf2BsCapCfgIpVersion OBJECT-TYPE
48     SYNTAX      WmanIf2IpVersionType
49     MAX-ACCESS  read-write
50     STATUS      current
51     DESCRIPTION
52        "This object indicates the configured version of IP that the
53         BS allows each SS to use on the 2nd Management Connection.
54         The value 'undefined' should not be used in this field."
55    ::= { wmanIf2BsCapabilitiesConfigEntry 10 }
56
57
58
59 wmanIf2BsCapCfgMacCsSupportBitMap OBJECT-TYPE
60     SYNTAX      WmanIf2MacCsBitMap
61     MAX-ACCESS  read-write
62     STATUS      current
63     DESCRIPTION
64        "This object indicates BS configured set of MAC convergence
65

```

```

1         sublayer support. When a bit is set, it indicates
2         the corresponding CS feature is supported."
3         ::= { wmanIf2BsCapabilitiesConfigEntry 11 }
4
5
6 wmanIf2BsCapCfgMaxNumOfClassifier OBJECT-TYPE
7     SYNTAX      WmanIf2MaxClassifiers
8     MAX-ACCESS  read-write
9     STATUS      current
10    DESCRIPTION
11        "This object indicates the configured maximum number of
12        admitted Classifiers per SS that the BS can support."
13    DEFVAL      { 0 }
14    ::= { wmanIf2BsCapabilitiesConfigEntry 12 }
15
16
17 wmanIf2BsCapCfgPhsSupport OBJECT-TYPE
18     SYNTAX      WmanIf2PhsSupportType
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22        "This object indicates the configured level of BS support
23        for PHS."
24    DEFVAL      { noPhsSupport }
25    ::= { wmanIf2BsCapabilitiesConfigEntry 13 }
26
27
28 wmanIf2BsCapCfgBandwidthAllocSupport OBJECT-TYPE
29     SYNTAX      WmanIf2BwAllocSupport
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33        "This field indicates configured properties of the BS for
34        bandwidth allocation purposes. The usage is defined by
35        WmanIf2CapBwAllocSupport."
36    ::= { wmanIf2BsCapabilitiesConfigEntry 14 }
37
38
39 wmanIf2BsCapCfgPduConstruction OBJECT-TYPE
40     SYNTAX      WmanIf2PduConstruction
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44        "Specifies configured capabilities for construction and
45        transmission of MAC PDUs. The usage is defined by
46        WmanIf2PduConstruction."
47    ::= { wmanIf2BsCapabilitiesConfigEntry 15 }
48
49
50 wmanIf2BsCapCfgTtgTransitionGap OBJECT-TYPE
51     SYNTAX      WmanIf2SsTransitionGap
52     UNITS       "us"
53     MAX-ACCESS  read-write
54     STATUS      current
55     DESCRIPTION
56        "This field indicates the configured transition speed
57        SSTTG for TDD and H-FDD SSs. The usage is defined by
58        WmanIf2SsTransitionGap."
59    ::= { wmanIf2BsCapabilitiesConfigEntry 16 }
60
61
62
63
64
65

```

```

1
2
3 wmanIf2BsCapCfgRtgTransitionGap OBJECT-TYPE
4     SYNTAX      WmanIf2SsTransitionGap
5     UNITS       "us"
6     MAX-ACCESS  read-write
7     STATUS      current
8     DESCRIPTION
9         "This field indicates the configured transition speed
10        SSRTG for TDD and H-FDD SSs. The usage is defined by
11        WmanIf2SsTransitionGap."
12        ::= { wmanIf2BsCapabilitiesConfigEntry 17 }
13
14
15 wmanIf2BsCapCfgDownlinkCidSupport OBJECT-TYPE
16     SYNTAX      WmanIf2NumOfCid
17     MAX-ACCESS  read-write
18     STATUS      current
19     DESCRIPTION
20         "This object shows the number of Downlink CIDs the SS can
21        support."
22        ::= { wmanIf2BsCapabilitiesConfigEntry 18 }
23
24
25
26 wmanIf2BsCapCfgPackingSupport OBJECT-TYPE
27     SYNTAX      WmanIf2PackingSupport
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31         "Indicates the availability of MS support for Packing."
32     REFERENCE
33         "Subclause 11.7.8.11 in IEEE Std 802.16e-2005"
34     ::= { wmanIf2BsCapabilitiesConfigEntry 19 }
35
36
37
38 wmanIf2BsCapCfgExtendedRtPsSupport OBJECT-TYPE
39     SYNTAX      WmanIf2ExtRtPsSupport
40     MAX-ACCESS  read-write
41     STATUS      current
42     DESCRIPTION
43         "Indicates the availability of SS support for extended
44        rtPs."
45     REFERENCE
46         "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
47     ::= { wmanIf2BsCapabilitiesConfigEntry 20 }
48
49
50
51 wmanIf2BsCapCfgMaxNumBurstToMs OBJECT-TYPE
52     SYNTAX      INTEGER (1..16)
53     MAX-ACCESS  read-write
54     STATUS      current
55     DESCRIPTION
56         "Maximum number of bursts transmitted concurrently to the MS
57        , including all bursts without CID or with CIDs matching
58        the MS CIDs."
59     REFERENCE
60         "Subclause 11.7.8.13 in IEEE Std 802.16e-2005"
61     ::= { wmanIf2BsCapabilitiesConfigEntry 21 }
62
63
64
65

```

```

1  wmanIf2BsCapCfgIpAddrAllocMethod OBJECT-TYPE
2      SYNTAX      WmanIf2IpAllocMethod
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "Indicates the method of allocating IP address for the
7            secondary management connection."
8      REFERENCE
9          "Subclause 11.7.11 in IEEE Std 802.16e-2005"
10     ::= { wmanIf2BsCapabilitiesConfigEntry 22 }
11
12
13
14  wmanIf2BsCapCfgArqAckType OBJECT-TYPE
15      SYNTAX      WmanIf2ArqAckType
16      MAX-ACCESS  read-write
17      STATUS      current
18      DESCRIPTION
19          "The value of this parameter specifies the ARQ ACK type
20            supported by the MS."
21      REFERENCE
22          "Subclause 11.7.23 in IEEE Std 802.16e-2005"
23     ::= { wmanIf2BsCapabilitiesConfigEntry 23 }
24
25
26
27  wmanIf2BsCapCfgMacHeader OBJECT-TYPE
28      SYNTAX      WmanIf2MacHeaderSupp
29      MAX-ACCESS  read-write
30      STATUS      current
31      DESCRIPTION
32          "Indicates whether or not the MS and BS support various
33            types of MAC header and extended subheaders."
34      REFERENCE
35          "Subclause 11.7.25 in IEEE Std 802.16e-2005"
36     ::= { wmanIf2BsCapabilitiesConfigEntry 24 }
37
38
39
40  wmanIf2BsSsCapCfgMaxMacLevelDlFrame OBJECT-TYPE
41      SYNTAX      WmanIf2MaxMacLevel
42      MAX-ACCESS  read-write
43      STATUS      current
44      DESCRIPTION
45          "Maximum amount of MAC level data the MS is capable of
46            processing per DL frame. A value of 0 indicates such
47            limitation does not exist, except the limitation of
48            the physical medium"
49      REFERENCE
50          "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
51     DEFVAL      { 0 }
52     ::= { wmanIf2BsCapabilitiesConfigEntry 25 }
53
54
55
56
57  wmanIf2BsSsCapCfgMaxMacLevelUlFrame OBJECT-TYPE
58      SYNTAX      WmanIf2MaxMacLevel
59      MAX-ACCESS  read-write
60      STATUS      current
61      DESCRIPTION
62          "Maximum amount of MAC level data the MS is capable of
63            processing per UL frame. A value of 0 indicates such
64
65

```

```

1      limitation does not exist, except the limitation of
2      the physical medium"
3
4      REFERENCE
5      "Subclause 11.7.8.10.1 in IEEE Std 802.16e-2005"
6      DEFVAL      { 0 }
7      ::= { wmanIf2BsCapabilitiesConfigEntry 26 }
8
9
10     wmanIf2BsCapCfgNumOfProvisionedSf OBJECT-TYPE
11     SYNTAX      Unsigned32 (0 .. 255)
12     MAX-ACCESS  read-write
13     STATUS      current
14     DESCRIPTION
15         "When a BS is to transmit multiple DSA transactions for
16         provisioned service flows, this object indicates how many
17         DSA transactions with provisioned service flows will be
18         transmitted."
19
20     REFERENCE
21     "Subclause 11.7.19 in IEEE Std 802.16e-2005"
22     ::= { wmanIf2BsCapabilitiesConfigEntry 27 }
23
24
25     wmanIf2BsSsActionsTable OBJECT-TYPE
26     SYNTAX      SEQUENCE OF WmanIf2BsSsActionsEntry
27     MAX-ACCESS  not-accessible
28     STATUS      current
29     DESCRIPTION
30         "This table contains all the actions specified for SSs in
31         the standard. The actions are routed down to SS using
32         unsolicited MAC messages: REG-RSP, DREG-REQ and RES-CMD.
33         The table also contains the parameters of the actions in
34         cases where they are specified by the standard."
35     ::= { wmanIf2BsCps 5 }
36
37
38
39     wmanIf2BsSsActionsEntry OBJECT-TYPE
40     SYNTAX      WmanIf2BsSsActionsEntry
41     MAX-ACCESS  not-accessible
42     STATUS      current
43     DESCRIPTION
44         "This table is indexed by wmanIf2BsSsActionsMacAddress. The
45         action can be requested for SS in any state not only those
46         registered. However BS will decide whether the action is
47         applicable to the SS based on its current state and execute
48         it or skip it as defined in each action definition."
49     INDEX { wmanIf2BsSsActionsMacAddress }
50     ::= { wmanIf2BsSsActionsTable 1 }
51
52
53
54
55     WmanIf2BsSsActionsEntry ::= SEQUENCE {
56         wmanIf2BsSsActionsMacAddress      MacAddress,
57         wmanIf2BsSsActionsResetSs        INTEGER,
58         wmanIf2BsSsActionsAbortSs        INTEGER,
59         wmanIf2BsSsActionsOverrideDnFreq Unsigned32,
60         wmanIf2BsSsActionsOverrideChannelId INTEGER,
61         wmanIf2BsSsActionsDeReRegSs      INTEGER,
62         wmanIf2BsSsActionsDeReRegSsCode  INTEGER,
63         wmanIf2BsSsActionsRowStatus      RowStatus }
64
65

```

```

1
2
3 wmanIf2BsSsActionsMacAddress OBJECT-TYPE
4     SYNTAX      MacAddress
5     MAX-ACCESS  not-accessible
6     STATUS      current
7     DESCRIPTION
8         "This object uniquely identifies the SS as an action
9         target."
10
11     ::= { wmanIf2BsSsActionsEntry 1 }
12
13 wmanIf2BsSsActionsResetSs OBJECT-TYPE
14     SYNTAX      INTEGER {actionsResetSsNoAction(0),
15                    actionsResetSs(1)}
16     MAX-ACCESS  read-create
17     STATUS      current
18     DESCRIPTION
19
20         "This object should be implemented as follows:
21         - When set to actionsResetSs value, instructs BS to send
22         RES-CMD to SS
23         - When set to value different than actionsResetSs it
24         should be ignored
25         - When read it should return actionsResetSsNoAction
26         The RES-CMD message shall be transmitted by the BS on an
27         SS Basic CID to force the SS to reset itself,
28         reinitialize its MAC, and repeat initial system access."
29
30     REFERENCE
31
32         "Subclause 6.3.2.3.22 in IEEE Std 802.16-2004"
33
34     ::= { wmanIf2BsSsActionsEntry 2 }
35
36 wmanIf2BsSsActionsAbortSs OBJECT-TYPE
37     SYNTAX      INTEGER {actionsAbortSsNoAction(0),
38                    actionsAbortSs(1),
39                    actionAbortSsParams(2)}
40
41     MAX-ACCESS  read-create
42     STATUS      current
43     DESCRIPTION
44
45         "This object should be implemented as follows:
46         - When set to actionsAbortSs value, it instructs BS to send
47         unsolicited RNG-RSP with Ranging Status equal to 'abort'
48         without override parameters
49         - When set to actionAbortSsParams value, it instructs BS to
50         send unsolicited RNG-RSP with Ranging Status equal to
51         'abort' and with 'Downlink Frequency Override' and
52         'Uplink Channel ID Override' parameters.
53         - When set to any other value it should be ignored
54         - When read it should returned actionsAbortSsNoAction"
55
56     REFERENCE
57
58         "Subclause 11.6, Table 365 in IEEE Std 802.16-2004"
59
60     ::= { wmanIf2BsSsActionsEntry 3 }
61
62 wmanIf2BsSsActionsOverrideDnFreq OBJECT-TYPE
63     SYNTAX      Unsigned32
64     UNITS       "kHz"
65

```

```

1      MAX-ACCESS  read-create
2      STATUS      current
3      DESCRIPTION
4          "This object is used as a parameter of the AbortSs action
5          with the code actionAbortSsParams. It is used for licensed
6          bands only. It defines the Center frequency, in kHz, of
7          new downlink channel where the SS should redo initial
8          ranging."
9
10     REFERENCE
11         "Subclause 11.6, Table 365 in IEEE Std 802.16-2004"
12     ::= { wmanIf2BsSsActionsEntry 4 }
13
14
15     wmanIf2BsSsActionsOverrideChannelId OBJECT-TYPE
16     SYNTAX      INTEGER (0..199)
17     MAX-ACCESS  read-create
18     STATUS      current
19     DESCRIPTION
20         "This object is used as a parameter of the AbortSs action
21         with the code actionAbortSsParams. It is coded as follows:
22         - Licensed bands: The identifier of the uplink channel
23         with which the SS is to redo initial ranging (not used
24         with PHYs without channelized uplinks).
25         - License-exempt bands: The Channel Nr (see 8.5.1) where
26         the SS should redo initial ranging."
27
28     REFERENCE
29         "Subclause 11.6, Table 365 in IEEE Std 802.16-2004"
30     ::= { wmanIf2BsSsActionsEntry 5 }
31
32
33
34     wmanIf2BsSsActionsDeReRegSs OBJECT-TYPE
35     SYNTAX      INTEGER {actionsDeReRegSsNoAction(0),
36                  actionsDeReRegSs(1)}
37     MAX-ACCESS  read-create
38     STATUS      current
39     DESCRIPTION
40         "This object should be implemented as follows:
41         - When set to actionsDeReRegSs value, instructs BS to
42         send DREG-CMD to SS with specified action code
43         - When set to value different than actionsDeReRegSs it
44         should be ignored
45         - When read it should return actionsDeReRegSsNoAction
46         The DREG-CMD message shall be transmitted by the BS on an
47         SS Basic CID to force the SS to change its access state.
48         Upon receiving a DREG-CMD, the SS shall take the action
49         indicated by the action code defined by
50         wmanIf2BsSsActionsDeReRegSsCode."
51
52     REFERENCE
53         "Subclause 6.3.2.3.26 in IEEE Std 802.16-2004"
54     ::= { wmanIf2BsSsActionsEntry 6 }
55
56
57
58     wmanIf2BsSsActionsDeReRegSsCode OBJECT-TYPE
59     SYNTAX      INTEGER {actionsDeReRegSsCodeChangeChan(0),
60                  actionsDeReRegSsCodeNoTransmit(1),
61                  actionsDeReRegSsCodeLtdTransmit(2),
62                  actionsDeReRegSsCodeResume(3)}
63
64
65

```

```

1      MAX-ACCESS  read-create
2      STATUS      current
3      DESCRIPTION
4          "This object defines the action code for
5          wmanIf2BsSsActionsDeReRegSs action. The codes are defined
6          as follows:
7          actionsDeReRegSsCodeChangeChan - SS shall leave the
8          current channel and attempt to access another channel.
9          actionsDeReRegSsCodeNoTransmit - SS shall listen to the
10         current channel but shall not transmit until an
11         RES-CMD message or DREG_CMD with an Action Code that
12         allows transmission is received.
13         actionsDeReRegSsCodeLtdTransmit - SS shall listen to the
14         current channel but only transmit on the Basic,
15         Primary Management and 2nd Management Connections.
16         actionsDeReRegSsCodeResume - SS shall return to normal
17         operation and may transmit on any of its active
18         connections."
19
20     REFERENCE
21         "Subclause 6.3.2.3.26, Table 55 in IEEE Std 802.16-2004"
22     ::= { wmanIf2BsSsActionsEntry 7 }
23
24 wmanIf2BsSsActionsRowStatus OBJECT-TYPE
25     SYNTAX      RowStatus
26     MAX-ACCESS  read-create
27     STATUS      current
28     DESCRIPTION
29         "This object is used to ensure that the write operation to
30         multiple columns is guaranteed to be treated as atomic
31         operation by agent."
32     ::= { wmanIf2BsSsActionsEntry 8 }
33
34 --
35 -- Base station PKM group
36 -- wmanIf2BsPkmObjects contain the Base Station Privacy Sublayer objects
37 --
38 wmanIf2BsPkmObjects OBJECT IDENTIFIER ::= { wmanIf2BsObjects 3 }
39
40 --
41 -- Table wmanIf2BsPkmBaseTable
42 --
43 wmanIf2BsPkmBaseTable OBJECT-TYPE
44     SYNTAX      SEQUENCE OF WmanIf2BsPkmBaseEntry
45     MAX-ACCESS  not-accessible
46     STATUS      current
47     DESCRIPTION
48         "This table describes the basic PKM attributes of each Base
49         Station wireless interface."
50     ::= { wmanIf2BsPkmObjects 1 }
51
52 wmanIf2BsPkmBaseEntry OBJECT-TYPE
53     SYNTAX      WmanIf2BsPkmBaseEntry
54     MAX-ACCESS  not-accessible
55     STATUS      current
56
57
58
59
60
61
62
63
64
65

```



```

1      DESCRIPTION
2          "Each entry contains objects describing attributes of one
3          BS wireless interface."
4      INDEX      { ifIndex }
5      ::= { wmanIf2BsPkmBaseTable 1 }
6
7
8      WmanIf2BsPkmBaseEntry ::= SEQUENCE {
9          wmanIf2BsPkmDefaultAuthLifetime      Integer32,
10         wmanIf2BsPkmDefaultTekLifetime       Integer32,
11         wmanIf2BsPkmDefaultSelfSigManufCertTrust INTEGER,
12         wmanIf2BsPkmCheckCertValidityPeriods TruthValue,
13         wmanIf2BsPkmAuthentInfos             Counter32,
14         wmanIf2BsPkmAuthRequests             Counter32,
15         wmanIf2BsPkmAuthReplies              Counter32,
16         wmanIf2BsPkmAuthRejects              Counter32,
17         wmanIf2BsPkmAuthInvalids             Counter32,
18         wmanIf2BsPkmAuthGraceTime            Integer32,
19         wmanIf2BsPkmTekGraceTime             Integer32,
20         wmanIf2BsPkmAuthWaitTimeout          Integer32,
21         wmanIf2BsPkmReauthWaitTimeout        Integer32,
22         wmanIf2BsPkmOpWaitTimeout            Integer32,
23         wmanIf2BsPkmRekeyWaitTimeout         Integer32,
24         wmanIf2BsPkmAuthRejectWaitTimeout    Integer32}
25
26
27
28
29
30     wmanIf2BsPkmDefaultAuthLifetime OBJECT-TYPE
31         SYNTAX      Integer32 (86400..6048000)
32         UNITS       "seconds"
33         MAX-ACCESS  read-write
34         STATUS      current
35         DESCRIPTION
36             "The value of this object is the default lifetime, in
37             seconds, the BS assigns to a new authorization key."
38         REFERENCE
39             "Table 343 in IEEE Std 802.16-2004"
40         DEFVAL      { 604800 }
41         ::= { wmanIf2BsPkmBaseEntry 1 }
42
43
44
45     wmanIf2BsPkmDefaultTekLifetime OBJECT-TYPE
46         SYNTAX      Integer32 (1800..604800)
47         UNITS       "seconds"
48         MAX-ACCESS  read-write
49         STATUS      current
50         DESCRIPTION
51             "The value of this object is the default lifetime, in
52             seconds, the BS assigns to a new Traffic Encryption
53             Key (TEK)."
54         REFERENCE
55             "Table 343 in IEEE Std 802.16-2004"
56         DEFVAL      { 43200 }
57         ::= { wmanIf2BsPkmBaseEntry 2 }
58
59
60
61
62
63     wmanIf2BsPkmDefaultSelfSigManufCertTrust OBJECT-TYPE
64         SYNTAX      INTEGER {trusted (1),
65
```

```

1         untrusted (2) }
2     MAX-ACCESS    read-write
3     STATUS        current
4     DESCRIPTION
5         "This object determines the default trust of all (new)
6         self-signed manufacturer certificates obtained after
7         setting the object."
8     ::= { wmanIf2BsPkmBaseEntry 3 }
9
10
11
12 wmanIf2BsPkmCheckCertValidityPeriods OBJECT-TYPE
13     SYNTAX        TruthValue
14     MAX-ACCESS    read-write
15     STATUS        current
16     DESCRIPTION
17         "Setting this object to TRUE causes all certificates
18         received thereafter to have their validity periods (and
19         their chain's validity periods) checked against the current
20         time of day. A FALSE setting will cause all certificates
21         received Thereafter to not have their validity periods
22         (nor their chain's validity periods) checked against the
23         current time of day."
24     ::= { wmanIf2BsPkmBaseEntry 4 }
25
26
27
28
29 wmanIf2BsPkmAuthentInfos OBJECT-TYPE
30     SYNTAX        Counter32
31     MAX-ACCESS    read-only
32     STATUS        current
33     DESCRIPTION
34         "The value of this object is the count of times the BS has
35         received an Authentication Information message from any
36         SS."
37     ::= { wmanIf2BsPkmBaseEntry 5 }
38
39
40
41 wmanIf2BsPkmAuthRequests OBJECT-TYPE
42     SYNTAX        Counter32
43     MAX-ACCESS    read-only
44     STATUS        current
45     DESCRIPTION
46         "The value of this object is the count of times the BS has
47         received an Authorization Request message from any SS"
48     ::= { wmanIf2BsPkmBaseEntry 6 }
49
50
51
52 wmanIf2BsPkmAuthReplies OBJECT-TYPE
53     SYNTAX        Counter32
54     MAX-ACCESS    read-only
55     STATUS        current
56     DESCRIPTION
57         "The value of this object is the count of times the BS has
58         transmitted an Authorization Reply message to any SS."
59     ::= { wmanIf2BsPkmBaseEntry 7 }
60
61
62
63 wmanIf2BsPkmAuthRejects OBJECT-TYPE
64     SYNTAX        Counter32
65     MAX-ACCESS    read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object is the count of times the BS has
4          transmitted an Authorization Reject message to any SS."
5          ::= { wmanIf2BsPkmBaseEntry 8 }
6
7
8      wmanIf2BsPkmAuthInvalids OBJECT-TYPE
9          SYNTAX      Counter32
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "The value of this object is the count of times the BS has
14             transmitted an Authorization Invalid message to any SS."
15             ::= { wmanIf2BsPkmBaseEntry 9 }
16
17
18
19     wmanIf2BsPkmAuthGraceTime OBJECT-TYPE
20         SYNTAX      Integer32 (300..3024000)
21         UNITS       "seconds"
22         MAX-ACCESS  read-write
23         STATUS      current
24         DESCRIPTION
25             "The value of this object is the grace time for an
26             authorization key. A SS is expected to start trying to get
27             a new authorization key beginning AuthGraceTime seconds
28             before the authorization key actually expires."
29             REFERENCE
30                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
31             DEFVAL   { 600 }
32             ::= { wmanIf2BsPkmBaseEntry 10 }
33
34
35
36
37     wmanIf2BsPkmTekGraceTime OBJECT-TYPE
38         SYNTAX      Integer32 (300..3024000)
39         UNITS       "seconds"
40         MAX-ACCESS  read-write
41         STATUS      current
42         DESCRIPTION
43             "The value of this object is the grace time for the TEK in
44             seconds. The SS is expected to start trying to acquire a
45             new TEK beginning TEK GraceTime seconds before the
46             expiration of the most recent TEK."
47             REFERENCE
48                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
49             DEFVAL   { 3600 }
50             ::= { wmanIf2BsPkmBaseEntry 11 }
51
52
53
54
55     wmanIf2BsPkmAuthWaitTimeout OBJECT-TYPE
56         SYNTAX      Integer32 (2..30)
57         UNITS       "seconds"
58         MAX-ACCESS  read-write
59         STATUS      current
60         DESCRIPTION
61             "The value of this object is the Authorize Wait Timeout."
62             REFERENCE
63                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
64
65

```

```

1      DEFVAL      { 10 }
2      ::= { wmanIf2BsPkmBaseEntry 12 }
3
4
5      wmanIf2BsPkmReauthWaitTimeout OBJECT-TYPE
6          SYNTAX      Integer32 (2..30)
7          UNITS        "seconds"
8          MAX-ACCESS  read-write
9          STATUS      current
10         DESCRIPTION
11             "The value of this object is the Reauthorize Wait Timeout
12             in seconds."
13         REFERENCE
14             "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
15         DEFVAL      { 10 }
16         ::= { wmanIf2BsPkmBaseEntry 13 }
17
18
19
20         wmanIf2BsPkmOpWaitTimeout OBJECT-TYPE
21             SYNTAX      Integer32 (1..10)
22             UNITS        "seconds"
23             MAX-ACCESS  read-write
24             STATUS      current
25             DESCRIPTION
26                 "The value of this object is the Operational Wait Timeout
27                 in seconds."
28             REFERENCE
29                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
30             DEFVAL      { 1 }
31             ::= { wmanIf2BsPkmBaseEntry 14 }
32
33
34
35
36         wmanIf2BsPkmRekeyWaitTimeout OBJECT-TYPE
37             SYNTAX      Integer32 (1..10)
38             UNITS        "seconds"
39             MAX-ACCESS  read-write
40             STATUS      current
41             DESCRIPTION
42                 "The value of this object is the Rekey Wait Timeout in
43                 seconds."
44             REFERENCE
45                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
46             DEFVAL      { 1 }
47             ::= { wmanIf2BsPkmBaseEntry 15 }
48
49
50
51
52         wmanIf2BsPkmAuthRejectWaitTimeout OBJECT-TYPE
53             SYNTAX      Integer32 (10..600)
54             UNITS        "seconds"
55             MAX-ACCESS  read-write
56             STATUS      current
57             DESCRIPTION
58                 "The value of this object is the Authorization Reject Wait
59                 Timeout in seconds."
60             REFERENCE
61                 "Table 343 and subclause 11.9.19 in IEEE Std 802.16-2004"
62             DEFVAL      { 60 }
63             ::= { wmanIf2BsPkmBaseEntry 16 }
64
65

```

```

1
2
3 --
4 -- Table wmanIf2BsSsPkmAuthTable
5 --
6 wmanIf2BsSsPkmAuthTable OBJECT-TYPE
7     SYNTAX      SEQUENCE OF WmanIf2BsSsPkmAuthEntry
8     MAX-ACCESS  not-accessible
9     STATUS      current
10    DESCRIPTION
11        "This table describes PKM attributes related
12         to the authorization for each SS. The BS maintains one
13         Primary Security Association with each Baseline
14         Privacy-enabled SS on each BS wireless interface."
15    ::= { wmanIf2BsPkmObjects 2 }
16
17
18
19 wmanIf2BsSsPkmAuthEntry OBJECT-TYPE
20     SYNTAX      WmanIf2BsSsPkmAuthEntry
21     MAX-ACCESS  not-accessible
22     STATUS      current
23     DESCRIPTION
24        "The BS MUST create one entry per SS per wireless
25         interface, based on the receipt of an Authorization
26         Request message and MUST not delete the entry before
27         the SS authorization permanently expires."
28     INDEX       { ifIndex, wmanIf2BsSsPkmAuthMacAddress }
29    ::= { wmanIf2BsSsPkmAuthTable 1 }
30
31
32
33
34 WmanIf2BsSsPkmAuthEntry ::= SEQUENCE {
35     wmanIf2BsSsPkmAuthMacAddress      MacAddress,
36     wmanIf2BsSsPkmAuthKeySequenceNumber Integer32,
37     wmanIf2BsSsPkmAuthExpiresOld      DateAndTime,
38     wmanIf2BsSsPkmAuthExpiresNew      DateAndTime,
39     wmanIf2BsSsPkmAuthLifetime         Integer32,
40     wmanIf2BsSsPkmAuthReset            INTEGER,
41     wmanIf2BsSsPkmAuthInfos            Counter64,
42     wmanIf2BsSsPkmAuthRequests         Counter64,
43     wmanIf2BsSsPkmAuthReplies          Counter64,
44     wmanIf2BsSsPkmAuthRejects          Counter64,
45     wmanIf2BsSsPkmAuthInvalids         Counter64,
46     wmanIf2BsSsPkmAuthRejectErrorCode INTEGER,
47     wmanIf2BsSsPkmAuthRejectErrorString SnmpAdminString,
48     wmanIf2BsSsPkmAuthInvalidErrorCode INTEGER,
49     wmanIf2BsSsPkmAuthInvalidErrorString SnmpAdminString,
50     wmanIf2BsSsPkmAuthPrimarySAId      INTEGER,
51     wmanIf2BsSsPkmAuthValidStatus      INTEGER}
52
53
54
55
56 wmanIf2BsSsPkmAuthMacAddress OBJECT-TYPE
57     SYNTAX      MacAddress
58     MAX-ACCESS  not-accessible
59     STATUS      current
60     DESCRIPTION
61        "The value of this object is the physical address of the SS
62         to which the authorization association applies."
63    ::= { wmanIf2BsSsPkmAuthEntry 1 }
64
65

```

```

1
2
3 wmanIf2BsSsPkmAuthKeySequenceNumber OBJECT-TYPE
4     SYNTAX      Integer32 (0..15)
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "The value of this object is the most recent authorization
9         key sequence number for this SS."
10
11     ::= { wmanIf2BsSsPkmAuthEntry 2 }
12
13 wmanIf2BsSsPkmAuthExpiresOld OBJECT-TYPE
14     SYNTAX      DateAndTime
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is the actual clock time for
19         expiration of the immediate predecessor of the most recent
20         authorization key for this FSM. If this FSM has only one
21         authorization key, then the value is the time of activation
22         of this FSM."
23
24     ::= { wmanIf2BsSsPkmAuthEntry 3 }
25
26
27 wmanIf2BsSsPkmAuthExpiresNew OBJECT-TYPE
28     SYNTAX      DateAndTime
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32         "The value of this object is the actual clock time for
33         expiration of the most recent authorization key for this
34         FSM"
35
36     ::= { wmanIf2BsSsPkmAuthEntry 4 }
37
38
39 wmanIf2BsSsPkmAuthLifetime OBJECT-TYPE
40     SYNTAX      Integer32 (86400..6048000)
41     UNITS       "seconds"
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45         "The vaue of this object is the lifetime, in seconds, the
46         BS assigns to an authorization key for this SS."
47
48     REFERENCE
49         "Table 343 in IEEE Std 802.16-2004"
50
51     DEFVAL     { 604800 }
52
53     ::= { wmanIf2BsSsPkmAuthEntry 5 }
54
55 wmanIf2BsSsPkmAuthReset OBJECT-TYPE
56     SYNTAX      INTEGER {noResetRequested(1),
57                       invalidateAuth(2),
58                       sendAuthInvalid(3),
59                       invalidateTeks(4)}
60
61     MAX-ACCESS  read-write
62     STATUS      current
63     DESCRIPTION
64         "Setting this object to invalidateAuth(2) causes the BS to
65

```

```

1      invalidate the current SS authorization key(s), but not to
2      transmit an Authorization Invalid message nor to invalidate
3      unicast TEKs. Setting this object to sendAuthInvalid(3)
4      causes the BS to invalidate the current SS authorization
5      key(s), and to transmit an Authorization Invalid message to
6      the SS, but not to invalidate unicast TEKs. Setting this
7      object to invalidateTekS(4) causes the BS to invalidate the
8      current SS authorization key(s), to transmit an
9      Authorization Invalid message to the SS, and to
10     invalidate all unicast TEKs associated with this SS
11     authorization. Reading this object returns the
12     most-recently-set value of this object, or returns
13     noResetRequested(1) if the object has not been set since
14     the last BS reboot."
15     ::= { wmanIf2BsSsPkmAuthEntry 6 }
16
17
18
19
20     wmanIf2BsSsPkmAuthInfos OBJECT-TYPE
21         SYNTAX      Counter64
22         MAX-ACCESS  read-only
23         STATUS      current
24         DESCRIPTION
25             "The value of this object is the count of times the BS has
26             received an Authentication Information message from this
27             SS."
28         ::= { wmanIf2BsSsPkmAuthEntry 7 }
29
30
31
32     wmanIf2BsSsPkmAuthRequests OBJECT-TYPE
33         SYNTAX      Counter64
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "The value of this object is the count of times the BS has
38             received an Authorization Request message from this SS."
39         ::= { wmanIf2BsSsPkmAuthEntry 8 }
40
41
42
43     wmanIf2BsSsPkmAuthReplies OBJECT-TYPE
44         SYNTAX      Counter64
45         MAX-ACCESS  read-only
46         STATUS      current
47         DESCRIPTION
48             "The value of this object is the count of times the BS has
49             transmitted an Authorization Reply message to this SS."
50         ::= { wmanIf2BsSsPkmAuthEntry 9 }
51
52
53
54     wmanIf2BsSsPkmAuthRejects OBJECT-TYPE
55         SYNTAX      Counter64
56         MAX-ACCESS  read-only
57         STATUS      current
58         DESCRIPTION
59             "The value of this object is the count of times the BS has
60             transmitted an Authorization Reject message to this SS."
61         ::= { wmanIf2BsSsPkmAuthEntry 10 }
62
63
64
65     wmanIf2BsSsPkmAuthInvalids OBJECT-TYPE

```

```

1      SYNTAX      Counter64
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "The value of this object is the count of times the BS has
6          transmitted an Authorization Invalid message to this SS."
7      ::= { wmanIf2BsSsPkmAuthEntry 11 }
8
9
10
11 wmanIf2BsSsPkmAuthRejectErrorCode OBJECT-TYPE
12     SYNTAX      INTEGER {noInformation(0),
13                 unauthorizedSs(1),
14                 unauthorizedSaid(2),
15                 permanentAuthorizationFailure(6)}
16     MAX-ACCESS  read-only
17     STATUS      current
18     DESCRIPTION
19         "The value of this object is the enumerated description of
20         the Error-Code in most recent Authorization Reject message
21         transmitted to the SS."
22     REFERENCE
23         "IEEE Std 802.16-2004; Table 371"
24     ::= { wmanIf2BsSsPkmAuthEntry 12 }
25
26
27
28
29 wmanIf2BsSsPkmAuthRejectErrorString OBJECT-TYPE
30     SYNTAX      SnmpAdminString (SIZE (0..128))
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "The value of this object is the Display-String in most
35         recent Authorization Reject message transmitted to the SS.
36         This is a zero length string if no Authorization Reject
37         message has been transmitted to the SS."
38     ::= { wmanIf2BsSsPkmAuthEntry 13 }
39
40
41
42 wmanIf2BsSsPkmAuthInvalidErrorCode OBJECT-TYPE
43     SYNTAX      INTEGER {noInformation(0),
44                 unauthorizedSs(1),
45                 unsolicited(3),
46                 invalidKeySequence(4),
47                 keyRequestAuthenticationFailure(5)}
48     MAX-ACCESS  read-only
49     STATUS      current
50     DESCRIPTION
51         "The value of this object is the enumerated description of
52         the Error-Code in most recent Authorization Invalid message
53         transmitted to the SS."
54     REFERENCE
55         "IEEE Std 802.16-2004; Table 371"
56     ::= { wmanIf2BsSsPkmAuthEntry 14 }
57
58
59
60
61 wmanIf2BsSsPkmAuthInvalidErrorString OBJECT-TYPE
62     SYNTAX      SnmpAdminString (SIZE (0..128))
63     MAX-ACCESS  read-only
64     STATUS      current
65

```



```

1      DESCRIPTION
2          "The value of this object is the Display-String in most
3          recent Authorization Invalid message transmitted to the SS.
4          This is a zero length string if no Authorization Invalid
5          message has been transmitted to the SS."
6      ::= { wmanIf2BsSsPkmAuthEntry 15 }
7
8
9
10     wmanIf2BsSsPkmAuthPrimarySAId OBJECT-TYPE
11         SYNTAX      INTEGER (0..65535)
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "The value of this object is the Primary Security
16             Association identifier."
17         REFERENCE
18             "IEEE Std 802.16-2004; 11.9.7"
19         ::= { wmanIf2BsSsPkmAuthEntry 16 }
20
21
22
23     wmanIf2BsSsPkmAuthValidStatus OBJECT-TYPE
24         SYNTAX      INTEGER {unknown (0),
25                     validSsChained (1),
26                     validSsTrusted (2),
27                     invalidSsUntrusted (3),
28                     invalidCAUntrusted (4),
29                     invalidSsOther (5),
30                     invalidCAOther (6)}
31         MAX-ACCESS  read-only
32         STATUS      current
33         DESCRIPTION
34             "Contains the reason why a SS's certificate is deemed valid
35             or invalid. Return unknown if the SS is running PKM mode.
36             ValidSsChained means the certificate is valid because it
37             chains to a valid certificate. ValidSsTrusted means the
38             certificate is valid because it has been provisioned to be
39             trusted. InvalidSsUntrusted means the certificate is
40             invalid because it has been provisioned to be untrusted.
41             InvalidCAUntrusted means the certificate is invalid
42             because it chains to an untrusted certificate.
43             InvalidSsOther and InvalidCAOther refer to errors in
44             parsing, validity periods, etc, which are attributable to
45             the SS certificate or its chain respectively."
46         ::= { wmanIf2BsSsPkmAuthEntry 17 }
47
48
49
50
51
52
53     --
54     -- Table wmanIf2BsPkmTekTable
55     --
56     wmanIf2BsPkmTekTable OBJECT-TYPE
57         SYNTAX      SEQUENCE OF WmanIf2BsPkmTekEntry
58         MAX-ACCESS  not-accessible
59         STATUS      current
60         DESCRIPTION
61             "This table describes the attributes of each Traffic
62             Encryption Key (TEK) association. The BS maintains one TEK
63             association per SAID on each BS wireless interface."
64
65

```

```

1      ::= { wmanIf2BsPkmObjects 3 }
2
3
4  wmanIf2BsPkmTekEntry OBJECT-TYPE
5      SYNTAX      WmanIf2BsPkmTekEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "Each entry contains objects describing attributes of one
10         TEK association on a particular BS wireless interface. The
11         BS MUST create one entry per SAID per wireless interface,
12         based on the receipt of a Key Request message, and MUST not
13         delete the entry before the SS authorization for the SAID
14         permanently expires."
15
16     INDEX      { ifIndex, wmanIf2BsPkmTekSAID }
17     ::= { wmanIf2BsPkmTekTable 1 }
18
19
20  WmanIf2BsPkmTekEntry ::= SEQUENCE {
21      wmanIf2BsPkmTekSAID          INTEGER,
22      wmanIf2BsPkmTekSAType        INTEGER,
23      wmanIf2BsPkmTekDataEncryptAlg WmanIf2DataEncryptAlgId,
24      wmanIf2BsPkmTekDataAuthAlg   WmanIf2DataAuthAlgId,
25      wmanIf2BsPkmTekEncryptAlg    WmanIf2TekEncryptAlgId,
26      wmanIf2BsPkmTekLifetime      Integer32,
27      wmanIf2BsPkmTekKeySequenceNumber Integer32,
28      wmanIf2BsPkmTekExpiresOld    DateAndTime,
29      wmanIf2BsPkmTekExpiresNew    DateAndTime,
30      wmanIf2BsPkmTekReset         TruthValue,
31      wmanIf2BsPkmKeyRequests      Counter32,
32      wmanIf2BsPkmKeyReplies       Counter32,
33      wmanIf2BsPkmKeyRejects       Counter32,
34      wmanIf2BsPkmTekInvalids      Counter32,
35      wmanIf2BsPkmKeyRejectErrorCode INTEGER,
36      wmanIf2BsPkmKeyRejectErrorString SnmpAdminString,
37      wmanIf2BsPkmTekInvalidErrorCode INTEGER,
38      wmanIf2BsPkmTekInvalidErrorString SnmpAdminString}
39
40
41  wmanIf2BsPkmTekSAID OBJECT-TYPE
42      SYNTAX      INTEGER (0..65535)
43      MAX-ACCESS  not-accessible
44      STATUS      current
45      DESCRIPTION
46          "The value of this object is the Security Association
47          ID (SAID)."

```

```

1         "The value of this object is the type of security
2         association. Dynamic does not apply to Ss running in PKM
3         mode."
4     REFERENCE
5         "IEEE Std 802.16-2004; subclause 11.9.18"
6     ::= { wmanIf2BsPkmTekEntry 2 }
7
8
9
10    wmanIf2BsPkmTekDataEncryptAlg OBJECT-TYPE
11        SYNTAX      WmanIf2DataEncryptAlgId
12        MAX-ACCESS  read-only
13        STATUS      current
14        DESCRIPTION
15            "The value of this object is the data encryption algorithm
16            being utilized."
17        REFERENCE
18            "Table 375, IEEE Std 802.16-2004"
19        ::= { wmanIf2BsPkmTekEntry 3 }
20
21
22
23    wmanIf2BsPkmTekDataAuthentAlg OBJECT-TYPE
24        SYNTAX      WmanIf2DataAuthAlgId
25        MAX-ACCESS  read-only
26        STATUS      current
27        DESCRIPTION
28            "The value of this object is the data authentication
29            algorithm being utilized."
30        REFERENCE
31            "Table 376, IEEE Std 802.16-2004"
32        ::= { wmanIf2BsPkmTekEntry 4 }
33
34
35
36    wmanIf2BsPkmTekEncryptAlg OBJECT-TYPE
37        SYNTAX      WmanIf2TekEncryptAlgId
38        MAX-ACCESS  read-only
39        STATUS      current
40        DESCRIPTION
41            "The value of this object is the TEK key encryption
42            algorithm being utilized."
43        REFERENCE
44            "Table 377, IEEE Std 802.16-2004"
45        ::= { wmanIf2BsPkmTekEntry 5 }
46
47
48
49    wmanIf2BsPkmTekLifetime OBJECT-TYPE
50        SYNTAX      Integer32 (1800..604800)
51        UNITS       "seconds"
52        MAX-ACCESS  read-only
53        STATUS      current
54        DESCRIPTION
55            "The value of this object is the lifetime, in seconds, the
56            BS assigns to keys for this TEK association."
57        REFERENCE
58            "Table 343 in IEEE Std 802.16-2004"
59        DEFVAL     { 43200 }
60        ::= { wmanIf2BsPkmTekEntry 6 }
61
62
63
64    wmanIf2BsPkmTekKeySequenceNumber OBJECT-TYPE
65

```

```

1      SYNTAX      Integer32 (0..3)
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "The value of this object is the most recent TEK key
6          sequence number for this SAID."
7      REFERENCE
8          "IEEE Std 802.16-2004; subclause 11.9.5"
9      ::= { wmanIf2BsPkmTekEntry 7 }
10
11
12
13 wmanIf2BsPkmTekExpiresOld OBJECT-TYPE
14     SYNTAX      DateAndTime
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is the actual clock time for
19         expiration of the immediate predecessor of the most recent
20         TEK for this FSM. If this FSM has only one TEK, then the
21         value is the time of activation of this FSM."
22     ::= { wmanIf2BsPkmTekEntry 8 }
23
24
25
26 wmanIf2BsPkmTekExpiresNew OBJECT-TYPE
27     SYNTAX      DateAndTime
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31         "The value of this object is the actual clock time for
32         expiration of the most recent TEK for this FSM."
33     ::= { wmanIf2BsPkmTekEntry 9 }
34
35
36
37 wmanIf2BsPkmTekReset OBJECT-TYPE
38     SYNTAX      TruthValue
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "Setting this object to TRUE causes the BS to invalidate
43         the current active TEK(s) (plural due to key transition
44         periods), and to generate a new TEK for the associated
45         SAID; the BS MAY also generate an unsolicited TEK Invalid
46         message, to optimize the TEK synchronization between the BS
47         and the SS. Reading this object always returns FALSE."
48     ::= { wmanIf2BsPkmTekEntry 10 }
49
50
51
52
53 wmanIf2BsPkmKeyRequests OBJECT-TYPE
54     SYNTAX      Counter32
55     MAX-ACCESS  read-only
56     STATUS      current
57     DESCRIPTION
58         "The value of this object is the count of times the BS has
59         received a Key Request message."
60     ::= { wmanIf2BsPkmTekEntry 11 }
61
62
63
64 wmanIf2BsPkmKeyReplies OBJECT-TYPE
65     SYNTAX      Counter32

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the count of times the BS has
5          transmitted a Key Reply message."
6      ::= { wmanIf2BsPkmTekEntry 12 }
7
8
9
10     wmanIf2BsPkmKeyRejects OBJECT-TYPE
11         SYNTAX      Counter32
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "The value of this object is the count of times the BS has
16             transmitted a Key Reject message."
17         ::= { wmanIf2BsPkmTekEntry 13 }
18
19
20     wmanIf2BsPkmTekInvalids OBJECT-TYPE
21         SYNTAX      Counter32
22         MAX-ACCESS  read-only
23         STATUS      current
24         DESCRIPTION
25             "The value of this object is the count of times the BS has
26             transmitted a TEK Invalid message."
27         ::= { wmanIf2BsPkmTekEntry 14 }
28
29
30
31     wmanIf2BsPkmKeyRejectErrorCode OBJECT-TYPE
32         SYNTAX      INTEGER {noInformation(0),
33                     unauthorizedSaid(2)}
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "The value of this object is the enumerated; description of
38             the Error-Code in the most recent Key Reject message sent
39             in response to a Key Request for this SAID."
40         REFERENCE
41             "IEEE Std 802.16-2004; Table 371"
42         ::= { wmanIf2BsPkmTekEntry 15 }
43
44
45
46     wmanIf2BsPkmKeyRejectErrorString OBJECT-TYPE
47         SYNTAX      SnmpAdminString (SIZE (0..128))
48         MAX-ACCESS  read-only
49         STATUS      current
50         DESCRIPTION
51             "The value of this object is the Display-String in the most
52             recent Key Reject message sent in response to a Key Request
53             for this SAID. This is a zero length string if no Key
54             Reject message has been received since reboot."
55         ::= { wmanIf2BsPkmTekEntry 16 }
56
57
58
59     wmanIf2BsPkmTekInvalidErrorCode OBJECT-TYPE
60         SYNTAX      INTEGER {noInformation(0),
61                     invalidKeySequence(4)}
62         MAX-ACCESS  read-only
63         STATUS      current
64
65

```

```

1      DESCRIPTION
2          "The value of this object is the enumerated description of
3          the Error-Code in the most recent TEK Invalid message sent
4          in association with this SAID."
5
6      REFERENCE
7          "IEEE Std 802.16-2004; Table 371"
8      ::= { wmanIf2BsPkmTekEntry 17 }
9
10
11 wmanIf2BsPkmTekInvalidErrorString OBJECT-TYPE
12     SYNTAX      SnmpAdminString (SIZE (0..128))
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "The value of this object is the Display-String in the most
17         recent TEK Invalid message sent in association with this
18         SAID. This is a zero length string if no TEK Invalid
19         message has been received since reboot."
20     ::= { wmanIf2BsPkmTekEntry 18 }
21
22
23
24 --
25 -- Base station Notification Group
26 -- wmanIf2BsNotificationObjects contains the BS SNMP Trap objects
27 --
28
29 wmanIf2BsNotification OBJECT IDENTIFIER ::= { wmanIf2BsObjects 4 }
30 wmanIf2BsTrapControl    OBJECT IDENTIFIER ::= { wmanIf2BsNotification 1
31 }
32 wmanIf2BsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIf2BsNotification 2
33 }
34
35
36 -- This object groups all NOTIFICATION-TYPE objects for BS.
37 -- It is defined following RFC2758 sections 8.5 and 8.6
38 -- for the compatibility with SNMPv1.
39 wmanIf2BsTrapPrefix OBJECT IDENTIFIER ::= { wmanIf2BsTrapDefinitions 0 }
40
41
42 wmanIf2BsTrapControlRegister    OBJECT-TYPE
43     SYNTAX      BITS {wmanIf2BsSsStatusNotification      (0),
44                     wmanIf2BsSsDynamicServiceFail      (1),
45                     wmanIf2BsSsRssiStatusChange        (2),
46                     wmanIf2BsSsRegistrer               (3),
47                     wmanIf2BsSsPkmFail                  (4),
48                     wmanIf2BsSsDynamicServiceFail2     (5),
49                     wmanIf2BsSsRegister2Trap           (6)}
50
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "The object is used to enable or disable Base Station traps.
55         From left to right, the set bit indicates the corresponding
56         Base Station trap is enabled."
57     ::= { wmanIf2BsTrapControl 1 }
58
59
60
61 wmanIf2BsStatusTrapControlRegister    OBJECT-TYPE
62     SYNTAX      BITS {unused(0),
63                     ssInitRangingSucc(1),
64                     ssInitRangingFail(2),
65

```

```

1         ssRegistered(3),
2         ssRegistrationFail(4),
3         ssDeregistered(5),
4         ssBasicCapabilitySucc(6),
5         ssBasicCapabilityFail(7),
6         ssAuthorizationSucc(8),
7         ssAuthorizationFail(9),
8         tftpSucc(10),
9         tftpFail(11),
10        sfCreationSucc(12),
11        sfCreationFail(13)}
12
13
14        MAX-ACCESS read-write
15        STATUS current
16        DESCRIPTION
17            "The object is used to enable or disable Base Station status
18            notification traps. The set bit indicates the corresponding
19            Base Station trap is enabled."
20        ::= { wmanIf2BsTrapControl 2 }
21
22
23
24        --
25        -- BS threshold Definitions
26        --
27        wmanIf2BsThresholdConfigTable OBJECT-TYPE
28            SYNTAX SEQUENCE OF WmanIf2BsThresholdConfigEntry
29            MAX-ACCESS not-accessible
30            STATUS current
31            DESCRIPTION
32                "This table contains threshold objects that can be set
33                to detect the threshold crossing events."
34            ::= { wmanIf2BsTrapControl 3 }
35
36
37
38        wmanIf2BsThresholdConfigEntry OBJECT-TYPE
39            SYNTAX WmanIf2BsThresholdConfigEntry
40            MAX-ACCESS not-accessible
41            STATUS current
42            DESCRIPTION
43                "This table provides one row for each BS sector, and is
44                indexed by ifIndex."
45            INDEX { ifIndex }
46            ::= { wmanIf2BsThresholdConfigTable 1 }
47
48
49
50        WmanIf2BsThresholdConfigEntry ::= SEQUENCE {
51            wmanIf2BsRssiLowThreshold Integer32,
52            wmanIf2BsRssiHighThreshold Integer32}
53
54
55        wmanIf2BsRssiLowThreshold OBJECT-TYPE
56            SYNTAX Integer32
57            UNITS "dBm"
58            MAX-ACCESS read-write
59            STATUS current
60            DESCRIPTION
61                "Low threshold for generating the RSSI alarm."
62            ::= { wmanIf2BsThresholdConfigEntry 1 }
63
64
65

```

```

1  wmanIf2BsRssiHighThreshold OBJECT-TYPE
2      SYNTAX      Integer32
3      UNITS       "dBm"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "High threshold for clearing the RSSI alarm."
8      ::= { wmanIf2BsThresholdConfigEntry 2 }
9
10
11  --
12  -- Subscriber station Notification Objects Definitions
13  --
14  --
15  wmanIf2BsSsNotificationObjectsTable OBJECT-TYPE
16      SYNTAX      SEQUENCE OF WmanIf2BsSsNotificationObjectsEntry
17      MAX-ACCESS  not-accessible
18      STATUS      current
19      DESCRIPTION
20          "This table contains SS notification objects that have been
21          reported by the trap."
22      ::= { wmanIf2BsTrapDefinitions 1 }
23
24
25
26  wmanIf2BsSsNotificationObjectsEntry OBJECT-TYPE
27      SYNTAX      WmanIf2BsSsNotificationObjectsEntry
28      MAX-ACCESS  not-accessible
29      STATUS      current
30      DESCRIPTION
31          "This table provides one row for each SS that has
32          generated traps, and is double indexed by
33          wmanIf2BsSsNotificationMacAddr and ifIndex for BS sector."
34      INDEX       { ifIndex, wmanIf2BsSsNotificationMacAddr }
35      ::= { wmanIf2BsSsNotificationObjectsTable 1 }
36
37
38
39  WmanIf2BsSsNotificationObjectsEntry ::= SEQUENCE {
40      wmanIf2BsSsNotificationMacAddr      MacAddress,
41      wmanIf2BsSsStatusValue              INTEGER,
42      wmanIf2BsSsStatusInfo               OCTET STRING,
43      wmanIf2BsDynamicServiceType         INTEGER,
44      wmanIf2BsDynamicServiceFailReason   OCTET STRING,
45      wmanIf2BsSsRssiStatus                INTEGER,
46      wmanIf2BsSsRssiStatusInfo           OCTET STRING,
47      wmanIf2BsSsRegisterStatus           INTEGER,
48      wmanIf2BsDynamicServiceFailSfid     Unsigned32 }
49
50
51
52  wmanIf2BsSsNotificationMacAddr OBJECT-TYPE
53      SYNTAX      MacAddress
54      MAX-ACCESS  read-only
55      STATUS      current
56      DESCRIPTION
57          "The MAC address of the SS, reporing the notofiation."
58      ::= { wmanIf2BsSsNotificationObjectsEntry 1 }
59
60
61
62  wmanIf2BsSsStatusValue OBJECT-TYPE
63      SYNTAX      INTEGER { ssInitRangingSucc(1),
64                  ssInitRangingFail(2),
65

```



```

1         ssRegistered(3),
2         ssRegistrationFail(4),
3         ssDeregistered(5),
4         ssBasicCapabilitySucc(6),
5         ssBasicCapabilityFail(7),
6         ssAuthorizationSucc(8),
7         ssAuthorizationFail(9),
8         tftpSucc(10),
9         tftpFail(11),
10        sfCreationSucc(12),
11        sfCreationFail(13)}
12
13        MAX-ACCESS    read-only
14        STATUS        current
15        DESCRIPTION
16            "This object indicates the status of a SS, as it goes
17            through network entry and initialization procedure."
18        ::= { wmanIf2BsSsNotificationObjectsEntry 2 }
19
20
21
22
23        wmanIf2BsSsStatusInfo OBJECT-TYPE
24            SYNTAX      OCTET STRING (SIZE(0..255))
25            MAX-ACCESS  read-only
26            STATUS      current
27            DESCRIPTION
28                "This object indicates the reason of SS's status change."
29            ::= { wmanIf2BsSsNotificationObjectsEntry 3 }
30
31
32
33        wmanIf2BsDynamicServiceType OBJECT-TYPE
34            SYNTAX      INTEGER {bsSfCreationReq(1),
35                               bsSfCreationRsp(2),
36                               bsSfCreationAck(3)}
37            MAX-ACCESS  read-only
38            STATUS      current
39            DESCRIPTION
40                "This object indicates the dynamic service flow
41                creation command type."
42            ::= { wmanIf2BsSsNotificationObjectsEntry 4 }
43
44
45
46        wmanIf2BsDynamicServiceFailReason OBJECT-TYPE
47            SYNTAX      OCTET STRING (SIZE(0..255))
48            MAX-ACCESS  read-only
49            STATUS      current
50            DESCRIPTION
51                "This object indicates the reason why the service flow
52                creation has failed."
53            ::= { wmanIf2BsSsNotificationObjectsEntry 5 }
54
55
56
57        wmanIf2BsSsRssiStatus OBJECT-TYPE
58            SYNTAX      INTEGER {bsRssiAlarm(1),
59                               bsRssiNoAlarm(2)}
60            MAX-ACCESS  read-only
61            STATUS      current
62            DESCRIPTION
63                "A RSSI alarm is generated when RSSI becomes lower than
64                wmanIf2BsLowRssiThreshold and is cleared when RSSI becomes
65

```

```

1         higher than wmanIf2BsLowRssiThreshold."
2     ::= { wmanIf2BsSsNotificationObjectsEntry 6 }
3
4
5 wmanIf2BsSsRssiStatusInfo OBJECT-TYPE
6     SYNTAX      OCTET STRING (SIZE(0..255))
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10
11         "This object indicates the reason why RSSI alarm is
12         generated."
13     ::= { wmanIf2BsSsNotificationObjectsEntry 7 }
14
15 wmanIf2BsSsRegisterStatus OBJECT-TYPE
16     SYNTAX      INTEGER {ssRegister(1),
17                    ssDeregister(2)}
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21
22         "This object indicates the status of SS registration."
23     ::= { wmanIf2BsSsNotificationObjectsEntry 8 }
24
25
26 wmanIf2BsDynamicServiceFailSfid OBJECT-TYPE
27     SYNTAX      Unsigned32 (1..4294967295)
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31
32         "This object identifies the dynamic service flow
33         for notification purposes."
34     ::= { wmanIf2BsSsNotificationObjectsEntry 9 }
35
36
37 --
38 -- Subscriber station Notification Trap Definitions
39 --
40
41 wmanIf2BsSsStatusNotificationTrap NOTIFICATION-TYPE
42     OBJECTS      {ifIndex,
43                    wmanIf2BsSsNotificationMacAddr,
44                    wmanIf2BsSsStatusValue,
45                    wmanIf2BsSsStatusInfo}
46     STATUS      current
47     DESCRIPTION
48
49         "This trap reports the status of a SS. Based on this
50         notification the NMS will issue an alarm with certain
51         severity depending on the status and the reason received."
52     ::= { wmanIf2BsTrapPrefix 1 }
53
54
55 wmanIf2BsSsDynamicServiceFailTrap NOTIFICATION-TYPE
56     OBJECTS      {ifIndex,
57                    wmanIf2BsSsNotificationMacAddr,
58                    wmanIf2BsDynamicServiceType,
59                    wmanIf2BsDynamicServiceFailReason}
60     STATUS      deprecated
61     DESCRIPTION
62
63         "Trap deprecated due to limited value without object
64         reporting SFID of victim service flow."
65

```

```

1         An event to report the failure of a dynamic service
2         operation happened during the dynamic services process
3         and detected in the Bs side."
4     ::= { wmanIf2BsTrapPrefix 2 }
5
6
7 wmanIf2BsSsRssiStatusChangeTrap NOTIFICATION-TYPE
8     OBJECTS      {ifIndex,
9                  wmanIf2BsSsNotificationMacAddr,
10                 wmanIf2BsSsRssiStatus,
11                 wmanIf2BsSsRssiStatusInfo}
12
13     STATUS      current
14
15     DESCRIPTION
16         "An event to report that the uplink RSSI is below
17         wmanIf2BsLowRssiThreshold, or above
18         wmanIf2BsHighRssiThreshold after restore."
19     ::= { wmanIf2BsTrapPrefix 3 }
20
21
22 wmanIf2BsSsPkmFailTrap NOTIFICATION-TYPE
23     OBJECTS      {wmanIf2BsSsNotificationMacAddr}
24
25     STATUS      current
26
27     DESCRIPTION
28         "An event to report the failure of a Pkm operation."
29     ::= { wmanIf2BsTrapPrefix 4 }
30
31
32 wmanIf2BsSsRegistrerTrap NOTIFICATION-TYPE
33     OBJECTS      {wmanIf2BsSsNotificationMacAddr,
34                 wmanIf2BsSsRegisterStatus}
35
36     STATUS      deprecated
37
38     DESCRIPTION
39         "Trap deprecated due to limited value without object ifIndex
40         reported.
41         An event to report SS registration status."
42     ::= { wmanIf2BsTrapPrefix 5 }
43
44
45 wmanIf2BsSsDynamicServiceFail2Trap NOTIFICATION-TYPE
46     OBJECTS      {ifIndex,
47                 wmanIf2BsSsNotificationMacAddr,
48                 wmanIf2BsDynamicServiceType,
49                 wmanIf2BsDynamicServiceFailReason,
50                 wmanIf2BsDynamicServiceFailSfid}
51
52     STATUS      current
53
54     DESCRIPTION
55         "An event reporting failure of DSx operation for a service
56         flow identified by wmanIf2BsDynamicServiceFailSfid and
57         detected in the Bs side."
58     ::= { wmanIf2BsTrapPrefix 6 }
59
60
61 wmanIf2BsSsRegister2Trap NOTIFICATION-TYPE
62     OBJECTS      {ifIndex,
63                 wmanIf2BsSsNotificationMacAddr,
64                 wmanIf2BsSsRegisterStatus}
65
66     STATUS      current
67
68     DESCRIPTION
69         "An event to report SS registration status for a given sector

```

```

1         identified by ifIndex."
2         ::= { wmanIf2BsTrapPrefix 7 }
3
4
5         --
6         -- Base station PHY Group
7         --
8         wmanIf2BsPhy OBJECT IDENTIFIER ::= { wmanIf2BsObjects 6 }
9
10        --
11        -- BS OFDM PHY objects
12        --
13        --
14        wmanIf2BsOfdmPhy OBJECT IDENTIFIER ::= { wmanIf2BsPhy 1 }
15
16        wmanIf2BsOfdmUplinkChannelTable OBJECT-TYPE
17            SYNTAX      SEQUENCE OF WmanIf2BsOfdmUplinkChannelEntry
18            MAX-ACCESS  not-accessible
19            STATUS      current
20            DESCRIPTION
21                "This table contains UCD channel attributes, defining the
22                transmission characteristics of uplink channels"
23            REFERENCE
24                "Table 349 and Table 352, in IEEE Std 802.16-2004"
25            ::= { wmanIf2BsOfdmPhy 1 }
26
27
28
29
30        wmanIf2BsOfdmUplinkChannelEntry OBJECT-TYPE
31            SYNTAX      WmanIf2BsOfdmUplinkChannelEntry
32            MAX-ACCESS  not-accessible
33            STATUS      current
34            DESCRIPTION
35                "This table provides one row for each uplink channel of
36                multi-sector BS, and is indexed by BS ifIndex. An entry
37                in this table exists for each ifEntry of BS with an
38                ifType of ieee80216WMAN."
39            INDEX { ifIndex }
40            ::= { wmanIf2BsOfdmUplinkChannelTable 1 }
41
42
43
44        WmanIf2BsOfdmUplinkChannelEntry ::= SEQUENCE {
45            wmanIf2BsOfdmCtBasedResvTimeout      INTEGER,
46            wmanIf2BsOfdmBwReqOppSize            INTEGER,
47            wmanIf2BsOfdmRangReqOppSize          INTEGER,
48            wmanIf2BsOfdmUplinkCenterFreq       Unsigned32,
49            wmanIf2BsOfdmNumSubChReqRegionFull  INTEGER,
50            wmanIf2BsOfdmNumSymbolsReqRegionFull INTEGER,
51            wmanIf2BsOfdmSubChFocusCtCode       INTEGER,
52            wmanIf2BsOfdmUpLinkChannelId        INTEGER}
53
54
55
56        wmanIf2BsOfdmCtBasedResvTimeout OBJECT-TYPE
57            SYNTAX      INTEGER (1..255)
58            MAX-ACCESS  read-write
59            STATUS      current
60            DESCRIPTION
61                "The number of UL-MAPs to receive before contention-based
62                reservation is attempted again for the same connection."
63            REFERENCE
64
65

```

```

1         "Table 349, in IEEE Std 802.16-2004"
2         ::= { wmanIf2BsOfdmUplinkChannelEntry 1 }
3
4
5 wmanIf2BsOfdmBwReqOppSize OBJECT-TYPE
6     SYNTAX      INTEGER (1..65535)
7     UNITS       "PS"
8     MAX-ACCESS  read-write
9     STATUS      current
10    DESCRIPTION
11        "Size (in units of PS) of PHY payload that SS may use to
12         format and transmit a bandwidth request message in a
13         contention request opportunity. The value includes all
14         PHY overhead as well as allowance for the MAC data the
15         message may hold."
16    REFERENCE
17        "Table 349, in IEEE Std 802.16-2004"
18    ::= { wmanIf2BsOfdmUplinkChannelEntry 2 }
19
20
21 wmanIf2BsOfdmRangReqOppSize OBJECT-TYPE
22     SYNTAX      INTEGER (1..65535)
23     UNITS       "PS"
24     MAX-ACCESS  read-write
25     STATUS      current
26     DESCRIPTION
27        "Size (in units of PS) of PHY payload that SS may use to
28         format and transmit a RNG-REQ message in a contention
29         request opportunity. The value includes all PHY overhead
30         as well as allowance for the MAC data the message may
31         hold and the maximum SS/BS roundtrip propagation delay."
32    REFERENCE
33        "Table 349, in IEEE Std 802.16-2004"
34    ::= { wmanIf2BsOfdmUplinkChannelEntry 3 }
35
36
37 wmanIf2BsOfdmUplinkCenterFreq OBJECT-TYPE
38     SYNTAX      Unsigned32
39     UNITS       "kHz"
40     MAX-ACCESS  read-write
41     STATUS      current
42     DESCRIPTION
43        " Uplink center frequency (kHz) "
44    REFERENCE
45        "Table 349, in IEEE Std 802.16-2004"
46    ::= { wmanIf2BsOfdmUplinkChannelEntry 4 }
47
48
49 wmanIf2BsOfdmNumSubChReqRegionFull OBJECT-TYPE
50     SYNTAX      INTEGER { oneSubchannel(0),
51                    twoSubchannels(1),
52                    fourSubchannels(2),
53                    eightSubchannels(3),
54                    sixteenSubchannels(4) }
55     MAX-ACCESS  read-write
56     STATUS      current
57     DESCRIPTION
58        "Number of subchannels used by each transmit

```

```

1           opportunity when REQ Region-Full is allocated in
2           subchannelization region."
3
4     REFERENCE
5         "Table 352, in IEEE Std 802.16-2004"
6     ::= { wmanIf2BsOfdmUplinkChannelEntry 5 }
7
8 wmanIf2BsOfdmNumSymbolsReqRegionFull OBJECT-TYPE
9     SYNTAX      INTEGER (0..31)
10    MAX-ACCESS  read-write
11    STATUS      current
12    DESCRIPTION
13        "Number of OFDM symbols used by each transmit
14        opportunity when REQ Region-Full is allocated in
15        subchannelization region."
16    REFERENCE
17        "Table 352, in IEEE Std 802.16-2004"
18    ::= { wmanIf2BsOfdmUplinkChannelEntry 6 }
19
20 wmanIf2BsOfdmSubChFocusCtCode OBJECT-TYPE
21     SYNTAX      INTEGER (0..8)
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         "Number of contention codes (CSE) that shall only be used to
26         request a subchannelized allocation. Default value 0.
27         Allowed values 0-8."
28     REFERENCE
29         "Table 352, in IEEE Std 802.16-2004"
30     DEFVAL     { 0 }
31     ::= { wmanIf2BsOfdmUplinkChannelEntry 7 }
32
33 wmanIf2BsOfdmUpLinkChannelId OBJECT-TYPE
34     SYNTAX      INTEGER (0..255)
35     MAX-ACCESS  read-write
36     STATUS      current
37     DESCRIPTION
38         "The identifier of the uplink channel to which the relevant
39         RNG-RSP or RNG-REQ message refers."
40     REFERENCE
41         "Subclause 6.3.2.3.4. Table 16, in IEEE Std 802.16-2004"
42     ::= { wmanIf2BsOfdmUplinkChannelEntry 8 }
43
44 wmanIf2BsOfdmDownlinkChannelTable OBJECT-TYPE
45     SYNTAX      SEQUENCE OF WmanIf2BsOfdmDownlinkChannelEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "This table contains DCD channel attributes, defining the
50         transmission characteristics of downlink channels"
51     REFERENCE
52         "Table 358, in IEEE Std 802.16-2004"
53     ::= { wmanIf2BsOfdmPhy 2 }
54
55 wmanIf2BsOfdmDownlinkChannelEntry OBJECT-TYPE

```

```

1      SYNTAX      WmanIf2BsOfdmDownlinkChannelEntry
2
3      MAX-ACCESS  not-accessible
4
5      STATUS      current
6
7      DESCRIPTION
8          "This table provides one row for each downlink channel of
9          multi-sector BS, and is indexed by BS ifIndex. An entry
10         in this table exists for each ifEntry of BS with an
11         ifType of ieee80216WMAN."
12
13     INDEX { ifIndex }
14     ::= { wmanIf2BsOfdmDownlinkChannelTable 1 }
15
16 WmanIf2BsOfdmDownlinkChannelEntry ::= SEQUENCE {
17     wmanIf2BsOfdmBsEIRP          INTEGER,
18     wmanIf2BsOfdmChannelNumber   WmanIf2ChannelNumber,
19     wmanIf2BsOfdmTTG             INTEGER,
20     wmanIf2BsOfdmRTG             INTEGER,
21     wmanIf2BsOfdmInitRngMaxRSS   INTEGER,
22     wmanIf2BsOfdmDownlinkCenterFreq Unsigned32,
23     wmanIf2BsOfdmBsId            WmanIf2BsIdType,
24     wmanIf2BsOfdmMacVersion      WmanIf2MacVersion,
25     wmanIf2BsOfdmFrameDurationCode INTEGER,
26     wmanIf2BsOfdmDownLinkChannelId INTEGER}
27
28
29 wmanIf2BsOfdmBsEIRP OBJECT-TYPE
30     SYNTAX      INTEGER (-32768..32767)
31     UNITS       "dBm"
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "The EIRP is the equivalent isotropic radiated power of
36         the base station, which is computed for a simple
37         single-antenna transmitter."
38     REFERENCE
39         "Table 358, in IEEE Std 802.16-2004"
40     ::= { wmanIf2BsOfdmDownlinkChannelEntry 1 }
41
42
43
44 wmanIf2BsOfdmChannelNumber OBJECT-TYPE
45     SYNTAX      WmanIf2ChannelNumber
46     MAX-ACCESS  read-write
47     STATUS      current
48     DESCRIPTION
49         "Downlink channel number as defined in 8.5.
50         Used for license-exempt operation only."
51     REFERENCE
52         "Table 358, in IEEE Std 802.16-2004"
53     ::= { wmanIf2BsOfdmDownlinkChannelEntry 2 }
54
55
56
57 wmanIf2BsOfdmTTG OBJECT-TYPE
58     SYNTAX      INTEGER (0..255)
59     MAX-ACCESS  read-write
60     STATUS      current
61     DESCRIPTION
62         "Transmit / Receive Transition Gap."
63     REFERENCE
64
65

```

```

1         "Table 358, in IEEE Std 802.16-2004"
2     ::= { wmanIf2BsOfdmDownlinkChannelEntry 3 }
3
4
5 wmanIf2BsOfdmRTG OBJECT-TYPE
6     SYNTAX      INTEGER (0..255)
7     MAX-ACCESS  read-write
8     STATUS      current
9     DESCRIPTION
10        "Receive / Transmit Transition Gap."
11    REFERENCE
12        "Table 358, in IEEE Std 802.16-2004"
13    ::= { wmanIf2BsOfdmDownlinkChannelEntry 4 }
14
15
16
17 wmanIf2BsOfdmInitRngMaxRSS OBJECT-TYPE
18     SYNTAX      INTEGER (-32768..32767)
19     UNITS       "dBm"
20     MAX-ACCESS  read-write
21     STATUS      current
22     DESCRIPTION
23        "Initial Ranging Max. equivalent isotropic received power
24         at BS Signed in units of 1 dBm."
25    REFERENCE
26        "Table 358, in IEEE Std 802.16-2004"
27    ::= { wmanIf2BsOfdmDownlinkChannelEntry 5 }
28
29
30
31 wmanIf2BsOfdmDownlinkCenterFreq OBJECT-TYPE
32     SYNTAX      Unsigned32
33     UNITS       "kHz"
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37        "Downlink center frequency (kHz)."
38    REFERENCE
39        "Table 358, in IEEE Std 802.16-2004"
40    ::= { wmanIf2BsOfdmDownlinkChannelEntry 6 }
41
42
43
44 wmanIf2BsOfdmBsId OBJECT-TYPE
45     SYNTAX      WmanIf2BsIdType
46     MAX-ACCESS  read-write
47     STATUS      current
48     DESCRIPTION
49        "Base station ID."
50    REFERENCE
51        "Table 358, in IEEE Std 802.16-2004"
52    ::= { wmanIf2BsOfdmDownlinkChannelEntry 7 }
53
54
55
56 wmanIf2BsOfdmMacVersion OBJECT-TYPE
57     SYNTAX      WmanIf2MacVersion
58     MAX-ACCESS  read-write
59     STATUS      current
60     DESCRIPTION
61        "This parameter specifies the version of 802.16 to which
62         the message originator conforms."
63    REFERENCE
64
65

```



```

1      "Table 358, in IEEE Std 802.16-2004"
2      ::= { wmanIf2BsOfdmDownlinkChannelEntry 8 }
3
4
5  wmanIf2BsOfdmFrameDurationCode OBJECT-TYPE
6      SYNTAX      INTEGER {duration2dot5ms(0),
7                  duration4ms(1),
8                  duration5ms(2),
9                  duration8ms(3),
10                 duration10ms(4),
11                 duration12dot5ms(5),
12                 duration20ms(6)}
13
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "The duration of the frame. The frame duration code
18         values are specified in Table 230."
19     REFERENCE
20         "Subclause 11.4.1, Table 230, in IEEE Std 802.16-2004"
21     ::= { wmanIf2BsOfdmDownlinkChannelEntry 9 }
22
23
24
25  wmanIf2BsOfdmDownLinkChannelId OBJECT-TYPE
26     SYNTAX      INTEGER (0..255)
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "The identifier of the downlink channel to which this
31         message refers."
32     REFERENCE
33         "Subclause 6.3.2.3.1. Table 15, in IEEE Std 802.16-2004"
34     ::= { wmanIf2BsOfdmDownlinkChannelEntry 10 }
35
36
37
38  wmanIf2BsOfdmUcdBurstProfileTable OBJECT-TYPE
39     SYNTAX      SEQUENCE OF WmanIf2BsOfdmUcdBurstProfileEntry
40     MAX-ACCESS  not-accessible
41     STATUS      current
42     DESCRIPTION
43         "This table contains UCD burst profiles for each uplink
44         channel"
45     REFERENCE
46         "Table 356, in IEEE Std 802.16-2004"
47     ::= { wmanIf2BsOfdmPhy 3 }
48
49
50
51  wmanIf2BsOfdmUcdBurstProfileEntry OBJECT-TYPE
52     SYNTAX      WmanIf2BsOfdmUcdBurstProfileEntry
53     MAX-ACCESS  not-accessible
54     STATUS      current
55     DESCRIPTION
56         "This table provides one row for each UCD burst profile.
57         This table is double indexed. The primary index is an
58         ifIndex with an ifType of ieee80216WMAN. The secondary
59         index is wmanIf2BsOfdmUiucIndex."
60     INDEX { ifIndex, wmanIf2BsOfdmUiucIndex }
61     ::= { wmanIf2BsOfdmUcdBurstProfileTable 1 }
62
63
64
65

```

```

1  WmanIf2BsOfdmUcdBurstProfileEntry ::= SEQUENCE {
2      wmanIf2BsOfdmUiucIndex          INTEGER,
3      wmanIf2BsOfdmUcdFecCodeType    WmanIf2OfdmFecCodeType,
4      wmanIf2BsOfdmFocusCtPowerBoost  INTEGER,
5      wmanIf2BsOfdmUcdTcsEnable      INTEGER,
6      wmanIf2BsOfdmUcdBurstProfileRowStatus  RowStatus}
7
8
9
10 wmanIf2BsOfdmUiucIndex OBJECT-TYPE
11     SYNTAX      INTEGER (5 .. 12)
12     MAX-ACCESS  not-accessible
13     STATUS      current
14     DESCRIPTION
15         "The Uplink Interval Usage Code indicates the uplink burst
16         profile in the UCD message, and is used along with ifIndex
17         to identify an entry in the
18         wmanIf2BsOfdmUcdBurstProfileTable."
19     REFERENCE
20         "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
21     ::= { wmanIf2BsOfdmUcdBurstProfileEntry 1 }
22
23
24
25 wmanIf2BsOfdmUcdFecCodeType OBJECT-TYPE
26     SYNTAX      WmanIf2OfdmFecCodeType
27     MAX-ACCESS  read-create
28     STATUS      current
29     DESCRIPTION
30         "Uplink FEC code type and modulation type"
31     REFERENCE
32         "Table 356, in IEEE Std 802.16-2004"
33     ::= { wmanIf2BsOfdmUcdBurstProfileEntry 2 }
34
35
36
37 wmanIf2BsOfdmFocusCtPowerBoost OBJECT-TYPE
38     SYNTAX      INTEGER (0 .. 255)
39     MAX-ACCESS  read-create
40     STATUS      current
41     DESCRIPTION
42         "The power boost in dB of focused contention carriers, as
43         described in 8.3.6.3.3."
44     REFERENCE
45         "Table 356, in IEEE Std 802.16-2004"
46     ::= { wmanIf2BsOfdmUcdBurstProfileEntry 3 }
47
48
49
50 wmanIf2BsOfdmUcdTcsEnable OBJECT-TYPE
51     SYNTAX      INTEGER {tcsDisabled(0),
52                    tcsEnabled(1)}
53     MAX-ACCESS  read-create
54     STATUS      current
55     DESCRIPTION
56         "This parameter determines the transmission convergence
57         sublayer, as described in 8.1.4.3, can be enabled on a
58         per-burst basis for both uplink and downlink. through
59         DIUC/UIUC messages."
60     REFERENCE
61         "Table 356, in IEEE Std 802.16-2004"
62     ::= { wmanIf2BsOfdmUcdBurstProfileEntry 4 }
63
64
65

```

```

1
2
3 wmanIf2BsOfdmUcdBurstProfileRowStatus OBJECT-TYPE
4     SYNTAX      RowStatus
5     MAX-ACCESS  read-create
6     STATUS      current
7     DESCRIPTION
8         "This object is used to create a new row or modify or
9         delete an existing row in this table.
10
11         If the implementator of this MIB has choosen not
12         to implement 'dynamic assignment' of profiles, this
13         object is not useful and should return noSuchName
14         upon SNMP request."
15
16     ::= { wmanIf2BsOfdmUcdBurstProfileEntry 5 }
17
18
19 wmanIf2BsOfdmDcdBurstProfileTable OBJECT-TYPE
20     SYNTAX      SEQUENCE OF WmanIf2BsOfdmDcdBurstProfileEntry
21     MAX-ACCESS  not-accessible
22     STATUS      current
23     DESCRIPTION
24         "This table provides one row for each DCD burst profile.
25         This table is double indexed. The primary index is an
26         ifIndex with an ifType of ieee80216WMAN. The secondary
27         index is wmanIf2BsOfdmDiucIndex."
28
29     REFERENCE
30         "Table 362, in IEEE Std 802.16-2004"
31
32     ::= { wmanIf2BsOfdmPhy 4 }
33
34
35 wmanIf2BsOfdmDcdBurstProfileEntry OBJECT-TYPE
36     SYNTAX      WmanIf2BsOfdmDcdBurstProfileEntry
37     MAX-ACCESS  not-accessible
38     STATUS      current
39     DESCRIPTION
40         "This table provides one row for each DCD burst profile.
41         This table is double indexed. The primary index is an
42         ifIndex with an ifType of ieee80216WMAN. The secondary
43         index is wmanIf2BsOfdmDiucIndex."
44
45     INDEX { ifIndex, wmanIf2BsOfdmDiucIndex }
46
47     ::= { wmanIf2BsOfdmDcdBurstProfileTable 1 }
48
49 WmanIf2BsOfdmDcdBurstProfileEntry ::= SEQUENCE {
50     wmanIf2BsOfdmDiucIndex          INTEGER,
51     wmanIf2BsOfdmDownlinkFrequency Unsigned32,
52     wmanIf2BsOfdmDcdFecCodeType    WmanIf2OfdmFecCodeType,
53     wmanIf2BsOfdmDiucMandatoryExitThresh INTEGER,
54     wmanIf2BsOfdmDiucMinEntryThresh INTEGER,
55     wmanIf2BsOfdmTcsEnable         INTEGER,
56     wmanIf2BsOfdmDcdBurstProfileRowStatus RowStatus }
57
58
59
60 wmanIf2BsOfdmDiucIndex OBJECT-TYPE
61     SYNTAX      INTEGER (1..11)
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65

```

```

1         "The Downlink Interval Usage Code indicates the downlink
2         burst profile in the DCD message, and is used along with
3         ifIndex to identify an entry in the
4         wmanIf2BsOfdmDcdBurstProfileTable."
5
6     REFERENCE
7         "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
8     ::= { wmanIf2BsOfdmDcdBurstProfileEntry 1 }
9
10
11 wmanIf2BsOfdmDownlinkFrequency OBJECT-TYPE
12     SYNTAX      Unsigned32
13     UNITS       "kHz"
14     MAX-ACCESS  read-create
15     STATUS      current
16     DESCRIPTION
17         "Downlink Frequency (kHz)."

```

```

1      SYNTAX      INTEGER {tcsDisabled (0),
2                      tcsEnabled (1)}
3
4      MAX-ACCESS  read-create
5      STATUS      current
6      DESCRIPTION
7          "Indicates whether Transmission COvergence Sublayer
8              is enabled or disabled."
9
10     REFERENCE
11         "Table 362, in IEEE Std 802.16-2004"
12     ::= { wmanIf2BsOfdmDcdBurstProfileEntry 6 }
13
14     wmanIf2BsOfdmDcdBurstProfileRowStatus OBJECT-TYPE
15     SYNTAX      RowStatus
16     MAX-ACCESS  read-create
17     STATUS      current
18     DESCRIPTION
19         "This object is used to create a new row or modify or
20             delete an existing row in this table.
21
22             If the implementator of this MIB has choosen not
23             to implement 'dynamic assignment' of profiles, this
24             object is not useful and should return noSuchName
25             upon SNMP request."
26     ::= { wmanIf2BsOfdmDcdBurstProfileEntry 7 }
27
28     wmanIf2BsOfdmConfigurationTable OBJECT-TYPE
29     SYNTAX      SEQUENCE OF WmanIf2BsOfdmConfigurationEntry
30     MAX-ACCESS  not-accessible
31     STATUS      current
32     DESCRIPTION
33         "This table contains BS configuration objects, specific to
34             OFDM PHY."
35     ::= { wmanIf2BsOfdmPhy 5 }
36
37     wmanIf2BsOfdmConfigurationEntry OBJECT-TYPE
38     SYNTAX      WmanIf2BsOfdmConfigurationEntry
39     MAX-ACCESS  not-accessible
40     STATUS      current
41     DESCRIPTION
42         "This table is indexed by ifIndex with an ifType of
43             ieee80216WMAN."
44     INDEX { ifIndex }
45     ::= { wmanIf2BsOfdmConfigurationTable 1 }
46
47     WmanIf2BsOfdmConfigurationEntry ::= SEQUENCE {
48         wmanIf2BsOfdmMinReqRegionFullTxOpp      INTEGER,
49         wmanIf2BsOfdmMinFocusedCtTxOpp          INTEGER,
50         wmanIf2BsOfdmMaxRoundTripDelay          INTEGER,
51         wmanIf2BsOfdmRangeAbortTimingThold      INTEGER,
52         wmanIf2BsOfdmRangeAbortPowerThold       INTEGER,
53         wmanIf2BsOfdmRangeAbortFreqThold        INTEGER,
54         wmanIf2BsOfdmDnlkRateId                 INTEGER,
55         wmanIf2BsOfdmRatioG                     INTEGER}
56
57
58
59
60
61
62
63
64
65

```

```

1  wmanIf2BsOfdmMinReqRegionFullTxOpp OBJECT-TYPE
2      SYNTAX      INTEGER (1..65535)
3      UNITS       "1/sec"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "The minimum number of Full bandwidth Req-Region Full
8          Transmit opportunities scheduled in the UL per second."
9      REFERENCE
10         "Subclause 6.3.7.4.3 in IEEE Std 802.16-2004"
11         ::= { wmanIf2BsOfdmConfigurationEntry 1 }
12
13  wmanIf2BsOfdmMinFocusedCtTxOpp OBJECT-TYPE
14      SYNTAX      INTEGER (0..65535)
15      UNITS       "1/sec"
16      MAX-ACCESS  read-write
17      STATUS      current
18      DESCRIPTION
19          "The minimum number of focused contention Transmit
20          opportunities scheduled in the UL per second. The value may
21          be 0 if the focused contention is not implemented."
22      REFERENCE
23         "Subclauses 6.3.6.4 and 8.3.7.3.3 in IEEE Std 802.16-2004"
24         ::= { wmanIf2BsOfdmConfigurationEntry 2 }
25
26  wmanIf2BsOfdmMaxRoundTripDelay OBJECT-TYPE
27      SYNTAX      INTEGER (1..65535)
28      UNITS       "us"
29      MAX-ACCESS  read-write
30      STATUS      current
31      DESCRIPTION
32          "Maximum supported round trip delay.
33          It is required to limit the cell size."
34      REFERENCE
35         "Subclause 8.3.5.1 in IEEE Std 802.16-2004"
36         ::= { wmanIf2BsOfdmConfigurationEntry 3 }
37
38  wmanIf2BsOfdmRangeAbortTimingThold OBJECT-TYPE
39      SYNTAX      INTEGER (0..255)
40      UNITS       "1/Fs"
41      MAX-ACCESS  read-write
42      STATUS      current
43      DESCRIPTION
44          "This object defines Tolerable Timing Offset. BS performs
45          Initial Ranging until the SS transmissions are within
46          limits that are deemed tolerable by the BS. If the SS does
47          not transmit within these limits after a number of
48          correction attempts then the BS aborts Initial Ranging."
49      REFERENCE
50         "Figure 63 and Table 365 in IEEE Std 802.16-2004"
51         ::= { wmanIf2BsOfdmConfigurationEntry 4 }
52
53  wmanIf2BsOfdmRangeAbortPowerThold OBJECT-TYPE
54      SYNTAX      INTEGER (0..255)
55

```

```

1      UNITS          "0.25dB"
2      MAX-ACCESS    read-write
3      STATUS        current
4      DESCRIPTION
5
6          "This object defines Tolerable Power Offset. BS performs
7          Initial Ranging until the SS transmissions are within
8          limits that are deemed tolerable by the BS. If the SS does
9          not transmit within these limits after a number of
10         correction attempts then the BS aborts Initial Ranging."
11
12     REFERENCE
13         "Figure 63 and Table 365 in IEEE Std 802.16-2004"
14     ::= { wmanIf2BsOfdmConfigurationEntry 5 }
15
16 wmanIf2BsOfdmRangeAbortFreqThold OBJECT-TYPE
17     SYNTAX          INTEGER (0..255)
18     UNITS           "Hz"
19     MAX-ACCESS      read-write
20     STATUS          current
21     DESCRIPTION
22
23         "This object defines Tolerable Frequency Offset. BS performs
24         Initial Ranging until the SS transmissions are within
25         limits that are deemed tolerable by the BS. If the SS does
26         not transmit within these limits after a number of
27         correction attempts then the BS aborts Initial Ranging."
28
29     REFERENCE
30         "Figure 63 and Table 365 in IEEE Std 802.16-2004"
31     ::= { wmanIf2BsOfdmConfigurationEntry 6 }
32
33
34 wmanIf2BsOfdmDnlkRateId OBJECT-TYPE
35     SYNTAX          INTEGER { dnlkRateIdBpsk1Over2(0),
36                             dnlkRateIdQpsk1Over2(1),
37                             dnlkRateIdQpsk3Over4(2),
38                             dnlkRateId16Qam1Over2(3),
39                             dnlkRateId16Qam3Over4(4),
40                             dnlkRateId64Qam2Over3(5),
41                             dnlkRateId64Qam3Over4(6) }
42     MAX-ACCESS      read-write
43     STATUS          current
44     DESCRIPTION
45
46         "The Rate ID to be used in the first downlink burst
47         immediately following the FCH. The Rate ID encoding is
48         static and cannot be changed during system operation. The
49         change of the Rate ID should be applied on system
50         re-intialisation (e.g. following sector or BS reset)."

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "The ratio of CP time to 'useful' time.Values
5          are 1/4, 1/8, 1/16 or 1/32."
6      REFERENCE
7          "Subclause 8.3.1.1.1 in IEEE Std 802.16-2004"
8      DEFVAL      { ratio1To4 }
9      ::= { wmanIf2BsOfdmConfigurationEntry 8 }
10
11
12
13 wmanIf2BsSsOfdmReqCapabilitiesTable OBJECT-TYPE
14     SYNTAX      SEQUENCE OF WmanIf2BsSsOfdmReqCapabilitiesEntry
15     MAX-ACCESS  not-accessible
16     STATUS      current
17     DESCRIPTION
18         "This table contains the basic capability information,
19         specific to OFDM Phy, of Ss that have been reported by
20         Ss to BS using RNG-REQ, SBC-REQ and REG-REQ messages.
21         Entries in this table should be created when an SS
22         registers with a BS."
23     ::= { wmanIf2BsOfdmPhy 6 }
24
25
26
27 wmanIf2BsSsOfdmReqCapabilitiesEntry OBJECT-TYPE
28     SYNTAX      WmanIf2BsSsOfdmReqCapabilitiesEntry
29     MAX-ACCESS  not-accessible
30     STATUS      current
31     DESCRIPTION
32         "This table provides one row for each SS that has been
33         registered in the BS. This table augments the table
34         wmanIf2BsRegisteredSsTable."
35     AUGMENTS { wmanIf2BsRegisteredSsEntry }
36     ::= { wmanIf2BsSsOfdmReqCapabilitiesTable 1 }
37
38
39
40
41 WmanIf2BsSsOfdmReqCapabilitiesEntry ::= SEQUENCE {
42     wmanIf2BsSsOfdmReqCapFftSizes          WmanIf2OfdmFftSizes,
43     wmanIf2BsSsOfdmReqCapSsDemodulator     WmanIf2OfdmSsDeModType,
44     wmanIf2BsSsOfdmReqCapSsModulator       WmanIf2OfdmSsModType,
45     wmanIf2BsSsOfdmReqCapFocusedCtSupport  WmanIf2OfdmFocusedCt,
46     wmanIf2BsSsOfdmReqCapTcSublayerSupport WmanIf2OfdmTcSublayer,
47     wmanIf2BsSsOfdmReqCapPrivteMapSupport  WmanIf2OfdmPrivMap,
48     wmanIf2BsSsOfdmReqCapUlPowerControl    WmanIf2OfdmUlPower,
49     wmanIf2BsSsOfdmReqCapLoopPwrControlSw  Unsigned32 }
50
51
52
53 wmanIf2BsSsOfdmReqCapFftSizes OBJECT-TYPE
54     SYNTAX      WmanIf2OfdmFftSizes
55     MAX-ACCESS  read-only
56     STATUS      current
57     DESCRIPTION
58         "This field indicates the FFT sizes supported by SS.
59         The usage is defined by WmanIf2OfdmFftSizes."
60     ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 1 }
61
62
63
64 wmanIf2BsSsOfdmReqCapSsDemodulator OBJECT-TYPE
65     SYNTAX      WmanIf2OfdmSsDeModType

```



```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This field indicates the different demodulator options
5          supported by SS for downlink.
6          The usage is defined by WmanIf2OfdmSsDeModType."
7      ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 2 }
8
9
10     wmanIf2BsSsOfdmReqCapSsModulator OBJECT-TYPE
11     SYNTAX      WmanIf2OfdmSsModType
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "This field indicates the different modulator options
16         supported by SS for uplink.
17         The usage is defined by WmanIf2OfdmSsModType."
18     ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 3 }
19
20
21     wmanIf2BsSsOfdmReqCapFocusedCtSupport OBJECT-TYPE
22     SYNTAX      WmanIf2OfdmFocusedCt
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "This field indicates whether the SS supports Focused
27         Contention. The usage is defined by
28         WmanIf2OfdmFocusedCt."
29     ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 4 }
30
31
32     wmanIf2BsSsOfdmReqCapTcSublayerSupport OBJECT-TYPE
33     SYNTAX      WmanIf2OfdmTcSublayer
34     MAX-ACCESS  read-only
35     STATUS      current
36     DESCRIPTION
37         "This field indicates whether or not the SS supports
38         the TC sublayer. The usage is defined by
39         WmanIf2OfdmTcSublayer."
40     ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 5 }
41
42
43     wmanIf2BsSsOfdmReqCapPrivteMapSupport OBJECT-TYPE
44     SYNTAX      WmanIf2OfdmPrivMap
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "This field indicates if the private map parameters
49         is supported."
50     ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 6 }
51
52
53     wmanIf2BsSsOfdmReqCapUlPowerControl OBJECT-TYPE
54     SYNTAX      WmanIf2OfdmUlPower
55     MAX-ACCESS  read-only
56     STATUS      current
57     DESCRIPTION
58         "This field indicates the uplink power control options
59         supported by SS."
60
61
62
63
64
65

```

```

1      ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 7 }
2
3
4      wmanIf2BsSsOfdmReqCapLoopPwrControlSw OBJECT-TYPE
5          SYNTAX      Unsigned32
6          MAX-ACCESS  read-only
7          STATUS      current
8          DESCRIPTION
9              "This field indicates the minimum number of frames that
10             SS takes to switch from the open loop power control
11             scheme to the closed loop power control scheme or
12             vice versa."
13
14      ::= { wmanIf2BsSsOfdmReqCapabilitiesEntry 8 }
15
16
17      wmanIf2BsSsOfdmRspCapabilitiesTable OBJECT-TYPE
18          SYNTAX      SEQUENCE OF WmanIf2BsSsOfdmRspCapabilitiesEntry
19          MAX-ACCESS  not-accessible
20          STATUS      current
21          DESCRIPTION
22              "This table contains the basic capability information,
23             specific to OFDM Phy, of SSS that have been negotiated
24             and agreed between BS and SS via RNG-REQ/RSP,
25             SBC-REQ/RSP and REG-REQ/RSP messages. This table
26             augments the wmanIf2BsRegisteredSsTable."
27
28          REFERENCE
29              "Subclause 6.3.2.3.7 in IEEE Std 802.16-2004"
30
31      ::= { wmanIf2BsOfdmPhy 7 }
32
33
34      wmanIf2BsSsOfdmRspCapabilitiesEntry OBJECT-TYPE
35          SYNTAX      WmanIf2BsSsOfdmRspCapabilitiesEntry
36          MAX-ACCESS  not-accessible
37          STATUS      current
38          DESCRIPTION
39              "This table provides one row for each SS that has been
40             registered in the BS. This table augments the
41             wmanIf2BsRegisteredSsTable. "
42
43          AUGMENTS { wmanIf2BsRegisteredSsEntry }
44      ::= { wmanIf2BsSsOfdmRspCapabilitiesTable 1 }
45
46
47      WmanIf2BsSsOfdmRspCapabilitiesEntry ::= SEQUENCE {
48          wmanIf2BsSsOfdmRspCapFftSizes      WmanIf2OfdmFftSizes,
49          wmanIf2BsSsOfdmRspCapSsDemodulator WmanIf2OfdmSsDeModType,
50          wmanIf2BsSsOfdmRspCapSsModulator   WmanIf2OfdmSsModType,
51          wmanIf2BsSsOfdmRspCapFocusedCtSupport WmanIf2OfdmFocusedCt,
52          wmanIf2BsSsOfdmRspCapTcSublayerSupport WmanIf2OfdmTcSublayer,
53          wmanIf2BsSsOfdmRspCapPrivteMapSupport WmanIf2OfdmPrivMap,
54          wmanIf2BsSsOfdmRspCapUlPowerControl WmanIf2OfdmUlPower,
55          wmanIf2BsSsOfdmRspCapLoopPwrControlSw Unsigned32}
56
57
58
59      wmanIf2BsSsOfdmRspCapFftSizes OBJECT-TYPE
60          SYNTAX      WmanIf2OfdmFftSizes
61          MAX-ACCESS  read-only
62          STATUS      current
63          DESCRIPTION
64              "This field indicates the FFT sizes negotiated with the
65

```

```

1         SS. The usage is defined by WmanIf2OfdmFftSizes."
2     ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 1 }
3
4
5 wmanIf2BsSsOfdmRspCapSsDemodulator OBJECT-TYPE
6     SYNTAX      WmanIf2OfdmSsDeModType
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10        "This field indicates the different demodulator options
11        negotiated for SS for downlink. The usage is defined by
12        WmanIf2OfdmSsDeModType."
13    ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 2 }
14
15
16 wmanIf2BsSsOfdmRspCapSsModulator OBJECT-TYPE
17     SYNTAX      WmanIf2OfdmSsModType
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21        "This field indicates the different modulator options
22        negotiated for SS for uplink. The usage is defined by
23        WmanIf2OfdmSsModType."
24    ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 3 }
25
26
27
28 wmanIf2BsSsOfdmRspCapFocusedCtSupport OBJECT-TYPE
29     SYNTAX      WmanIf2OfdmFocusedCt
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33        "This field indicates whether the SS has negotiated the
34        support for Focused Contention. The usage is defined by
35        WmanIf2OfdmFocusedCt."
36    ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 4 }
37
38
39
40 wmanIf2BsSsOfdmRspCapTcSublayerSupport OBJECT-TYPE
41     SYNTAX      WmanIf2OfdmTcSublayer
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45        "This field indicates whether the SS has negotiated
46        support for the TC sublayer. The usage is defined by
47        WmanIf2OfdmTcSublayer."
48    ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 5 }
49
50
51
52 wmanIf2BsSsOfdmRspCapPrivteMapSupport OBJECT-TYPE
53     SYNTAX      WmanIf2OfdmPrivMap
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57        "This field iindicates if the private map parameters
58        is supported."
59    ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 6 }
60
61
62
63 wmanIf2BsSsOfdmRspCapUlPowerControl OBJECT-TYPE
64     SYNTAX      WmanIf2OfdmUlPower
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This field indicates the uplink power control options
5          supported by SS."
6      ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 7 }
7
8
9
10     wmanIf2BsSsOfdmRspCapLoopPwrControlSw OBJECT-TYPE
11         SYNTAX      Unsigned32
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "This field indicates the minimum number of frames that
16             SS takes to switch from the open loop power control
17             scheme to the closed loop power control scheme or
18             vice versa."
19         ::= { wmanIf2BsSsOfdmRspCapabilitiesEntry 8 }
20
21
22
23     wmanIf2BsOfdmCapabilitiesTable OBJECT-TYPE
24         SYNTAX      SEQUENCE OF WmanIf2BsOfdmCapabilitiesEntry
25         MAX-ACCESS  not-accessible
26         STATUS      current
27         DESCRIPTION
28             "This table contains the basic capabilities, specific to
29             OFDM Phy, of the BS as implemented in BS hardware and
30             software. These capabilities along with the configuration
31             for them (wmanIf2BsOfdmCapabilitiesConfigTable) are used
32             for negotiation of basic capabilities with SS using
33             RNG-RSP, SBC-RSP and REG-RSP messages. The negotiated
34             capabilities are obtained by interSubclause of SS raw
35             reported capabilities, BS raw capabilities and BS
36             configured capabilities. The objects in the table have
37             read-only access. The table is maintained by BS."
38         ::= { wmanIf2BsOfdmPhy 8 }
39
40
41
42
43     wmanIf2BsOfdmCapabilitiesEntry OBJECT-TYPE
44         SYNTAX      WmanIf2BsOfdmCapabilitiesEntry
45         MAX-ACCESS  not-accessible
46         STATUS      current
47         DESCRIPTION
48             "This table provides one row for each BS sector and is
49             indexed by ifIndex."
50         INDEX { ifIndex }
51         ::= { wmanIf2BsOfdmCapabilitiesTable 1 }
52
53
54
55     WmanIf2BsOfdmCapabilitiesEntry ::= SEQUENCE {
56         wmanIf2BsOfdmCapFftSizes          WmanIf2OfdmFftSizes,
57         wmanIf2BsOfdmCapSsDemodulator     WmanIf2OfdmSsDeModType,
58         wmanIf2BsOfdmCapSsModulator       WmanIf2OfdmSsModType,
59         wmanIf2BsOfdmCapFocusedCtSupport  WmanIf2OfdmFocusedCt,
60         wmanIf2BsOfdmCapTcSublayerSupport WmanIf2OfdmTcSublayer,
61         wmanIf2BsOfdmCapPrivteMapSupport  WmanIf2OfdmPrivMap,
62         wmanIf2BsSsOfdmCapUlPowerControl  WmanIf2OfdmUlPower,
63         wmanIf2BsSsOfdmCapLoopPwrControlSw Unsigned32 }
64
65

```

```

1
2
3 wmanIf2BsOfdmCapFftSizes OBJECT-TYPE
4     SYNTAX      WmanIf2OfdmFftSizes
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "This field indicates the FFT sizes supported by the BS.
9         The usage is defined by WmanIf2OfdmFftSizes."
10    ::= { wmanIf2BsOfdmCapabilitiesEntry 1 }
11
12
13 wmanIf2BsOfdmCapSsDemodulator OBJECT-TYPE
14     SYNTAX      WmanIf2OfdmSsDeModType
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "This field indicates the different BS demodulator options
19         for uplink supported by the BS. The usage is defined by
20         WmanIf2OfdmSsDeModType."
21    ::= { wmanIf2BsOfdmCapabilitiesEntry 2 }
22
23
24
25 wmanIf2BsOfdmCapSsModulator OBJECT-TYPE
26     SYNTAX      WmanIf2OfdmSsModType
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "This field indicates the different BS modulator options
31         for downlink supported by the BS. The usage is defined by
32         WmanIf2OfdmSsModType."
33    ::= { wmanIf2BsOfdmCapabilitiesEntry 3 }
34
35
36
37 wmanIf2BsOfdmCapFocusedCtSupport OBJECT-TYPE
38     SYNTAX      WmanIf2OfdmFocusedCt
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "This field indicates the BS support for Focused
43         Contention. The usage is defined by
44         WmanIf2OfdmFocusedCt."
45    ::= { wmanIf2BsOfdmCapabilitiesEntry 4 }
46
47
48
49 wmanIf2BsOfdmCapTcSublayerSupport OBJECT-TYPE
50     SYNTAX      WmanIf2OfdmTcSublayer
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "This field indicates the BS supports for TC sublayer. The
55         usage is defined by WmanIf2OfdmTcSublayer."
56    ::= { wmanIf2BsOfdmCapabilitiesEntry 5 }
57
58
59
60 wmanIf2BsOfdmCapPrivteMapSupport OBJECT-TYPE
61     SYNTAX      WmanIf2OfdmPrivMap
62     MAX-ACCESS  read-only
63     STATUS      current
64     DESCRIPTION
65

```

```

1         "This field indicates if the private map parameters
2         is supported."
3         ::= { wmanIf2BsOfdmCapabilitiesEntry 6 }
4
5
6 wmanIf2BsSsOfdmCapUlPowerControl OBJECT-TYPE
7     SYNTAX      WmanIf2OfdmUlPower
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11        "This field indicates the uplink power control options
12        supported by SS."
13        ::= { wmanIf2BsOfdmCapabilitiesEntry 7 }
14
15
16 wmanIf2BsSsOfdmCapLoopPwrControlSw OBJECT-TYPE
17     SYNTAX      Unsigned32
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21        "This field indicates the minimum number of frames that
22        SS takes to switch from the open loop power control
23        scheme to the closed loop power control scheme or
24        vice versa."
25        ::= { wmanIf2BsOfdmCapabilitiesEntry 8 }
26
27
28 wmanIf2BsOfdmCapabilitiesConfigTable OBJECT-TYPE
29     SYNTAX      SEQUENCE OF WmanIf2BsOfdmCapabilitiesConfigEntry
30     MAX-ACCESS  not-accessible
31     STATUS      current
32     DESCRIPTION
33        "This table contains the configuration for basic
34        capabilities of BS, specific to OFDM Phy. The table is
35        intended to be used to restrict the Capabilities
36        implemented by BS, for example in order to comply with
37        local regulatory requirements. The BS should use the
38        configuration along with the implemented Capabilities
39        (wmanIf2BsOfdmPhyTable) for negotiation of basic
40        capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP
41        messages. The negotiated capabilities are obtained by
42        interSubclause of SS reported capabilities, BS raw
43        capabilities and BS configured capabilities. The objects
44        in the table have read-write access. The rows are created
45        by BS as a copy of wmanIf2BsBasicCapabilitiesTable
46        and can be modified by NMS."
47        ::= { wmanIf2BsOfdmPhy 9 }
48
49
50 wmanIf2BsOfdmCapabilitiesConfigEntry OBJECT-TYPE
51     SYNTAX      WmanIf2BsOfdmCapabilitiesConfigEntry
52     MAX-ACCESS  not-accessible
53     STATUS      current
54     DESCRIPTION
55        "This table provides one row for each BS sector and is
56        indexed by ifIndex."
57     INDEX { ifIndex }
58     ::= { wmanIf2BsOfdmCapabilitiesConfigTable 1 }
59
60
61
62
63
64
65

```

```

1
2 WmanIf2BsOfdmCapabilitiesConfigEntry ::= SEQUENCE {
3     wmanIf2BsOfdmCapCfgFftSizes           WmanIf2OfdmFftSizes,
4     wmanIf2BsOfdmCapCfgSsDemodulator      WmanIf2OfdmSsDeModType,
5     wmanIf2BsOfdmCapCfgSsModulator        WmanIf2OfdmSsModType,
6     wmanIf2BsOfdmCapCfgFocusedCtSupport   WmanIf2OfdmFocusedCt,
7     wmanIf2BsOfdmCapCfgTcSublayerSupport  WmanIf2OfdmTcSublayer,
8     wmanIf2BsOfdmCapCfgPrivteMapSupport   WmanIf2OfdmPrivMap,
9     wmanIf2BsSsOfdmCapCfgUlPowerControl   WmanIf2OfdmUlPower,
10    wmanIf2BsSsOfdmCapCfgLoopPwrControlSw  Unsigned32}
11
12
13
14 wmanIf2BsOfdmCapCfgFftSizes OBJECT-TYPE
15     SYNTAX      WmanIf2OfdmFftSizes
16     MAX-ACCESS  read-write
17     STATUS      current
18     DESCRIPTION
19         "This field indicates the FFT sizes support configured for
20         the BS. The usage is defined by WmanIf2OfdmFftSizes."
21     ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 1 }
22
23
24
25 wmanIf2BsOfdmCapCfgSsDemodulator OBJECT-TYPE
26     SYNTAX      WmanIf2OfdmSsDeModType
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "This field indicates the different BS demodulator options
31         configured for uplink. The usage is defined by
32         WmanIf2OfdmSsDeModType."
33     ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 2 }
34
35
36
37 wmanIf2BsOfdmCapCfgSsModulator OBJECT-TYPE
38     SYNTAX      WmanIf2OfdmSsModType
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "This field indicates the different BS modulator options
43         configured for downlink. The usage is defined by
44         WmanIf2OfdmSsModType."
45     ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 3 }
46
47
48
49 wmanIf2BsOfdmCapCfgFocusedCtSupport OBJECT-TYPE
50     SYNTAX      WmanIf2OfdmFocusedCt
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "This field indicates the BS support configured for
55         Focused Contention. The usage is defined by
56         WmanIf2OfdmFocusedCt."
57     ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 4 }
58
59
60
61 wmanIf2BsOfdmCapCfgTcSublayerSupport OBJECT-TYPE
62     SYNTAX      WmanIf2OfdmTcSublayer
63     MAX-ACCESS  read-write
64     STATUS      current
65

```

```

1      DESCRIPTION
2          "This field indicates the BS support configured for TC
3          sublayer. The usage is defined by
4          WmanIf2OfdmTcSublayer."
5          ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 5 }
6
7
8      wmanIf2BsOfdmCapCfgPrivteMapSupport OBJECT-TYPE
9          SYNTAX      WmanIf2OfdmPrivMap
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "This field iindicates if the private map parameters
14             is supported."
15             ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 6 }
16
17
18
19     wmanIf2BsSsOfdmCapCfgUlPowerControl OBJECT-TYPE
20         SYNTAX      WmanIf2OfdmUlPower
21         MAX-ACCESS  read-write
22         STATUS      current
23         DESCRIPTION
24             "This field indicates the uplink power control options
25             supported by SS."
26             ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 7 }
27
28
29
30     wmanIf2BsSsOfdmCapCfgLoopPwrControlSw OBJECT-TYPE
31         SYNTAX      Unsigned32
32         MAX-ACCESS  read-write
33         STATUS      current
34         DESCRIPTION
35             "This field indicates he minimum number of frames that
36             SS takes to switch from the open loop power control
37             scheme to the closed loop power control scheme or
38             vice versa."
39             ::= { wmanIf2BsOfdmCapabilitiesConfigEntry 8 }
40
41
42
43     --
44     -- BS OFDMA PHY objects
45     --
46     wmanIf2BsOfdmaPhy OBJECT IDENTIFIER ::= { wmanIf2BsPhy 2 }
47
48
49     wmanIf2BsOfdmaUplinkChannelTable OBJECT-TYPE
50         SYNTAX      SEQUENCE OF WmanIf2BsOfdmaUplinkChannelEntry
51         MAX-ACCESS  not-accessible
52         STATUS      current
53         DESCRIPTION
54             "This table contains UCD channel attributes, defining the
55             transmission characteristics of uplink channels"
56         REFERENCE
57             "Table 349 and Table 353, in IEEE Std 802.16-2004"
58             ::= { wmanIf2BsOfdmaPhy 1 }
59
60
61
62     wmanIf2BsOfdmaUplinkChannelEntry OBJECT-TYPE
63         SYNTAX      WmanIf2BsOfdmaUplinkChannelEntry
64         MAX-ACCESS  not-accessible
65

```



```

1      STATUS      current
2      DESCRIPTION
3          "This table provides one row for each uplink channel of
4          multi-sector BS, and is indexed by BS ifIndex. An entry
5          in this table exists for each ifEntry of BS with an
6          ifType of ieee80216WMAN."
7      INDEX      { ifIndex }
8      ::= { wmanIf2BsOfdmaUplinkChannelTable 1 }
9
10
11
12  WmanIf2BsOfdmaUplinkChannelEntry ::= SEQUENCE {
13      wmanIf2BsOfdmaCtBasedResvTimeout      INTEGER,
14      wmanIf2BsOfdmaBwReqOppSize            INTEGER,
15      wmanIf2BsOfdmaRangReqOppSize          INTEGER,
16      wmanIf2BsOfdmaUplinkCenterFreq        Unsigned32,
17      wmanIf2BsOfdmaInitRngCodes            INTEGER,
18      wmanIf2BsOfdmaPeriodicRngCodes        INTEGER,
19      wmanIf2BsOfdmaBWRngCodes              INTEGER,
20      wmanIf2BsOfdmaPerRngBackoffStart      INTEGER,
21      wmanIf2BsOfdmaPerRngBackoffEnd        INTEGER,
22      wmanIf2BsOfdmaStartOfRngCodes         INTEGER,
23      wmanIf2BsOfdmaPermutationBase         INTEGER,
24      wmanIf2BsOfdmaULAllocSubchBitmap      OCTET STRING,
25      wmanIf2BsOfdmaOptPermULAllocSubchBitmap OCTET STRING,
26      wmanIf2BsOfdmaBandAMCAllocThreshold   INTEGER,
27      wmanIf2BsOfdmaBandAMCReleaseThreshold INTEGER,
28      wmanIf2BsOfdmaBandAMCAllocTimer       INTEGER,
29      wmanIf2BsOfdmaBandAMCReleaseTimer     INTEGER,
30      wmanIf2BsOfdmaBandStatRepMAXPeriod    INTEGER,
31      wmanIf2BsOfdmaBandAMCRetryTimer       INTEGER,
32      wmanIf2BsOfdmaSafetyChAllocThreshold  INTEGER,
33      wmanIf2BsOfdmaSafetyChReleaseThreshold INTEGER,
34      wmanIf2BsOfdmaSafetyChAllocTimer      INTEGER,
35      wmanIf2BsOfdmaSafetyChReleaseTimer    INTEGER,
36      wmanIf2BsOfdmaBinStatRepMAXPeriod     INTEGER,
37      wmanIf2BsOfdmaSafetyChARetryTimer     INTEGER,
38      wmanIf2BsOfdmaHARQAackDelayULBurst    INTEGER,
39      wmanIf2BsOfdmaCQICHBandAMCTranaDelay  INTEGER}
40
41
42
43
44
45
46
47  wmanIf2BsOfdmaCtBasedResvTimeout OBJECT-TYPE
48      SYNTAX      INTEGER (1..255)
49      MAX-ACCESS  read-write
50      STATUS      current
51      DESCRIPTION
52          "The number of UL-MAPs to receive before contention-based
53          reservation is attempted again for the same connection."
54      REFERENCE
55          "Table 349, in IEEE Std 802.16-2004"
56      ::= { wmanIf2BsOfdmaUplinkChannelEntry 1 }
57
58
59
60  wmanIf2BsOfdmaBwReqOppSize OBJECT-TYPE
61      SYNTAX      INTEGER (1..65535)
62      UNITS       "PS"
63      MAX-ACCESS  read-write
64      STATUS      deprecated
65

```

```

1      DESCRIPTION
2          "Size (in units of PS) of PHY payload that SS may use to
3          format and transmit a bandwidth request message in a
4          contention request opportunity. The value includes all
5          PHY overhead as well as allowance for the MAC data the
6          message may hold."
7
8      REFERENCE
9          "Table 349, in IEEE Std 802.16-2004"
10         ::= { wmanIf2BsOfdmaUplinkChannelEntry 2 }
11
12
13 wmanIf2BsOfdmaRangReqOppSize OBJECT-TYPE
14     SYNTAX      INTEGER (1..65535)
15     UNITS       "PS"
16     MAX-ACCESS  read-write
17     STATUS      deprecated
18     DESCRIPTION
19         "Size (in units of PS) of PHY payload that SS may use to
20         format and transmit a RNG-REQ message in a contention
21         request opportunity. The value includes all PHY overhead
22         as well as allowance for the MAC data the message may
23         hold and the maximum SS/BS roundtrip propagation delay."
24
25     REFERENCE
26         "Table 352, in IEEE Std 802.16e-2005"
27         ::= { wmanIf2BsOfdmaUplinkChannelEntry 3 }
28
29
30
31 wmanIf2BsOfdmaUplinkCenterFreq OBJECT-TYPE
32     SYNTAX      Unsigned32
33     UNITS       "kHz"
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37         " Uplink center frequency (kHz) "
38
39     REFERENCE
40         "Table 349, in IEEE Std 802.16-2004"
41         ::= { wmanIf2BsOfdmaUplinkChannelEntry 4 }
42
43
44
45 wmanIf2BsOfdmaInitRngCodes OBJECT-TYPE
46     SYNTAX      INTEGER (0..255)
47     MAX-ACCESS  read-write
48     STATUS      deprecated
49     DESCRIPTION
50         "Number of initial ranging CDMA codes. Possible values are
51         0..255. The total number of wmanIf2BsOfdmaInitRngCodes,
52         wmanIf2BsOfdmaPeriodicRngCodes and wmanIf2BsOfdmaBWReqCodes
53         shall be equal or less than 256."
54
55     REFERENCE
56         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
57     DEFVAL      { 30 }
58     ::= { wmanIf2BsOfdmaUplinkChannelEntry 5 }
59
60
61
62 wmanIf2BsOfdmaPeriodicRngCodes OBJECT-TYPE
63     SYNTAX      INTEGER (0..255)
64     MAX-ACCESS  read-write
65     STATUS      deprecated

```

```

1      DESCRIPTION
2          "Number of periodic ranging CDMA codes. Possible values are
3              0..255. The total number of wmanIf2BsOfdmaInitRngCodes,
4              wmanIf2BsOfdmaPeriodicRngCodes and wmanIf2BsOfdmaBWReqCodes
5              shall be equal or less than 256."
6
7      REFERENCE
8          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
9
10     DEFVAL      { 30 }
11     ::= { wmanIf2BsOfdmaUplinkChannelEntry 6 }
12
13     wmanIf2BsOfdmaBWReqCodes OBJECT-TYPE
14         SYNTAX      INTEGER (0..255)
15         MAX-ACCESS  read-write
16         STATUS      deprecated
17         DESCRIPTION
18             "Number of bandwidth request codes. Possible values are
19                 0..255. The total number of wmanIf2BsOfdmaInitRngCodes,
20                 wmanIf2BsOfdmaPeriodicRngCodes and wmanIf2BsOfdmaBWReqCodes
21                 shall be equal or less than 256."
22
23         REFERENCE
24             "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
25
26         DEFVAL      { 30 }
27         ::= { wmanIf2BsOfdmaUplinkChannelEntry 7 }
28
29
30     wmanIf2BsOfdmaPerRngBackoffStart OBJECT-TYPE
31         SYNTAX      INTEGER (0..15)
32         MAX-ACCESS  read-write
33         STATUS      deprecated
34         DESCRIPTION
35             "Initial backoff window size for periodic ranging contention,
36             , expressed as a power of 2. Range: 0..15 (the highest order
37             bits shall be unused and set to 0)."

```

```

1      DESCRIPTION
2          "Indicates the starting number, S, of the group of codes
3          used for this uplink. All the ranging codes used on this
4          uplink will be between S and ((S+N+M+L) mod 256). Where,
5          N: the number of initial-ranging codes
6          M: the number of periodic-ranging codes
7          L: the number of bandwidth-request codes
8          O: the number of handover-ranging codes"
9
10     REFERENCE
11         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
12     DEFVAL      { 0 }
13     ::= { wmanIf2BsOfdmaUplinkChannelEntry 10 }
14
15
16 wmanIf2BsOfdmaPermutationBase OBJECT-TYPE
17     SYNTAX      INTEGER (0..255)
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21         "Determines the UL_PermBase parameter for the subcarrier
22         permutation to be used on this uplink channel.
23         UL_PermBase = 7 LSBs of Permutation base."
24     REFERENCE
25         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
26     DEFVAL      { 0 }
27     ::= { wmanIf2BsOfdmaUplinkChannelEntry 11 }
28
29
30 wmanIf2BsOfdmaULAllocSubchBitmap OBJECT-TYPE
31     SYNTAX      OCTET STRING (SIZE (9))
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "This is a bitmap describing the physical sub-channels
36         allocated to the segment in the UL, when using the uplink
37         PUSC permutation. The LSB of the first byte shall correspond
38         to subchannel 0. For any bit that is not set, the
39         corresponding subchannel shall not be used by the SS on
40         that segment"
41     REFERENCE
42         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
43     ::= { wmanIf2BsOfdmaUplinkChannelEntry 12 }
44
45
46 wmanIf2BsOfdmaOptPermULAllocSubchBitmap OBJECT-TYPE
47     SYNTAX      OCTET STRING (SIZE (13))
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51         "This is a bitmap describing the sub-channels allocated to
52         the segment in the UL, when using the uplink optional PUSC
53         permutation (see 8.4.6.2.5 in IEEE Std 802.16-2004). The
54         LSB of the first byte shall correspond to subchannel 0.
55         For any bit that is not set, the corresponding subchannel
56         shall not be used by the SS on that segment. When this TLV
57         is not present, BS may allocate any subchannels to an SS."
58     REFERENCE
59
60
61
62
63
64
65

```

```

1         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
2         ::= { wmanIf2BsOfdmaUplinkChannelEntry 13 }
3
4
5 wmanIf2BsOfdmaBandAMCAallocThreshold OBJECT-TYPE
6     SYNTAX      INTEGER (0 .. 255)
7     UNITS       "dB"
8     MAX-ACCESS  read-write
9     STATUS      current
10    DESCRIPTION
11        "Threshold of the maximum of the standard deviations of the
12         individual bands CINR measurements over time to trigger
13         mode transition from normal subchannel to Band AMC"
14    REFERENCE
15        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
16    ::= { wmanIf2BsOfdmaUplinkChannelEntry 14 }
17
18
19
20 wmanIf2BsOfdmaBandAMCReleaseThreshold OBJECT-TYPE
21     SYNTAX      INTEGER (0 .. 255)
22     UNITS       "dB"
23     MAX-ACCESS  read-write
24     STATUS      current
25     DESCRIPTION
26        "Threshold of the maximum of the standard deviations of the
27         individual bands CINR measurements over time to trigger
28         mode transition from Band AMC to normal subchannel"
29    REFERENCE
30        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
31    ::= { wmanIf2BsOfdmaUplinkChannelEntry 15 }
32
33
34
35
36 wmanIf2BsOfdmaBandAMCAallocTimer OBJECT-TYPE
37     SYNTAX      INTEGER (0 .. 255)
38     UNITS       "Frame"
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42        "Minimum required number of frames to measure the average
43         and standard deviation for the event of Band AMC triggering"
44    REFERENCE
45        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
46    ::= { wmanIf2BsOfdmaUplinkChannelEntry 16 }
47
48
49
50 wmanIf2BsOfdmaBandAMCReleaseTimer OBJECT-TYPE
51     SYNTAX      INTEGER (0 .. 255)
52     UNITS       "Frame"
53     MAX-ACCESS  read-write
54     STATUS      current
55     DESCRIPTION
56        "Minimum required number of frames to measure the average
57         and standard deviation for the event triggering from Band
58         AMC to normal subchannel"
59    REFERENCE
60        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
61    ::= { wmanIf2BsOfdmaUplinkChannelEntry 17 }
62
63
64
65

```

```

1  wmanIf2BsOfdmaBandStatRepMAXPeriod OBJECT-TYPE
2      SYNTAX      INTEGER (0 .. 255)
3      UNITS       "Frame"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "Maximum period between refreshing the Band CINR
8            measurement by the unsolicited REP-RSP"
9      REFERENCE
10         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
11         ::= { wmanIf2BsOfdmaUplinkChannelEntry 18 }
12
13
14
15  wmanIf2BsOfdmaBandAMCRetryTimer OBJECT-TYPE
16      SYNTAX      INTEGER (0 .. 255)
17      UNITS       "Frame"
18      MAX-ACCESS  read-write
19      STATUS      current
20      DESCRIPTION
21          "Backoff timer between consecutive mode transitions from
22            normal subchannel to Band AMC when the previous request
23            is failed"
24      REFERENCE
25         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
26         ::= { wmanIf2BsOfdmaUplinkChannelEntry 19 }
27
28
29
30
31  wmanIf2BsOfdmaSafetyChAllocThreshold OBJECT-TYPE
32      SYNTAX      INTEGER (0 .. 255)
33      UNITS       "dB"
34      MAX-ACCESS  read-write
35      STATUS      deprecated
36      DESCRIPTION
37          "This object defines the OFDMA safety channel allocation
38            threshold."
39      REFERENCE
40         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
41         ::= { wmanIf2BsOfdmaUplinkChannelEntry 20 }
42
43
44
45  wmanIf2BsOfdmaSafetyChReleaseThreshold OBJECT-TYPE
46      SYNTAX      INTEGER (0 .. 255)
47      UNITS       "dB"
48      MAX-ACCESS  read-write
49      STATUS      deprecated
50      DESCRIPTION
51          "This object defines the OFDMA safety channel release
52            threshold."
53      REFERENCE
54         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
55         ::= { wmanIf2BsOfdmaUplinkChannelEntry 21 }
56
57
58
59
60  wmanIf2BsOfdmaSafetyChAllocTimer OBJECT-TYPE
61      SYNTAX      INTEGER (0 .. 255)
62      UNITS       "Frame"
63      MAX-ACCESS  read-write
64      STATUS      deprecated
65

```

```

1      DESCRIPTION
2          "This object defines the OFDMA safety channel allocation
3          timer."
4
5      REFERENCE
6          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
7      ::= { wmanIf2BsOfdmaUplinkChannelEntry 22 }
8
9
10     wmanIf2BsOfdmaSafetyChReleaseTimer OBJECT-TYPE
11     SYNTAX      INTEGER (0 .. 255)
12     UNITS       "Frame"
13     MAX-ACCESS  read-write
14     STATUS      deprecated
15     DESCRIPTION
16         "This object defines the OFDMA safety channel release
17         timer."
18     REFERENCE
19         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
20     ::= { wmanIf2BsOfdmaUplinkChannelEntry 23 }
21
22
23
24     wmanIf2BsOfdmaBinStatRepMAXPeriod OBJECT-TYPE
25     SYNTAX      INTEGER (0 .. 255)
26     UNITS       "Frame"
27     MAX-ACCESS  read-write
28     STATUS      deprecated
29     DESCRIPTION
30         "This object defines the OFDMA bin status reporting
31         maximum period."
32     REFERENCE
33         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
34     ::= { wmanIf2BsOfdmaUplinkChannelEntry 24 }
35
36
37
38     wmanIf2BsOfdmaSafetyChaRetryTimer OBJECT-TYPE
39     SYNTAX      INTEGER (0 .. 255)
40     UNITS       "Frame"
41     MAX-ACCESS  read-write
42     STATUS      deprecated
43     DESCRIPTION
44         "This object defines the OFDMA safety channel retry
45         timer."
46     REFERENCE
47         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
48     ::= { wmanIf2BsOfdmaUplinkChannelEntry 25 }
49
50
51
52     wmanIf2BsOfdmaHARQAackDelayULBurst OBJECT-TYPE
53     SYNTAX      INTEGER { oneframeoffset(1),
54                      twoframesoffset(2),
55                      threeframesoffset(3) }
56     MAX-ACCESS  read-write
57     STATUS      deprecated
58     DESCRIPTION
59         "This object defines the OFDMA H-ARQ ACK delay for UL burst.
60         1 = one frame offset
61         2 = two frames offset
62         3 = three frames offset"
63
64
65

```

```

1      REFERENCE
2          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
3          ::= { wmanIf2BsOfdmaUplinkChannelEntry 26 }
4
5
6      wmanIf2BsOfdmaCQICHBandAMCTranaDelay OBJECT-TYPE
7          SYNTAX      INTEGER (0 .. 255)
8          UNITS       "Frame"
9          MAX-ACCESS  read-write
10         STATUS      deprecated
11         DESCRIPTION
12             "This object defines the OFDMA CQICH band AMC transition
13             delay."
14         REFERENCE
15             "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
16             ::= { wmanIf2BsOfdmaUplinkChannelEntry 27 }
17
18
19
20     wmanIf2BsOfdmaDownlinkChannelTable OBJECT-TYPE
21         SYNTAX      SEQUENCE OF WmanIf2BsOfdmaDownlinkChannelEntry
22         MAX-ACCESS  not-accessible
23         STATUS      current
24         DESCRIPTION
25             "This table contains DCD channel attributes, defining the
26             transmission characteristics of downlink channels"
27         REFERENCE
28             "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
29             ::= { wmanIf2BsOfdmaPhy 2 }
30
31
32
33     wmanIf2BsOfdmaDownlinkChannelEntry OBJECT-TYPE
34         SYNTAX      WmanIf2BsOfdmaDownlinkChannelEntry
35         MAX-ACCESS  not-accessible
36         STATUS      current
37         DESCRIPTION
38             "This table provides one row for each downlink channel of
39             multi-sector BS, and is indexed by BS ifIndex. An entry in
40             this table exists for each ifEntry of BS with an ifType of
41             ieee80216WMAN."
42         INDEX       { ifIndex }
43         ::= { wmanIf2BsOfdmaDownlinkChannelTable 1 }
44
45
46
47     WmanIf2BsOfdmaDownlinkChannelEntry ::= SEQUENCE {
48         wmanIf2BsOfdmaBsEIRP                INTEGER,
49         wmanIf2BsOfdmaChannelNumber         WmanIf2ChannelNumber,
50         wmanIf2BsOfdmaTTG                   INTEGER,
51         wmanIf2BsOfdmaRTG                   INTEGER,
52         wmanIf2BsOfdmaInitRngMaxRSS         INTEGER,
53         wmanIf2BsOfdmaDownlinkCenterFreq   Unsigned32,
54         wmanIf2BsOfdmaBsId                  WmanIf2BsIdType,
55         wmanIf2BsOfdmaMacVersion            WmanIf2MacVersion,
56         wmanIf2BsOfdmaFrameDurationCode    INTEGER,
57         wmanIf2BsOfdmaSizeCqichIdField     INTEGER,
58         wmanIf2BsOfdmaHARQAackDelayBurst   INTEGER}
59
60
61
62
63     wmanIf2BsOfdmaBsEIRP OBJECT-TYPE
64         SYNTAX      INTEGER (-32768..32767)
65

```



```

1      UNITS      "dBm"
2      MAX-ACCESS read-write
3      STATUS     current
4      DESCRIPTION
5          "The EIRP is the equivalent isotropic radiated power of
6          the base station, which is computed for a simple
7          single-antenna transmitter."
8      REFERENCE
9          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
10     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 1 }
11
12 wmanIf2BsOfdmaChannelNumber OBJECT-TYPE
13     SYNTAX      WmanIf2ChannelNumber
14     MAX-ACCESS read-write
15     STATUS     current
16     DESCRIPTION
17         "Downlink channel number as defined in 8.5. Used for
18         license-exempt operation only."
19     REFERENCE
20         "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
21     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 2 }
22
23 wmanIf2BsOfdmaTTG OBJECT-TYPE
24     SYNTAX      INTEGER (0..255)
25     UNITS      "PS"
26     MAX-ACCESS read-write
27     STATUS     current
28     DESCRIPTION
29         "Transmit / Receive Transition Gap. Used on TDD system only."
30     REFERENCE
31         "Subclause 11.4.1, Table 358, in IEEE Std 802.16e-2005"
32     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 3 }
33
34 wmanIf2BsOfdmaRTG OBJECT-TYPE
35     SYNTAX      INTEGER (0..255)
36     UNITS      "PS"
37     MAX-ACCESS read-write
38     STATUS     current
39     DESCRIPTION
40         "Receive / Transmit Transition Gap. Used on TDD system only."
41     REFERENCE
42         "Subclause 11.4.1, Table 358, in IEEE Std 802.16e-2005"
43     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 4 }
44
45 wmanIf2BsOfdmaInitRngMaxRSS OBJECT-TYPE
46     SYNTAX      INTEGER (-32768..32767)
47     UNITS      "dBm"
48     MAX-ACCESS read-write
49     STATUS     current
50     DESCRIPTION
51         "Initial Ranging Max. equivalent isotropic received power
52         at BS Signed in units of 1 dBm."
53     REFERENCE
54         "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"

```

```

1      ::= { wmanIf2BsOfdmaDownlinkChannelEntry 5 }
2
3
4  wmanIf2BsOfdmaDownlinkCenterFreq OBJECT-TYPE
5      SYNTAX      Unsigned32
6      UNITS       "kHz"
7      MAX-ACCESS  read-write
8      STATUS      current
9      DESCRIPTION
10         "Downlink center frequency (kHz)."
```

REFERENCE

```

11         "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
12
13     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 6 }
14
15
16  wmanIf2BsOfdmaBsId OBJECT-TYPE
17      SYNTAX      WmanIf2BsIdType
18      MAX-ACCESS  read-write
19      STATUS      current
20      DESCRIPTION
21         "Base station ID."
```

REFERENCE

```

22         "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
23
24     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 7 }
25
26
27  wmanIf2BsOfdmaMacVersion OBJECT-TYPE
28      SYNTAX      WmanIf2MacVersion
29      MAX-ACCESS  read-write
30      STATUS      current
31      DESCRIPTION
32         "This parameter specifies the version of 802.16 to which
33         the message originator conforms."
```

REFERENCE

```

34         "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
35
36     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 8 }
37
38
39  wmanIf2BsOfdmaFrameDurationCode OBJECT-TYPE
40      SYNTAX      INTEGER { aASGap(0),
41                          duration2ms(1),
42                          duration2dot5ms(2),
43                          duration4ms(3),
44                          duration5ms(4),
45                          duration8ms(5),
46                          duration10ms(6),
47                          duration12dot5ms(7),
48                          duration20ms(8) }
49      MAX-ACCESS  read-write
50      STATUS      current
51      DESCRIPTION
52         "The duration of the frame. The frame duration code values
53         are specified in Table 274."
```

REFERENCE

```

54         "Table 273, in IEEE Std 802.16-2004"
55
56     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 9 }
57
58
59  wmanIf2BsOfdmaSizeCqichIdField OBJECT-TYPE
60
61
62
63
64
65
```

```

1      SYNTAX      INTEGER {threebits(1),
2                      fourbits(2),
3                      fivebits(3),
4                      sixbits(4),
5                      sevenbits(5),
6                      eightbits(6),
7                      ninebits(7)}
8
9      MAX-ACCESS  read-write
10     STATUS      deprecated
11     DESCRIPTION
12
13         "This object defines the size of CQICH ID field.
14         0 = Reserved
15         1 = 3 bits
16         2 = 4 bits
17         3 = 5 bits
18         4 = 6 bits
19         5 = 7 bits
20         6 = 8 bits
21         7 = 9 bits
22         8...255 = Reserved"
23
24     REFERENCE
25
26         "Subclause 11.3.1, Table 358, in IEEE Std 802.16-2004"
27     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 10 }
28
29
30 wmanIf2BsOfdmaHARQAackDelayBurst OBJECT-TYPE
31     SYNTAX      INTEGER {oneframeoffset(1),
32                      twoframesoffset(2),
33                      threeframesoffset(3)}
34
35     MAX-ACCESS  read-write
36     STATUS      deprecated
37     DESCRIPTION
38
39         "This object defines the OFDMA H-ARQ ACK delay for DL burst.
40         1 = one frame offset
41         2 = two frames offset
42         3 = three frames offset"
43
44     REFERENCE
45
46         "Subclause 11.3.1, Table 358, in IEEE Std 802.16-2004"
47     ::= { wmanIf2BsOfdmaDownlinkChannelEntry 11 }
48
49 wmanIf2BsOfdmaUcdBurstProfileTable OBJECT-TYPE
50     SYNTAX      SEQUENCE OF WmanIf2BsOfdmaUcdBurstProfileEntry
51     MAX-ACCESS  not-accessible
52     STATUS      current
53     DESCRIPTION
54
55         "This table contains UCD burst profiles for each uplink
56         channel"
57
58     REFERENCE
59
60         "Table 356, in IEEE Std 802.16-2004"
61     ::= { wmanIf2BsOfdmaPhy 3 }
62
63 wmanIf2BsOfdmaUcdBurstProfileEntry OBJECT-TYPE
64     SYNTAX      WmanIf2BsOfdmaUcdBurstProfileEntry
65     MAX-ACCESS  not-accessible
66     STATUS      current

```

```

1      DESCRIPTION
2          "This table provides one row for each UCD burst profile.
3          This table is double indexed. The primary index is an
4          ifIndex with an ifType of ieee80216WMAN. The secondary
5          index is wmanIf2BsOfdmaUiucIndex."
6      INDEX          { ifIndex, wmanIf2BsOfdmaUiucIndex }
7      ::= { wmanIf2BsOfdmaUcdBurstProfileTable 1 }
8
9
10
11      WmanIf2BsOfdmaUcdBurstProfileEntry ::= SEQUENCE {
12          wmanIf2BsOfdmaUiucIndex          INTEGER,
13          wmanIf2BsOfdmaUcdFecCodeType     WmanIf2OfdmaUcdFecCode,
14          wmanIf2BsOfdmaRangingDataRatio   INTEGER,
15          wmanIf2BsOfdmaNorCOverNOVERRIDE  OCTET STRING,
16          wmanIf2BsOfdmaUcdBurstProfileRowStatus RowStatus}
17
18
19      wmanIf2BsOfdmaUiucIndex OBJECT-TYPE
20          SYNTAX          INTEGER (1 .. 10)
21          MAX-ACCESS      not-accessible
22          STATUS          current
23          DESCRIPTION
24              "The Uplink Interval Usage Code indicates the uplink burst
25              profile in the UCD message, and is used along with ifIndex
26              to identify an entry in the
27              wmanIf2BsOfdmaUcdBurstProfileTable."
28          REFERENCE
29              "Subclause 8.4.5.4.1, in IEEE Std 802.16-2004"
30          ::= { wmanIf2BsOfdmaUcdBurstProfileEntry 1 }
31
32
33
34
35      wmanIf2BsOfdmaUcdFecCodeType OBJECT-TYPE
36          SYNTAX          WmanIf2OfdmaUcdFecCode
37          MAX-ACCESS      read-create
38          STATUS          current
39          DESCRIPTION
40              "Uplink FEC code type and modulation type"
41          REFERENCE
42              "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
43          ::= { wmanIf2BsOfdmaUcdBurstProfileEntry 2 }
44
45
46
47      wmanIf2BsOfdmaRangingDataRatio OBJECT-TYPE
48          SYNTAX          INTEGER (-128 .. 127)
49          UNITS           "dB"
50          MAX-ACCESS      read-create
51          STATUS          current
52          DESCRIPTION
53              "Reducing factor in units of 1 dB, between the power used
54              for this burst and power should be used for CDMA Ranging."
55          REFERENCE
56              "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
57          ::= { wmanIf2BsOfdmaUcdBurstProfileEntry 3 }
58
59
60
61      wmanIf2BsOfdmaNorCOverNOVERRIDE OBJECT-TYPE
62          SYNTAX          OCTET STRING (SIZE (5))
63          MAX-ACCESS      read-create
64          STATUS          deprecated
65

```

```

1      DESCRIPTION
2          "This is a list of numbers, where each number is encoded by
3          one nibble, and interpreted as a signed integer. The nibbles
4          correspond in order to the list define by Table 334 in IEEE
5          Std 802.16-2004 starting from the second line, such that
6          the LS nibble of the first byte corresponds to the second
7          line in the table. The number encoded by each nibble
8          represents the difference in normalized C/N relative to the
9          previous line in the table"
10
11     REFERENCE
12         "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
13     ::= { wmanIf2BsOfdmaUcdBurstProfileEntry 4 }
14
15
16 wmanIf2BsOfdmaUcdBurstProfileRowStatus OBJECT-TYPE
17     SYNTAX      RowStatus
18     MAX-ACCESS  read-create
19     STATUS      current
20     DESCRIPTION
21         "This object is used to create a new row or modify or delete
22         an existing row in this table. If the implementator of this
23         MIB has chosen not to implement 'dynamic assignment' of
24         profiles, this object is not useful and should return
25         noSuchName upon SNMP request."
26     ::= { wmanIf2BsOfdmaUcdBurstProfileEntry 5 }
27
28
29
30
31 wmanIf2BsOfdmaDcdBurstProfileTable OBJECT-TYPE
32     SYNTAX      SEQUENCE OF WmanIf2BsOfdmaDcdBurstProfileEntry
33     MAX-ACCESS  not-accessible
34     STATUS      current
35     DESCRIPTION
36         "This table provides one row for each DCD burst profile.
37         This table is double indexed. The primary index is an
38         ifIndex with an ifType of ieee80216WMAN. The secondary
39         index is wmanIf2BsOfdmaDiucIndex."
40     ::= { wmanIf2BsOfdmaPhy 4 }
41
42
43
44 wmanIf2BsOfdmaDcdBurstProfileEntry OBJECT-TYPE
45     SYNTAX      WmanIf2BsOfdmaDcdBurstProfileEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "This table provides one row for each DCD burst profile.
50         This table is double indexed. The primary index is an
51         ifIndex with an ifType of ieee80216WMAN. The secondary
52         index is wmanIf2BsOfdmaDiucIndex."
53     INDEX      { ifIndex, wmanIf2BsOfdmaDiucIndex }
54     ::= { wmanIf2BsOfdmaDcdBurstProfileTable 1 }
55
56
57
58 WmanIf2BsOfdmaDcdBurstProfileEntry ::= SEQUENCE {
59     wmanIf2BsOfdmaDiucIndex          INTEGER,
60     wmanIf2BsOfdmaDownlinkFrequency Unsigned32,
61     wmanIf2BsOfdmaDcdFecCodeType    WmanIf2OfdmaDcdFecCode,
62     wmanIf2BsOfdmaDiucMandatoryExitThresh INTEGER,
63     wmanIf2BsOfdmaDiucMinEntryThresh INTEGER,
64
65

```

```

1         wmanIf2BsOfdmaDcdBurstProfileRowStatus RowStatus}
2
3
4 wmanIf2BsOfdmaDiucIndex OBJECT-TYPE
5     SYNTAX      INTEGER (0 .. 12)
6     MAX-ACCESS  not-accessible
7     STATUS      current
8     DESCRIPTION
9         "The Downlink Interval Usage Code indicates the downlink
10        burst profile in the DCD message, and is used along with
11        ifIndex to identify an entry in the
12        wmanIf2BsOfdmaDcdBurstProfileTable."
13
14     REFERENCE
15         "Subclause 8.4.5.3.1, in IEEE Std 802.16-2004"
16     ::= { wmanIf2BsOfdmaDcdBurstProfileEntry 1 }
17
18
19 wmanIf2BsOfdmaDownlinkFrequency OBJECT-TYPE
20     SYNTAX      Unsigned32
21     UNITS       "kHz"
22     MAX-ACCESS  read-create
23     STATUS      current
24     DESCRIPTION
25         "Downlink Frequency (kHz)."

```

```

1         "DIUC minimum entry threshold: 0 - 63.75 dB The minimum
2         CINR required to start using this DIUC when changing from
3         a more robust DIUC is required, in 0.25 dB units."
4
5     REFERENCE
6         "Subclause 11.4.2, Table 363, in IEEE Std 802.16-2004"
7     ::= { wmanIf2BsOfdmaDcdBurstProfileEntry 5 }
8
9
10    wmanIf2BsOfdmaDcdBurstProfileRowStatus OBJECT-TYPE
11        SYNTAX      RowStatus
12        MAX-ACCESS  read-create
13        STATUS      current
14        DESCRIPTION
15            "This object is used to create a new row or modify or delete
16            an existing row in this table. If the implementator of this
17            MIB has chosen not to implement 'dynamic assignment' of
18            profiles, this object is not useful and should return
19            noSuchName upon SNMP request."
20        ::= { wmanIf2BsOfdmaDcdBurstProfileEntry 6 }
21
22
23
24    wmanIf2BsSsOfdmaReqCapabilitiesTable OBJECT-TYPE
25        SYNTAX      SEQUENCE OF WmanIf2BsSsOfdmaReqCapabilitiesEntry
26        MAX-ACCESS  not-accessible
27        STATUS      current
28        DESCRIPTION
29            "This table contains the basic capability information,
30            specific to OFDMA Phy, of MSs that have been reported by
31            MSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages.
32            Entries in this table should be created when an MS
33            registers with a BS."
34        ::= { wmanIf2BsOfdmaPhy 5 }
35
36
37
38    wmanIf2BsSsOfdmaReqCapabilitiesEntry OBJECT-TYPE
39        SYNTAX      WmanIf2BsSsOfdmaReqCapabilitiesEntry
40        MAX-ACCESS  not-accessible
41        STATUS      current
42        DESCRIPTION
43            "This table provides one row for each MS that has been
44            registered in the BS. This table augments the table
45            wmanIf2BsRegisteredSsTable."
46        AUGMENTS { wmanIf2BsRegisteredSsEntry }
47        ::= { wmanIf2BsSsOfdmaReqCapabilitiesTable 1 }
48
49
50
51    WmanIf2BsSsOfdmaReqCapabilitiesEntry ::= SEQUENCE {
52        wmanIf2BsSsOfdmaReqCapFftSizes          WmanIf2OfdmaFftSizes,
53        wmanIf2BsSsOfdmaReqCapDemodulator      WmanIf2OfdmaMsDeModType,
54        wmanIf2BsSsOfdmaReqCapModulator        WmanIf2OfdmaMsModType,
55        wmanIf2BsSsOfdmaReqCapNoHarqChannel    Unsigned32,
56        wmanIf2BsSsOfdmaReqCapPermutation      WmanIf2OfdmaPermutation,
57        wmanIf2BsSsOfdmaReqCapDemMimo          WmanIf2OfdmaDemMimo,
58        wmanIf2BsSsOfdmaReqCapMimoCapability   WmanIf2OfdmaMimoCap,
59        wmanIf2BsSsOfdmaReqCapUlMimo           WmanIf2OfdmaUlMimo,
60        wmanIf2BsSsOfdmaReqCapPrivateMap       WmanIf2OfdmaPrivMap,
61        wmanIf2BsSsOfdmaReqCapAasCapability    WmanIf2OfdmaAasCap,
62        wmanIf2BsSsOfdmaReqCapCinrMeasurement WmanIf2OfdmaCinrCap,
63
64
65

```

```

1      wmanIf2BsSsOfdmaReqCapUlPowerControl      WmanIf2OfdmaUlPower,
2      wmanIf2BsSsOfdmaReqCapMapCapability      WmanIf2OfdmaMapCap,
3      wmanIf2BsSsOfdmaReqCapUlControlChannel  WmanIf2OfdmaUlCntlCh,
4      wmanIf2BsSsOfdmaReqCapCistCapability    WmanIf2OfdmaMsCistCap,
5      wmanIf2BsSsOfdmaReqCapMaxHarqBurst     WmanIf2OfdmaMaxHarq,
6      wmanIf2BsSsOfdmaReqCapModMimo         WmanIf2OfdmaModMimo,
7      wmanIf2BsSsOfdmaReqCapSdmaPilot       WmanIf2SdmaPilotCap,
8      wmanIf2BsSsOfdmaReqCapMultipleBurst   WmanIf2MultiBurst,
9      wmanIf2BsSsOfdmaReqCapIncrHarqBuffer   WmanIf2IncrHarqBuf,
10     wmanIf2BsSsOfdmaReqCapChaseHarqBuffer  WmanIf2ChaseHarqBuf}
11
12
13
14     wmanIf2BsSsOfdmaReqCapFftSizes OBJECT-TYPE
15         SYNTAX      WmanIf2OfdmaFftSizes
16         MAX-ACCESS  read-only
17         STATUS      current
18         DESCRIPTION
19             "This field indicates the FFT sizes supported by MS."
20         REFERENCE
21             "Subclause 11.8.3.7.1 in IEEE 802.16e-2005"
22         ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 1 }
23
24
25
26     wmanIf2BsSsOfdmaReqCapDemodulator OBJECT-TYPE
27         SYNTAX      WmanIf2OfdmaMsDeModType
28         MAX-ACCESS  read-only
29         STATUS      current
30         DESCRIPTION
31             "This field indicates the different demodulator options
32             supported by MS for downlink."
33         REFERENCE
34             "Subclause 11.8.3.7.2 in IEEE Std 802.16e-2005"
35         ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 2 }
36
37
38
39     wmanIf2BsSsOfdmaReqCapModulator OBJECT-TYPE
40         SYNTAX      WmanIf2OfdmaMsModType
41         MAX-ACCESS  read-only
42         STATUS      current
43         DESCRIPTION
44             "This field indicates the different modulator options
45             supported by MS for uplink."
46         REFERENCE
47             "Subclause 11.8.3.7.3 in IEEE Std 802.16e-2005"
48         ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 3 }
49
50
51
52     wmanIf2BsSsOfdmaReqCapNoHarqChannel OBJECT-TYPE
53         SYNTAX      Unsigned32
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "This field specifies the number of uplink H-ARQ
58             channels (n) the SS supports, where n = 1..16.
59             The value of this object should be 0..15."
60         ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 4 }
61
62
63
64     wmanIf2BsSsOfdmaReqCapPermutation OBJECT-TYPE
65

```



```

1      SYNTAX      WmanIf2OfdmaPermutation
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "This field indicates the OFDMA MS Permutation support."
6      REFERENCE
7          "Subclause 11.8.3.7.4 in IEEE 802.16e"
8      ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 5 }
9
10
11
12     wmanIf2BsSsOfdmaReqCapDemMimo OBJECT-TYPE
13     SYNTAX      WmanIf2OfdmaDemMimo
14     MAX-ACCESS  read-only
15     STATUS      current
16     DESCRIPTION
17         "This field indicates the different MIMO options supported
18         by a WirelessMAN-OFDMA PHY SS in the downlink."
19     REFERENCE
20         "Subclause 11.8.3.7.5 in IEEE 802.16e"
21     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 6 }
22
23
24
25     wmanIf2BsSsOfdmaReqCapMimoCapability OBJECT-TYPE
26     SYNTAX      WmanIf2OfdmaMimoCap
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "This field indicates the MIMO capability of OFDMA MS
31         demodulator."
32     REFERENCE
33         "Subclause 11.8.3.7.5 in IEEE 802.16e"
34     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 7 }
35
36
37
38     wmanIf2BsSsOfdmaReqCapUlMimo OBJECT-TYPE
39     SYNTAX      WmanIf2OfdmaUlMimo
40     MAX-ACCESS  read-only
41     STATUS      current
42     DESCRIPTION
43         "This field indicates different MIMO options supported
44         by a OFDMA PHY SS in the uplink"
45     REFERENCE
46         "Subclause 11.8.3.7.6 in IEEE 802.16e"
47     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 8 }
48
49
50
51     wmanIf2BsSsOfdmaReqCapPrivateMap OBJECT-TYPE
52     SYNTAX      WmanIf2OfdmaPrivMap
53     MAX-ACCESS  read-only
54     STATUS      current
55     DESCRIPTION
56         "This field indicates AAS private map parameters
57         supported by a OFDMA SS"
58     REFERENCE
59         "Subclause 11.8.3.7.7 in IEEE 802.16e"
60     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 9 }
61
62
63
64     wmanIf2BsSsOfdmaReqCapAasCapability OBJECT-TYPE
65

```

```

1      SYNTAX      WmanIf2OfdmaAasCap
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "This field indicates different AAS options
6          supported by a OFDMA PHY SS in the downlink"
7      REFERENCE
8          "Subclause 11.8.3.7.8 in IEEE 802.16e"
9      ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 10 }
10
11
12
13 wmanIf2BsSsOfdmaReqCapCinrMeasurement OBJECT-TYPE
14     SYNTAX      WmanIf2OfdmaCinrCap
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "This field indicates the CINR measurement capability
19         supported by a OFDMA PHY SS in the downlink."
20     REFERENCE
21         "Subclause 11.8.3.7.9 in IEEE 802.16e"
22     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 11 }
23
24
25
26 wmanIf2BsSsOfdmaReqCapUlPowerControl OBJECT-TYPE
27     SYNTAX      WmanIf2OfdmaUlPower
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31         "This field indicates the power control options
32         supported by a OFDMA PHY SS for uplink transmission."
33     REFERENCE
34         "Subclause 11.8.3.7.11 in IEEE 802.16e"
35     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 12 }
36
37
38
39 wmanIf2BsSsOfdmaReqCapMapCapability OBJECT-TYPE
40     SYNTAX      WmanIf2OfdmaMapCap
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44         "This field indicates the different MAP options supported
45         by a OFDMA PHY SS"
46     REFERENCE
47         "Subclause 11.8.3.7.11 in IEEE 802.16e"
48     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 13 }
49
50
51
52 wmanIf2BsSsOfdmaReqCapUlControlChannel OBJECT-TYPE
53     SYNTAX      WmanIf2OfdmaUlCntlCh
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57         "This field indicates the different uplink control channels
58         supported by a OFDMA PHY SS."
59     REFERENCE
60         "Subclause 11.8.3.7.13 in IEEE 802.16e"
61     ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 14 }
62
63
64
65

```

```

1  wmanIf2BsSsOfdmaReqCapCistCapability OBJECT-TYPE
2      SYNTAX      WmanIf2OfdmaMsCistCap
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "This field indicates the MS capability of supporting CSIT
7          (uplink sounding)."

```

```

1
2
3 wmanIf2BsSsOfdmaReqCapIncrHarqBuffer OBJECT-TYPE
4     SYNTAX      WmanIf2IncrHarqBuf
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "This field indicates the maximal number of data
9         bits the SS is able to use for buffering for NEP/NSCH
10        based incremental redundancy CTC in downlink and uplink
11        transmissions."
12
13    REFERENCE
14        "Subclause 11.8.3.7.19 in IEEE 802.16e"
15    ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 20 }
16
17
18 wmanIf2BsSsOfdmaReqCapChaseHarqBuffer OBJECT-TYPE
19     SYNTAX      WmanIf2ChaseHarqBuf
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "This field indicates the maximal number of data
24         bits the SS is able to use for buffering for
25         DIUC/duration based HARQ methods (Chase combining and
26         CC-IR) in downlink and uplink transmissions."
27
28    REFERENCE
29        "Subclause 11.8.3.7.19 in IEEE 802.16e"
30    ::= { wmanIf2BsSsOfdmaReqCapabilitiesEntry 21 }
31
32
33 wmanIf2BsSsOfdmaRspCapabilitiesTable OBJECT-TYPE
34     SYNTAX      SEQUENCE OF WmanIf2BsSsOfdmaRspCapabilitiesEntry
35     MAX-ACCESS  not-accessible
36     STATUS      current
37     DESCRIPTION
38         "This table contains the basic capability information,
39         specific to OFDMA Phy, of MSs that have been reported by
40         MSs to BS using RNG-REQ, SBC-REQ and REG-REQ messages.
41         Entries in this table should be created when an MS
42         registers with a BS."
43
44    ::= { wmanIf2BsOfdmaPhy 6 }
45
46
47
48 wmanIf2BsSsOfdmaRspCapabilitiesEntry OBJECT-TYPE
49     SYNTAX      WmanIf2BsSsOfdmaRspCapabilitiesEntry
50     MAX-ACCESS  not-accessible
51     STATUS      current
52     DESCRIPTION
53         "This table provides one row for each MS that has been
54         registered in the BS. This table augments the table
55         wmanIf2BsRegisteredSsTable."
56
57    AUGMENTS { wmanIf2BsRegisteredSsEntry }
58    ::= { wmanIf2BsSsOfdmaRspCapabilitiesTable 1 }
59
60
61 WmanIf2BsSsOfdmaRspCapabilitiesEntry ::= SEQUENCE {
62     wmanIf2BsSsOfdmaRspCapFftSizes      WmanIf2OfdmaFftSizes,
63     wmanIf2BsSsOfdmaRspCapDemodulator   WmanIf2OfdmaMsDeModType,
64     wmanIf2BsSsOfdmaRspCapModulator     WmanIf2OfdmaMsModType,
65

```

```

1      wmanIf2BsSsOfdmaRspCapNoHarqChannel      Unsigned32,
2      wmanIf2BsSsOfdmaRspCapPermutation      WmanIf2OfdmaPermutation,
3      wmanIf2BsSsOfdmaRspCapDemMimo          WmanIf2OfdmaDemMimo,
4      wmanIf2BsSsOfdmaRspCapMimoCapability   WmanIf2OfdmaMimoCap,
5      wmanIf2BsSsOfdmaRspCapUlMimo          WmanIf2OfdmaUlMimo,
6      wmanIf2BsSsOfdmaRspCapPrivateMap      WmanIf2OfdmaPrivMap,
7      wmanIf2BsSsOfdmaRspCapAasCapability    WmanIf2OfdmaAasCap,
8      wmanIf2BsSsOfdmaRspCapCinrMeasurement WmanIf2OfdmaCinrCap,
9      wmanIf2BsSsOfdmaRspCapUlPowerControl  WmanIf2OfdmaUlPower,
10     wmanIf2BsSsOfdmaRspCapMapCapability    WmanIf2OfdmaMapCap,
11     wmanIf2BsSsOfdmaRspCapUlControlChannel WmanIf2OfdmaUlCntlCh,
12     wmanIf2BsSsOfdmaRspCapCistCapability   WmanIf2OfdmaMsCistCap,
13     wmanIf2BsSsOfdmaRspCapMaxHarqBurst    WmanIf2OfdmaMaxHarq,
14     wmanIf2BsSsOfdmaRspCapModMimo         WmanIf2OfdmaModMimo,
15     wmanIf2BsSsOfdmaRspCapSdmaPilot       WmanIf2SdmaPilotCap,
16     wmanIf2BsSsOfdmaRspCapMultipleBurst   WmanIf2MultiBurst,
17     wmanIf2BsSsOfdmaRspCapIncrHarqBuffer  WmanIf2IncrHarqBuf,
18     wmanIf2BsSsOfdmaRspCapChaseHarqBuffer WmanIf2ChaseHarqBuf }
19
20
21
22
23
24     wmanIf2BsSsOfdmaRspCapFftSizes OBJECT-TYPE
25         SYNTAX          WmanIf2OfdmaFftSizes
26         MAX-ACCESS     read-only
27         STATUS          current
28         DESCRIPTION
29             "This field indicates the FFT sizes negotiated with the
30             MS."
31         REFERENCE
32             "Subclause 11.8.3.7.1 in IEEE 802.16e-2005"
33         ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 1 }
34
35
36
37     wmanIf2BsSsOfdmaRspCapDemodulator OBJECT-TYPE
38         SYNTAX          WmanIf2OfdmaMsDeModType
39         MAX-ACCESS     read-only
40         STATUS          current
41         DESCRIPTION
42             "This field indicates the different demodulator options
43             negotiated for MS for downlink."
44         REFERENCE
45             "Subclause 11.8.3.7.2 in IEEE Std 802.16e-2005"
46         ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 2 }
47
48
49
50     wmanIf2BsSsOfdmaRspCapModulator OBJECT-TYPE
51         SYNTAX          WmanIf2OfdmaMsModType
52         MAX-ACCESS     read-only
53         STATUS          current
54         DESCRIPTION
55             "This field indicates the different modulator options
56             negotiated for MS for uplink."
57         REFERENCE
58             "Subclause 11.8.3.7.3 in IEEE Std 802.16e-2005"
59         ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 3 }
60
61
62
63     wmanIf2BsSsOfdmaRspCapNoHarqChannel OBJECT-TYPE
64         SYNTAX          Unsigned32
65

```

```

1      MAX-ACCESS    read-only
2      STATUS       current
3      DESCRIPTION
4          "This field specifies the number of uplink H-ARQ
5          channels (n) the SS supports, where n = 1..16.
6          The value of this object should be 0..15."
7      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 4 }
8
9
10     wmanIf2BsSsOfdmaRspCapPermutation OBJECT-TYPE
11     SYNTAX       WmanIf2OfdmaPermutation
12     MAX-ACCESS   read-only
13     STATUS       current
14     DESCRIPTION
15         "This field indicates the OFDMA MS Permutation support
16         negotiated for MS."
17     REFERENCE
18         "Subclause 11.8.3.7.4 in IEEE 802.16e"
19     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 5 }
20
21
22     wmanIf2BsSsOfdmaRspCapDemMimo OBJECT-TYPE
23     SYNTAX       WmanIf2OfdmaDemMimo
24     MAX-ACCESS   read-only
25     STATUS       current
26     DESCRIPTION
27         "This field indicates the different MIMO options supported
28         by a WirelessMAN-OFDMA PHY SS in the downlink."
29     REFERENCE
30         "Subclause 11.8.3.7.5 in IEEE 802.16e"
31     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 6 }
32
33
34     wmanIf2BsSsOfdmaRspCapMimoCapability OBJECT-TYPE
35     SYNTAX       WmanIf2OfdmaMimoCap
36     MAX-ACCESS   read-only
37     STATUS       current
38     DESCRIPTION
39         "This field indicates the MIMO capability of OFDMA MS
40         demodulator."
41     REFERENCE
42         "Subclause 11.8.3.7.5 in IEEE 802.16e"
43     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 7 }
44
45
46     wmanIf2BsSsOfdmaRspCapUlMimo OBJECT-TYPE
47     SYNTAX       WmanIf2OfdmaUlMimo
48     MAX-ACCESS   read-only
49     STATUS       current
50     DESCRIPTION
51         "This field indicates different MIMO options supported
52         by a OFDMA PHY SS in the uplink"
53     REFERENCE
54         "Subclause 11.8.3.7.6 in IEEE 802.16e"
55     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 8 }
56
57
58     wmanIf2BsSsOfdmaRspCapPrivateMap OBJECT-TYPE
59     SYNTAX       WmanIf2OfdmaPrivMap
60
61
62
63
64
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This field indicates AAS private map parameters
5          supported by a OFDMA SS"
6      REFERENCE
7          "Subclause 11.8.3.7.7 in IEEE 802.16e"
8      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 9 }
9
10
11
12  wmanIf2BsSsOfdmaRspCapAasCapability OBJECT-TYPE
13      SYNTAX      WmanIf2OfdmaAasCap
14      MAX-ACCESS  read-only
15      STATUS      current
16      DESCRIPTION
17          "This field indicates different AAS options
18          supported by a OFDMA PHY SS in the downlink"
19      REFERENCE
20          "Subclause 11.8.3.7.8 in IEEE 802.16e"
21      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 10 }
22
23
24
25  wmanIf2BsSsOfdmaRspCapCinrMesurement OBJECT-TYPE
26      SYNTAX      WmanIf2OfdmaCinrCap
27      MAX-ACCESS  read-only
28      STATUS      current
29      DESCRIPTION
30          "This field indicates the CINR measurement capability
31          supported by a OFDMA PHY SS in the downlink."
32      REFERENCE
33          "Subclause 11.8.3.7.9 in IEEE 802.16e"
34      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 11 }
35
36
37
38  wmanIf2BsSsOfdmaRspCapUlPowerControl OBJECT-TYPE
39      SYNTAX      WmanIf2OfdmaUlPower
40      MAX-ACCESS  read-only
41      STATUS      current
42      DESCRIPTION
43          "This field indicates the power control options
44          supported by a OFDMA PHY SS for uplink transmission."
45      REFERENCE
46          "Subclause 11.8.3.7.11 in IEEE 802.16e"
47      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 12 }
48
49
50
51  wmanIf2BsSsOfdmaRspCapMapCapability OBJECT-TYPE
52      SYNTAX      WmanIf2OfdmaMapCap
53      MAX-ACCESS  read-only
54      STATUS      current
55      DESCRIPTION
56          "This field indicates the different MAP options supported
57          by a OFDMA PHY SS"
58      REFERENCE
59          "Subclause 11.8.3.7.11 in IEEE 802.16e"
60      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 13 }
61
62
63
64  wmanIf2BsSsOfdmaRspCapUlControlChannel OBJECT-TYPE
65

```

```

1      SYNTAX      WmanIf2OfdmaUlCntlCh
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "This field indicates the different uplink control channels
6          supported by a OFDMA PHY SS."
7      REFERENCE
8          "Subclause 11.8.3.7.13 in IEEE 802.16e"
9      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 14 }
10
11
12
13  wmanIf2BsSsOfdmaRspCapCistCapability OBJECT-TYPE
14      SYNTAX      WmanIf2OfdmaMsCistCap
15      MAX-ACCESS  read-only
16      STATUS      current
17      DESCRIPTION
18          "This field indicates the MS capability of supporting CSIT
19          (uplink sounding)."
20      REFERENCE
21          "Subclause 11.8.3.7.14 in IEEE 802.16e"
22      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 15 }
23
24
25
26  wmanIf2BsSsOfdmaRspCapMaxHarqBurst OBJECT-TYPE
27      SYNTAX      WmanIf2OfdmaMaxHarq
28      MAX-ACCESS  read-only
29      STATUS      current
30      DESCRIPTION
31          "This field indicates the maximum number of UL/DL HARQ
32          burst allocations for the SS in a single UL/DL subframe."
33      REFERENCE
34          "Subclause 11.8.3.7.15 in IEEE 802.16e"
35      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 16 }
36
37
38
39  wmanIf2BsSsOfdmaRspCapModMimo OBJECT-TYPE
40      SYNTAX      WmanIf2OfdmaModMimo
41      MAX-ACCESS  read-only
42      STATUS      current
43      DESCRIPTION
44          "This field indicates the MIMO capability of OFDMA SS
45          modulator."
46      REFERENCE
47          "Subclause 11.8.3.7.16 in IEEE 802.16e"
48      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 17 }
49
50
51
52  wmanIf2BsSsOfdmaRspCapSdmaPilot OBJECT-TYPE
53      SYNTAX      WmanIf2SdmaPilotCap
54      MAX-ACCESS  read-only
55      STATUS      current
56      DESCRIPTION
57          "This field indicates the SDMA pilot pattern support
58          for AMC zone."
59      REFERENCE
60          "Subclause 11.8.3.7.17 in IEEE 802.16e"
61      ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 18 }
62
63
64
65

```



```

1  wmanIf2BsSsOfdmaRspCapMultipleBurst OBJECT-TYPE
2      SYNTAX      WmanIf2MultiBurst
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "This field indicates whether multiple FEC types are
7              supported in DL/UL burst profiles."
8      REFERENCE
9          "Subclause 11.8.3.7.18 in IEEE 802.16e"
10     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 19 }
11
12  wmanIf2BsSsOfdmaRspCapIncrHarqBuffer OBJECT-TYPE
13      SYNTAX      WmanIf2IncrHarqBuf
14      MAX-ACCESS  read-only
15      STATUS      current
16      DESCRIPTION
17          "This field indicates the maximal number of data
18              bits the SS is able to use for buffering for NEP/NSCH
19              based incremental redundancy CTC in downlink and uplink
20              transmissions."
21      REFERENCE
22          "Subclause 11.8.3.7.19 in IEEE 802.16e"
23     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 20 }
24
25  wmanIf2BsSsOfdmaRspCapChaseHarqBuffer OBJECT-TYPE
26      SYNTAX      WmanIf2ChaseHarqBuf
27      MAX-ACCESS  read-only
28      STATUS      current
29      DESCRIPTION
30          "This field indicates the maximal number of data
31              bits the SS is able to use for buffering for
32              DIUC/duration based HARQ methods (Chase combining and
33              CC-IR) in downlink and uplink transmissions."
34      REFERENCE
35          "Subclause 11.8.3.7.19 in IEEE 802.16e"
36     ::= { wmanIf2BsSsOfdmaRspCapabilitiesEntry 21 }
37
38  wmanIf2BsOfdmaCapabilitiesTable OBJECT-TYPE
39      SYNTAX      SEQUENCE OF WmanIf2BsOfdmaCapabilitiesEntry
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "This table contains the basic capabilities, specific to
44              OFDMA Phy, of the BS as implemented in BS hardware and
45              software. These capabilities along with the configuration
46              for them (wmanIf2BsOfdmaCapabilitiesConfigTable) are used
47              for negotiation of basic capabilities with SS using
48              RNG-RSP, SBC-RSP and REG-RSP messages. The negotiated
49              capabilities are obtained by interSubclause of MS raw
50              reported capabilities, BS raw capabilities and BS
51              configured capabilities. The objects in the table have
52              read-only access. The table is maintained by BS."
53     ::= { wmanIf2BsOfdmaPhy 7 }
54
55
56
57
58
59
60
61
62
63
64
65

```

```

1  wmanIf2BsOfdmaCapabilitiesEntry OBJECT-TYPE
2      SYNTAX          WmanIf2BsOfdmaCapabilitiesEntry
3      MAX-ACCESS     not-accessible
4      STATUS         current
5      DESCRIPTION
6          "This table provides one row for each BS sector and is
7              indexed by ifIndex."
8      INDEX { ifIndex }
9      ::= { wmanIf2BsOfdmaCapabilitiesTable 1 }
10
11
12
13  WmanIf2BsOfdmaCapabilitiesEntry ::= SEQUENCE {
14      wmanIf2BsOfdmaCapFftSizes          WmanIf2OfdmaFftSizes,
15      wmanIf2BsOfdmaCapDemodulator       WmanIf2OfdmaMsDeModType,
16      wmanIf2BsOfdmaCapModulator         WmanIf2OfdmaMsModType,
17      wmanIf2BsOfdmaCapNoHarqChannel     Unsigned32,
18      wmanIf2BsOfdmaCapPermutation       WmanIf2OfdmaPermutation,
19      wmanIf2BsSsOfdmaCapDemMimo         WmanIf2OfdmaDemMimo,
20      wmanIf2BsSsOfdmaCapMimoCapability  WmanIf2OfdmaMimoCap,
21      wmanIf2BsSsOfdmaCapUlMimo         WmanIf2OfdmaUlMimo,
22      wmanIf2BsSsOfdmaCapPrivateMap     WmanIf2OfdmaPrivMap,
23      wmanIf2BsSsOfdmaCapAasCapability   WmanIf2OfdmaAasCap,
24      wmanIf2BsSsOfdmaCapCinrMeasurement WmanIf2OfdmaCinrCap,
25      wmanIf2BsSsOfdmaCapUlPowerControl WmanIf2OfdmaUlPower,
26      wmanIf2BsSsOfdmaCapMapCapability   WmanIf2OfdmaMapCap,
27      wmanIf2BsSsOfdmaCapUlControlChannel WmanIf2OfdmaUlCntlCh,
28      wmanIf2BsSsOfdmaCapCistCapability  WmanIf2OfdmaMsCistCap,
29      wmanIf2BsSsOfdmaCapMaxHarqBurst    WmanIf2OfdmaMaxHarq,
30      wmanIf2BsSsOfdmaCapModMimo         WmanIf2OfdmaModMimo,
31      wmanIf2BsSsOfdmaCapSdmaPilot       WmanIf2SdmaPilotCap,
32      wmanIf2BsSsOfdmaCapMultipleBurst   WmanIf2MultiBurst,
33      wmanIf2BsSsOfdmaCapIncrHarqBuffer  WmanIf2IncrHarqBuf,
34      wmanIf2BsSsOfdmaCapChaseHarqBuffer WmanIf2ChaseHarqBuf }
35
36
37
38
39
40
41  wmanIf2BsOfdmaCapFftSizes OBJECT-TYPE
42      SYNTAX          WmanIf2OfdmaFftSizes
43      MAX-ACCESS     read-only
44      STATUS         current
45      DESCRIPTION
46          "This field indicates the FFT sizes supported by BS."
47      REFERENCE
48          "Subclause 11.8.3.7.1 in IEEE 802.16e-2005"
49      ::= { wmanIf2BsOfdmaCapabilitiesEntry 1 }
50
51
52
53  wmanIf2BsOfdmaCapDemodulator OBJECT-TYPE
54      SYNTAX          WmanIf2OfdmaMsDeModType
55      MAX-ACCESS     read-only
56      STATUS         current
57      DESCRIPTION
58          "This field indicates the different demodulator options
59              supported by BS."
60      REFERENCE
61          "Subclause 11.8.3.7.2 in IEEE Std 802.16e-2005"
62      ::= { wmanIf2BsOfdmaCapabilitiesEntry 2 }
63
64
65

```

```

1  wmanIf2BsOfdmaCapModulator OBJECT-TYPE
2      SYNTAX      WmanIf2OfdmaMsModType
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "This field indicates the different modulator options
7              supported by BS."
8      REFERENCE
9          "Subclause 11.8.3.7.3 in IEEE Std 802.16e-2005"
10     ::= { wmanIf2BsOfdmaCapabilitiesEntry 3 }
11
12
13
14  wmanIf2BsOfdmaCapNoHarqChannel OBJECT-TYPE
15      SYNTAX      Unsigned32
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19          "This field specifies the number of uplink H-ARQ
20              channels (n) the SS supports, where n = 1..16.
21              The value of this object should be 0..15."
22     ::= { wmanIf2BsOfdmaCapabilitiesEntry 4 }
23
24
25
26  wmanIf2BsOfdmaCapPermutation OBJECT-TYPE
27      SYNTAX      WmanIf2OfdmaPermutation
28      MAX-ACCESS  read-only
29      STATUS      current
30      DESCRIPTION
31          "This field indicates the OFDMA MS Permutation support
32              supported by BS."
33      REFERENCE
34          "Subclause 11.8.3.7.4 in IEEE 802.16e"
35     ::= { wmanIf2BsOfdmaCapabilitiesEntry 5 }
36
37
38
39  wmanIf2BsSsOfdmaCapDemMimo OBJECT-TYPE
40      SYNTAX      WmanIf2OfdmaDemMimo
41      MAX-ACCESS  read-only
42      STATUS      current
43      DESCRIPTION
44          "This field indicates the different MIMO options supported
45              by a WirelessMAN-OFDMA PHY SS in the downlink."
46      REFERENCE
47          "Subclause 11.8.3.7.5 in IEEE 802.16e"
48     ::= { wmanIf2BsOfdmaCapabilitiesEntry 6 }
49
50
51
52  wmanIf2BsSsOfdmaCapMimoCapability OBJECT-TYPE
53      SYNTAX      WmanIf2OfdmaMimoCap
54      MAX-ACCESS  read-only
55      STATUS      current
56      DESCRIPTION
57          "This field indicates the MIMO capability of OFDMA MS
58              demodulator."
59      REFERENCE
60          "Subclause 11.8.3.7.5 in IEEE 802.16e"
61     ::= { wmanIf2BsOfdmaCapabilitiesEntry 7 }
62
63
64
65

```

```

1  wmanIf2BsSsOfdmaCapUlMimo OBJECT-TYPE
2      SYNTAX      WmanIf2OfdmaUlMimo
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "This field indicates different MIMO options supported
7          by a OFDMA PHY SS in the uplink"
8      REFERENCE
9          "Subclause 11.8.3.7.6 in IEEE 802.16e"
10     ::= { wmanIf2BsOfdmaCapabilitiesEntry 8 }
11
12
13
14  wmanIf2BsSsOfdmaCapPrivateMap OBJECT-TYPE
15      SYNTAX      WmanIf2OfdmaPrivMap
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19          "This field indicates AAS private map parameters
20          supported by a OFDMA SS"
21      REFERENCE
22          "Subclause 11.8.3.7.7 in IEEE 802.16e"
23     ::= { wmanIf2BsOfdmaCapabilitiesEntry 9 }
24
25
26
27  wmanIf2BsSsOfdmaCapAasCapability OBJECT-TYPE
28      SYNTAX      WmanIf2OfdmaAasCap
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "This field indicates different AAS options
33          supported by a OFDMA PHY SS in the downlink"
34      REFERENCE
35          "Subclause 11.8.3.7.8 in IEEE 802.16e"
36     ::= { wmanIf2BsOfdmaCapabilitiesEntry 10 }
37
38
39
40  wmanIf2BsSsOfdmaCapCinrMeasurement OBJECT-TYPE
41      SYNTAX      WmanIf2OfdmaCinrCap
42      MAX-ACCESS  read-only
43      STATUS      current
44      DESCRIPTION
45          "This field indicates the CINR measurement capability
46          supported by a OFDMA PHY SS in the downlink."
47      REFERENCE
48          "Subclause 11.8.3.7.9 in IEEE 802.16e"
49     ::= { wmanIf2BsOfdmaCapabilitiesEntry 11 }
50
51
52
53  wmanIf2BsSsOfdmaCapUlPowerControl OBJECT-TYPE
54      SYNTAX      WmanIf2OfdmaUlPower
55      MAX-ACCESS  read-only
56      STATUS      current
57      DESCRIPTION
58          "This field indicates the power control options
59          supported by a OFDMA PHY SS for uplink transmission."
60      REFERENCE
61          "Subclause 11.8.3.7.11 in IEEE 802.16e"
62     ::= { wmanIf2BsOfdmaCapabilitiesEntry 12 }
63
64
65

```

```

1
2
3 wmanIf2BsSsOfdmaCapMapCapability OBJECT-TYPE
4     SYNTAX      WmanIf2OfdmaMapCap
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "This field indicates the different MAP options supported
9         by a OFDMA PHY SS"
10
11    REFERENCE
12        "Subclause 11.8.3.7.11 in IEEE 802.16e"
13    ::= { wmanIf2BsOfdmaCapabilitiesEntry 13 }
14
15 wmanIf2BsSsOfdmaCapUlControlChannel OBJECT-TYPE
16     SYNTAX      WmanIf2OfdmaUlCntlCh
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20         "This field indicates the different uplink control channels
21         supported by a OFDMA PHY SS."
22
23    REFERENCE
24        "Subclause 11.8.3.7.13 in IEEE 802.16e"
25    ::= { wmanIf2BsOfdmaCapabilitiesEntry 14 }
26
27
28
29 wmanIf2BsSsOfdmaCapCistCapability OBJECT-TYPE
30     SYNTAX      WmanIf2OfdmaMsCistCap
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "This field indicates the MS capability of supporting CSIT
35         (uplink sounding)."
36
37    REFERENCE
38        "Subclause 11.8.3.7.14 in IEEE 802.16e"
39    ::= { wmanIf2BsOfdmaCapabilitiesEntry 15 }
40
41
42 wmanIf2BsSsOfdmaCapMaxHarqBurst OBJECT-TYPE
43     SYNTAX      WmanIf2OfdmaMaxHarq
44     MAX-ACCESS  read-only
45     STATUS      current
46     DESCRIPTION
47         "This field indicates the maximum number of UL/DL HARQ
48         burst allocations for the SS in a single UL/DL subframe."
49
50    REFERENCE
51        "Subclause 11.8.3.7.15 in IEEE 802.16e"
52    ::= { wmanIf2BsOfdmaCapabilitiesEntry 16 }
53
54
55 wmanIf2BsSsOfdmaCapModMimo OBJECT-TYPE
56     SYNTAX      WmanIf2OfdmaModMimo
57     MAX-ACCESS  read-only
58     STATUS      current
59     DESCRIPTION
60         "This field indicates the MIMO capability of OFDMA SS
61         modulator."
62
63    REFERENCE
64        "Subclause 11.8.3.7.16 in IEEE 802.16e"
65

```

```

1      ::= { wmanIf2BsOfdmaCapabilitiesEntry 17 }
2
3
4  wmanIf2BsSsOfdmaCapSdmaPilot OBJECT-TYPE
5      SYNTAX      WmanIf2SdmaPilotCap
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "This field indicates the SDMA pilot pattern support
10         for AMC zone."
11
12     REFERENCE
13         "Subclause 11.8.3.7.17 in IEEE 802.16e"
14     ::= { wmanIf2BsOfdmaCapabilitiesEntry 18 }
15
16
17  wmanIf2BsSsOfdmaCapMultipleBurst OBJECT-TYPE
18     SYNTAX      WmanIf2MultiBurst
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "This field indicates whether multiple FEC types are
23         supported in DL/UL burst profiles."
24
25     REFERENCE
26         "Subclause 11.8.3.7.18 in IEEE 802.16e"
27     ::= { wmanIf2BsOfdmaCapabilitiesEntry 19 }
28
29
30  wmanIf2BsSsOfdmaCapIncrHarqBuffer OBJECT-TYPE
31     SYNTAX      WmanIf2IncrHarqBuf
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "This field indicates the maximal number of data
36         bits the SS is able to use for buffering for NEP/NSCH
37         based incremental redundancy CTC in downlink and uplink
38         transmissions."
39
40     REFERENCE
41         "Subclause 11.8.3.7.19 in IEEE 802.16e"
42     ::= { wmanIf2BsOfdmaCapabilitiesEntry 20 }
43
44
45  wmanIf2BsSsOfdmaCapChaseHarqBuffer OBJECT-TYPE
46     SYNTAX      WmanIf2ChaseHarqBuf
47     MAX-ACCESS  read-only
48     STATUS      current
49     DESCRIPTION
50         "This field indicates the maximal number of data
51         bits the SS is able to use for buffering for
52         DIUC/duration based HARQ methods (Chase combining and
53         CC-IR) in downlink and uplink transmissions."
54
55     REFERENCE
56         "Subclause 11.8.3.7.19 in IEEE 802.16e"
57     ::= { wmanIf2BsOfdmaCapabilitiesEntry 21 }
58
59
60
61  wmanIf2BsOfdmaCapabilitiesConfigTable OBJECT-TYPE
62     SYNTAX      SEQUENCE OF WmanIf2BsOfdmaCapabilitiesConfigEntry
63     MAX-ACCESS  not-accessible
64     STATUS      current
65

```

```

1      DESCRIPTION
2          "This table contains the configuration for basic
3          capabilities of BS, specific to OFDMA Phy. The table is
4          intended to be used to restrict the Capabilities
5          implemented by BS, for example in order to comply with
6          local regulatory requirements. The BS should use the
7          configuration along with the implemented Capabilities
8          (wmanIf2BsOfdmaPhyTable) for negotiation of basic
9          capabilities with SS using RNG-RSP, SBC-RSP and REG-RSP
10         messages. The negotiated capabilities are obtained by
11         interSubclause of MS reported capabilities, BS raw
12         capabilities and BS configured capabilities. The objects
13         in the table have read-write access. The rows are created
14         by BS as a copy of wmanIf2BsBasicCapabilitiesTable
15         and can be modified by NMS."
16         ::= { wmanIf2BsOfdmaPhy 8 }
17
18
19
20
21
22     wmanIf2BsOfdmaCapabilitiesConfigEntry OBJECT-TYPE
23         SYNTAX      WmanIf2BsOfdmaCapabilitiesConfigEntry
24         MAX-ACCESS  not-accessible
25         STATUS      current
26         DESCRIPTION
27             "This table provides one row for each BS sector and is
28             indexed by ifIndex."
29         INDEX { ifIndex }
30         ::= { wmanIf2BsOfdmaCapabilitiesConfigTable 1 }
31
32
33
34     WmanIf2BsOfdmaCapabilitiesConfigEntry ::= SEQUENCE {
35         wmanIf2BsOfdmaCapCfgFftSizes          WmanIf2OfdmaFftSizes,
36         wmanIf2BsOfdmaCapCfgDemodulator      WmanIf2OfdmaMsDeModType,
37         wmanIf2BsOfdmaCapCfgModulator        WmanIf2OfdmaMsModType,
38         wmanIf2BsOfdmaCapCfgNoHarqChannel    Unsigned32,
39         wmanIf2BsOfdmaCapCfgPermutation      WmanIf2OfdmaPermutation,
40         wmanIf2BsSsOfdmaCapCfgDemMimo        WmanIf2OfdmaDemMimo,
41         wmanIf2BsSsOfdmaCapCfgMimoCapability WmanIf2OfdmaMimoCap,
42         wmanIf2BsSsOfdmaCapCfgULMimo        WmanIf2OfdmaULMimo,
43         wmanIf2BsSsOfdmaCapCfgPrivateMap     WmanIf2OfdmaPrivMap,
44         wmanIf2BsSsOfdmaCapCfgAasCapability  WmanIf2OfdmaAasCap,
45         wmanIf2BsSsOfdmaCapCfgCinrMeasurement WmanIf2OfdmaCinrCap,
46         wmanIf2BsSsOfdmaCapCfgULPowerControl WmanIf2OfdmaULPower,
47         wmanIf2BsSsOfdmaCapCfgMapCapability  WmanIf2OfdmaMapCap,
48         wmanIf2BsSsOfdmaCapCfgULControlChannel WmanIf2OfdmaULCntlCh,
49         wmanIf2BsSsOfdmaCapCfgCistCapability WmanIf2OfdmaMsCistCap,
50         wmanIf2BsSsOfdmaCapCfgMaxHarqBurst   WmanIf2OfdmaMaxHarq,
51         wmanIf2BsSsOfdmaCapCfgModMimo        WmanIf2OfdmaModMimo,
52         wmanIf2BsSsOfdmaCapCfgSdmaPilot      WmanIf2SdmaPilotCap,
53         wmanIf2BsSsOfdmaCapCfgMultipleBurst  WmanIf2MultiBurst,
54         wmanIf2BsSsOfdmaCapCfgIncrHarqBuffer WmanIf2IncrHarqBuf,
55         wmanIf2BsSsOfdmaCapCfgChaseHarqBuffer WmanIf2ChaseHarqBuf }
56
57
58
59
60
61     wmanIf2BsOfdmaCapCfgFftSizes OBJECT-TYPE
62         SYNTAX      WmanIf2OfdmaFftSizes
63         MAX-ACCESS  read-write
64         STATUS      current
65

```

```

1      DESCRIPTION
2          "This field indicates the FFT sizes configured for the BS."
3      REFERENCE
4          "Subclause 11.8.3.7.1 in IEEE 802.16e-2005"
5          ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 1 }
6
7
8      wmanIf2BsOfdmaCapCfgDemodulator OBJECT-TYPE
9          SYNTAX      WmanIf2OfdmaMsDeModType
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "This field indicates the different demodulator options
14             configured for the BS."
15         REFERENCE
16             "Subclause 11.8.3.7.2 in IEEE Std 802.16e-2005"
17             ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 2 }
18
19
20
21     wmanIf2BsOfdmaCapCfgModulator OBJECT-TYPE
22         SYNTAX      WmanIf2OfdmaMsModType
23         MAX-ACCESS  read-write
24         STATUS      current
25         DESCRIPTION
26             "This field indicates the different modulator options
27             configured for the BS."
28         REFERENCE
29             "Subclause 11.8.3.7.3 in IEEE Std 802.16e-2005"
30             ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 3 }
31
32
33
34
35     wmanIf2BsOfdmaCapCfgNoHarqChannel OBJECT-TYPE
36         SYNTAX      Unsigned32
37         MAX-ACCESS  read-only
38         STATUS      current
39         DESCRIPTION
40             "This field specifies the number of uplink H-ARQ
41             channels (n) the SS supports, where n = 1..16.
42             The value of this object should be 0..15."
43             ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 4 }
44
45
46
47     wmanIf2BsOfdmaCapCfgPermutation OBJECT-TYPE
48         SYNTAX      WmanIf2OfdmaPermutation
49         MAX-ACCESS  read-write
50         STATUS      current
51         DESCRIPTION
52             "This field indicates the OFDMA MS Permutation support
53             configured for the BS."
54         REFERENCE
55             "Subclause 11.8.3.7.4 in IEEE 802.16e"
56             ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 5 }
57
58
59
60     wmanIf2BsSsOfdmaCapCfgDemMimo OBJECT-TYPE
61         SYNTAX      WmanIf2OfdmaDemMimo
62         MAX-ACCESS  read-write
63         STATUS      current
64         DESCRIPTION
65

```



```

1         "This field indicates the different MIMO options supported
2         by a WirelessMAN-OFDMA PHY SS in the downlink."
3     REFERENCE
4         "Subclause 11.8.3.7.5 in IEEE 802.16e"
5     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 6 }
6
7
8     wmanIf2BsSsOfdmaCapCfgMimoCapability OBJECT-TYPE
9     SYNTAX      WmanIf2OfdmaMimoCap
10    MAX-ACCESS  read-write
11    STATUS      current
12    DESCRIPTION
13        "This field indicates the MIMO capability of OFDMA MS
14        demodulator."
15    REFERENCE
16        "Subclause 11.8.3.7.5 in IEEE 802.16e"
17    ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 7 }
18
19
20
21    wmanIf2BsSsOfdmaCapCfgUlMimo OBJECT-TYPE
22    SYNTAX      WmanIf2OfdmaUlMimo
23    MAX-ACCESS  read-write
24    STATUS      current
25    DESCRIPTION
26        "This field indicates different MIMO options supported
27        by a OFDMA PHY SS in the uplink"
28    REFERENCE
29        "Subclause 11.8.3.7.6 in IEEE 802.16e"
30    ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 8 }
31
32
33
34    wmanIf2BsSsOfdmaCapCfgPrivateMap OBJECT-TYPE
35    SYNTAX      WmanIf2OfdmaPrivMap
36    MAX-ACCESS  read-write
37    STATUS      current
38    DESCRIPTION
39        "This field indicates AAS private map parameters
40        supported by a OFDMA SS"
41    REFERENCE
42        "Subclause 11.8.3.7.7 in IEEE 802.16e"
43    ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 9 }
44
45
46
47
48    wmanIf2BsSsOfdmaCapCfgAasCapability OBJECT-TYPE
49    SYNTAX      WmanIf2OfdmaAasCap
50    MAX-ACCESS  read-write
51    STATUS      current
52    DESCRIPTION
53        "This field indicates different AAS options
54        supported by a OFDMA PHY SS in the downlink"
55    REFERENCE
56        "Subclause 11.8.3.7.8 in IEEE 802.16e"
57    ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 10 }
58
59
60
61    wmanIf2BsSsOfdmaCapCfgCinrMeasurement OBJECT-TYPE
62    SYNTAX      WmanIf2OfdmaCinrCap
63    MAX-ACCESS  read-write
64    STATUS      current
65

```

```

1      DESCRIPTION
2          "This field indicates the CINR measurement capability
3          supported by a OFDMA PHY SS in the downlink."
4      REFERENCE
5          "Subclause 11.8.3.7.9 in IEEE 802.16e"
6          ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 11 }
7
8
9      wmanIf2BsSsOfdmaCapCfgUlPowerControl OBJECT-TYPE
10     SYNTAX      WmanIf2OfdmaUlPower
11     MAX-ACCESS  read-write
12     STATUS      current
13     DESCRIPTION
14         "This field indicates the power control options
15         supported by a OFDMA PHY SS for uplink transmission."
16     REFERENCE
17         "Subclause 11.8.3.7.11 in IEEE 802.16e"
18         ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 12 }
19
20
21     wmanIf2BsSsOfdmaCapCfgMapCapability OBJECT-TYPE
22     SYNTAX      WmanIf2OfdmaMapCap
23     MAX-ACCESS  read-write
24     STATUS      current
25     DESCRIPTION
26         "This field indicates the different MAP options supported
27         by a OFDMA PHY SS"
28     REFERENCE
29         "Subclause 11.8.3.7.11 in IEEE 802.16e"
30         ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 13 }
31
32
33     wmanIf2BsSsOfdmaCapCfgUlControlChannel OBJECT-TYPE
34     SYNTAX      WmanIf2OfdmaUlCntlCh
35     MAX-ACCESS  read-write
36     STATUS      current
37     DESCRIPTION
38         "This field indicates the different uplink control channels
39         supported by a OFDMA PHY SS."
40     REFERENCE
41         "Subclause 11.8.3.7.13 in IEEE 802.16e"
42         ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 14 }
43
44
45     wmanIf2BsSsOfdmaCapCfgCistCapability OBJECT-TYPE
46     SYNTAX      WmanIf2OfdmaMsCistCap
47     MAX-ACCESS  read-write
48     STATUS      current
49     DESCRIPTION
50         "This field indicates the MS capability of supporting CSIT
51         (uplink sounding)."
52     REFERENCE
53         "Subclause 11.8.3.7.14 in IEEE 802.16e"
54         ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 15 }
55
56
57     wmanIf2BsSsOfdmaCapCfgMaxHarqBurst OBJECT-TYPE
58     SYNTAX      WmanIf2OfdmaMaxHarq
59     MAX-ACCESS  read-write
60
61
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "This field indicates the maximum number of UL/DL HARQ
4          burst allocations for the SS in a single UL/DL subframe."
5      REFERENCE
6          "Subclause 11.8.3.7.15 in IEEE 802.16e"
7      ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 16 }
8
9
10     wmanIf2BsSsOfdmaCapCfgModMimo OBJECT-TYPE
11     SYNTAX      WmanIf2OfdmaModMimo
12     MAX-ACCESS  read-write
13     STATUS      current
14     DESCRIPTION
15         "This field indicates the MIMO capability of OFDMA SS
16         modulator."
17     REFERENCE
18         "Subclause 11.8.3.7.16 in IEEE 802.16e"
19     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 17 }
20
21
22     wmanIf2BsSsOfdmaCapCfgSdmaPilot OBJECT-TYPE
23     SYNTAX      WmanIf2SdmaPilotCap
24     MAX-ACCESS  read-write
25     STATUS      current
26     DESCRIPTION
27         "This field indicates the SDMA pilot pattern support
28         for AMC zone."
29     REFERENCE
30         "Subclause 11.8.3.7.17 in IEEE 802.16e"
31     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 18 }
32
33
34     wmanIf2BsSsOfdmaCapCfgMultipleBurst OBJECT-TYPE
35     SYNTAX      WmanIf2MultiBurst
36     MAX-ACCESS  read-write
37     STATUS      current
38     DESCRIPTION
39         "This field indicates whether multiple FEC types are
40         supported in DL/UL burst profiles."
41     REFERENCE
42         "Subclause 11.8.3.7.18 in IEEE 802.16e"
43     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 19 }
44
45
46     wmanIf2BsSsOfdmaCapCfgIncrHarqBuffer OBJECT-TYPE
47     SYNTAX      WmanIf2IncrHarqBuf
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51         "This field indicates the maximal number of data
52         bits the SS is able to use for buffering for NEP/NSCH
53         based incremental redundancy CTC in downlink and uplink
54         transmissions."
55     REFERENCE
56         "Subclause 11.8.3.7.19 in IEEE 802.16e"
57     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 20 }
58
59
60
61
62
63
64
65

```

```

1  wmanIf2BsSsOfdmaCapCfgChaseHarqBuffer OBJECT-TYPE
2      SYNTAX      WmanIf2ChaseHarqBuf
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "This field indicates the maximal number of data
7          bits the SS is able to use for buffering for
8          DIUC/duration based HARQ methods (Chase combining and
9          CC-IR) in downlink and uplink transmissions."
10     REFERENCE
11         "Subclause 11.8.3.7.19 in IEEE 802.16e"
12     ::= { wmanIf2BsOfdmaCapabilitiesConfigEntry 21 }
13
14  wmanIf2BsOfdmaExUplinkChannelTable OBJECT-TYPE
15      SYNTAX      SEQUENCE OF WmanIf2BsOfdmaExUplinkChannelEntry
16      MAX-ACCESS  not-accessible
17      STATUS      current
18      DESCRIPTION
19          "This table contains UCD channel attributes, defining the
20          transmission characteristics of uplink channels"
21     REFERENCE
22         "Table 349 and Table 353, in IEEE Std 802.16e-2005"
23     ::= { wmanIf2BsOfdmaPhy 9 }
24
25  wmanIf2BsOfdmaExUplinkChannelEntry OBJECT-TYPE
26      SYNTAX      WmanIf2BsOfdmaExUplinkChannelEntry
27      MAX-ACCESS  not-accessible
28      STATUS      current
29      DESCRIPTION
30          "This table provides one row for each uplink channel of
31          multi-sector BS, and is indexed by BS ifIndex. An entry
32          in this table exists for each ifEntry of BS with an
33          ifType of ieee80216WMAN."
34     AUGMENTS { wmanIf2BsOfdmaUplinkChannelEntry }
35     ::= { wmanIf2BsOfdmaExUplinkChannelTable 1 }
36
37  WmanIf2BsOfdmaExUplinkChannelEntry ::= SEQUENCE {
38      wmanIf2BsOfdmaExHandoverRangingStart    INTEGER,
39      wmanIf2BsOfdmaExHandoverRangingEnd      INTEGER,
40      wmanIf2BsOfdmaExHARQAackDelayDLBurst    WmanIf2HarqAckDelay,
41      wmanIf2BsOfdmaExULAmcAlloPhyBandsBitmap OCTET STRING,
42      wmanIf2BsOfdmaExMaxRetransmission        INTEGER,
43      wmanIf2BsOfdmaExNormalizedCnOverride     OCTET STRING,
44      wmanIf2BsOfdmaExSizeOfCqichId           INTEGER,
45      wmanIf2BsOfdmaExNormalizedCnValue        INTEGER,
46      wmanIf2BsOfdmaExNormalizedCnOverride2    OCTET STRING,
47      wmanIf2BsOfdmaExBandAmcEntryAvgCinr     INTEGER,
48      wmanIf2BsOfdmaExAasPreambleUpperBond    INTEGER,
49      wmanIf2BsOfdmaExAasPreambleLowerBond    INTEGER,
50      wmanIf2BsOfdmaExAasBeamSelectAllowed    WmanIf2AasBeamSel,
51      wmanIf2BsOfdmaExCqichIndicationFlag     OCTET STRING,
52      wmanIf2BsOfdmaExUpPowerAdjStep          Unsigned32,
53      wmanIf2BsOfdmaExDownPowerAdjStep        Unsigned32,
54      wmanIf2BsOfdmaExMinPowerOffsetAdj       INTEGER,
55  }
56
57
58
59
60
61
62
63
64
65

```

```

1      wmanIf2BsOfdmaExMaxPowerOffsetAdj      INTEGER,
2      wmanIf2BsOfdmaExHandoverRngCodes      INTEGER,
3      wmanIf2BsOfdmaExTxPwrRepThreshold      INTEGER,
4      wmanIf2BsOfdmaExTprPower              INTEGER,
5      wmanIf2BsOfdmaExAlphaPavg             INTEGER,
6      wmanIf2BsOfdmaExCqichTxPwrRepThreshold INTEGER,
7      wmanIf2BsOfdmaExCqichTprPower         INTEGER,
8      wmanIf2BsOfdmaExCqichAlphaPavg        INTEGER,
9      wmanIf2BsOfdmaExNormalizedCnChSounding INTEGER,
10     wmanIf2BsOfdmaExInitialRngInterval     INTEGER,
11     wmanIf2BsOfdmaExInitialRngBackoffStart INTEGER,
12     wmanIf2BsOfdmaExInitialRngBackoffEnd   INTEGER,
13     wmanIf2BsOfdmaExBwRequestBackoffStart  INTEGER,
14     wmanIf2BsOfdmaExBwRequestBackoffEnd    INTEGER}
15
16
17
18
19     wmanIf2BsOfdmaExHandoverRangingStart OBJECT-TYPE
20         SYNTAX      INTEGER (0..15)
21         MAX-ACCESS  read-write
22         STATUS      current
23         DESCRIPTION
24             "Initial backoff window size for MS performing initial
25             ranging during handover process, expressed as a power
26             of 2."
27
28         REFERENCE
29             "Table 349, in IEEE Std 802.16e-2005"
30         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 1 }
31
32
33     wmanIf2BsOfdmaExHandoverRangingEnd OBJECT-TYPE
34         SYNTAX      INTEGER (0..15)
35         MAX-ACCESS  read-write
36         STATUS      current
37         DESCRIPTION
38             "Final backoff window size for MS performing initial
39             ranging during handover process, expressed as a power
40             of 2."
41
42         REFERENCE
43             "Table 349, in IEEE Std 802.16e-2005"
44         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 2 }
45
46
47
48     wmanIf2BsOfdmaExHARQAackDelayDLBurst OBJECT-TYPE
49         SYNTAX      WmanIf2HarqAckDelay
50         MAX-ACCESS  read-write
51         STATUS      current
52         DESCRIPTION
53             "This object defines the OFDMA H-ARQ ACK delay for DL
54             burst."
55
56         REFERENCE
57             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
58         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 3 }
59
60
61     wmanIf2BsOfdmaExUlAmcAlloPhyBandsBitmap OBJECT-TYPE
62         SYNTAX      OCTET STRING (SIZE (6))
63         MAX-ACCESS  read-write
64         STATUS      current
65

```

```

1      DESCRIPTION
2          "A bitmap describing the physical bands allocated to the
3          segment in the UL, when using the optional AMC permutation
4          with regular MAPs (see 8.4.6.3). The LSB of the first byte
5          shall correspond to the physical band 0. For any bit that
6          is not set, the corresponding physical bands shall not be
7          used by the SS on that segment. When this TLV is not
8          present, BS may allocate any physical bands to an SS."
9
10     REFERENCE
11         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
12     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 4 }
13
14
15     wmanIf2BsOfdmaExMaxRetransmission OBJECT-TYPE
16     SYNTAX      INTEGER (1..255)
17     MAX-ACCESS  read-write
18     STATUS      current
19     DESCRIPTION
20         "Maximum number of retransmission in UL HARQ."
21     REFERENCE
22         "Table 353, in IEEE Std 802.16e-2005"
23     DEFVAL      { 4 }
24     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 5 }
25
26
27
28     wmanIf2BsOfdmaExNormalizedCnOverride OBJECT-TYPE
29     SYNTAX      OCTET STRING (SIZE (8))
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "This is a list of numbers, where each number is encoded by
34         one nibble, and interpreted as a signed integer. The
35         nibbles correspond in order to the list define by Table
36         334, starting from the second line, such that the LS
37         nibble of the first byte corresponds to the second line in
38         the table. The number encoded by each nibble represents
39         the difference in normalized C/N relative to the previous
40         line in the table."
41     REFERENCE
42         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
43     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 6 }
44
45
46
47
48
49     wmanIf2BsOfdmaExSizeOfCqichId OBJECT-TYPE
50     SYNTAX      INTEGER (0..7)
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "Size of CQICH ID field.
55
56         0 = 0 bits
57         1 = 3 bits
58         2 = 4 bits
59         3 = 5 bits
60         4 = 6 bits
61         5 = 7 bits
62         6 = 8 bits
63         7 = 9 bits"
64
65

```

```

1      REFERENCE
2          "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
3      DEFVAL      { 0 }
4          ::= { wmanIf2BsOfdmaExUplinkChannelEntry 7 }
5
6
7      wmanIf2BsOfdmaExNormalizedCnValue OBJECT-TYPE
8          SYNTAX      INTEGER (-128..128)
9          UNITS      "dB"
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "It shall be interpreted as signed integer in dB. It
14             corresponds to the normalized C/N value in the first line
15             (counting except for header cell of table)"
16         REFERENCE
17             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
18         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 8 }
19
20
21
22
23     wmanIf2BsOfdmaExNormalizedCnOverride2 OBJECT-TYPE
24         SYNTAX      OCTET STRING (SIZE (7))
25         MAX-ACCESS  read-write
26         STATUS      current
27         DESCRIPTION
28             "This is a list of numbers, where each number is encoded
29             by one nibble, and interpreted as a signed integer. The
30             nibbles correspond in order to the list define by Table
31             334, starting from the second line (counting except for
32             the header cell of table), such that the LS nibble of
33             the first byte corresponds to the second line in the
34             table. The number encoded by each nibble represents the
35             difference in normalized C/N relative to the previous
36             line in the table."
37         REFERENCE
38             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
39         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 9 }
40
41
42
43
44     wmanIf2BsOfdmaExBandAmcEntryAvgCinr OBJECT-TYPE
45         SYNTAX      INTEGER (-128..128)
46         UNITS      "dB"
47         MAX-ACCESS  read-write
48         STATUS      current
49         DESCRIPTION
50             "Threshold of the average CINR of the whole bandwidth to
51             trigger mode transition from normal subchannel to AMC"
52         REFERENCE
53             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
54         ::= { wmanIf2BsOfdmaExUplinkChannelEntry 10 }
55
56
57
58
59     wmanIf2BsOfdmaExAasPreambleUpperBond OBJECT-TYPE
60         SYNTAX      INTEGER (-128..128)
61         UNITS      "0.25 dB"
62         MAX-ACCESS  read-write
63         STATUS      current
64         DESCRIPTION
65

```

```

1         "Upper bound of AAS preamble."
2     REFERENCE
3         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
4     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 11 }
5
6
7     wmanIf2BsOfdmaExAasPreambleLowerBond OBJECT-TYPE
8         SYNTAX      INTEGER (-128..128)
9         UNITS       "0.25 dB"
10        MAX-ACCESS  read-write
11        STATUS      current
12        DESCRIPTION
13            "Lower bound of AAS preamble."
14        REFERENCE
15            "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
16        ::= { wmanIf2BsOfdmaExUplinkChannelEntry 12 }
17
18
19
20    wmanIf2BsOfdmaExAasBeamSelectAllowed OBJECT-TYPE
21        SYNTAX      WmanIf2AasBeamSel
22        UNITS       "0.25 dB"
23        MAX-ACCESS  read-write
24        STATUS      current
25        DESCRIPTION
26            "Indicate whether unsolicited AAS Beam Select messages
27            (see 6.3.2.3.41 in IEEE 802.16e-2005) should be sent by
28            the MS."
29        REFERENCE
30            "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
31        DEFVAL     { allowed }
32        ::= { wmanIf2BsOfdmaExUplinkChannelEntry 13 }
33
34
35
36
37    wmanIf2BsOfdmaExCqichIndicationFlag OBJECT-TYPE
38        SYNTAX      OCTET STRING (SIZE (1))
39        MAX-ACCESS  read-write
40        STATUS      current
41        DESCRIPTION
42            "The N MSB values of this field represents the N-bit
43            payload value on the Fast-Feedback channel reserved as
44            indication flag for MS to initiate feedback on the
45            Feedback header, where N is the number of payload bits
46            used for S/N measurement feedback on the Fast-Feedback
47            channel. The value shall not be set to all zeros."
48        REFERENCE
49            "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
50        ::= { wmanIf2BsOfdmaExUplinkChannelEntry 14 }
51
52
53
54
55    wmanIf2BsOfdmaExUpPowerAdjStep OBJECT-TYPE
56        SYNTAX      Unsigned32
57        UNITS       "0.01 dB"
58        MAX-ACCESS  read-write
59        STATUS      current
60        DESCRIPTION
61            "MS-specific up power offset adjustment step"
62        REFERENCE
63            "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
64
65

```



```

1      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 15 }
2
3
4      wmanIf2BsOfdmaExDownPowerAdjStep OBJECT-TYPE
5          SYNTAX      Unsigned32
6          UNITS       "0.01 dB"
7          MAX-ACCESS  read-write
8          STATUS      current
9          DESCRIPTION
10             "MS-specific down power offset adjustment step"
11          REFERENCE
12             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
13      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 16 }
14
15
16      wmanIf2BsOfdmaExMinPowerOffsetAdj OBJECT-TYPE
17          SYNTAX      INTEGER
18          UNITS       "0.1 dB"
19          MAX-ACCESS  read-write
20          STATUS      current
21          DESCRIPTION
22             "Minimum level of power offset adjustment"
23          REFERENCE
24             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
25      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 17 }
26
27
28      wmanIf2BsOfdmaExMaxPowerOffsetAdj OBJECT-TYPE
29          SYNTAX      INTEGER
30          UNITS       "0.1 dB"
31          MAX-ACCESS  read-write
32          STATUS      current
33          DESCRIPTION
34             "Minimum level of power offset adjustment"
35          REFERENCE
36             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
37      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 18 }
38
39
40      wmanIf2BsOfdmaExHandoverRngCodes OBJECT-TYPE
41          SYNTAX      INTEGER (0..255)
42          MAX-ACCESS  read-write
43          STATUS      current
44          DESCRIPTION
45             "Number of handover ranging CDMA codes"
46          REFERENCE
47             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
48      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 19 }
49
50
51      wmanIf2BsOfdmaExInitialRngInterval OBJECT-TYPE
52          SYNTAX      INTEGER
53          MAX-ACCESS  read-write
54          STATUS      current
55          DESCRIPTION
56             "Number of frames between initial ranging interval
57              allocation."
58          REFERENCE
59             "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
60
61
62
63
64
65

```

```

1      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 20 }
2
3
4  wmanIf2BsOfdmaExTxPwrRepThreshold OBJECT-TYPE
5      SYNTAX      INTEGER (0..15)
6      UNITS       "dB"
7      MAX-ACCESS  read-write
8      STATUS      current
9      DESCRIPTION
10         "Tx power report threshold.
11         wmanIf2BsOfdmaExTxPwrRepThreshold = 0b1111 means infinite."
12
13      REFERENCE
14         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
15         Std 802.16e-2005"
16
17      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 21 }
18
19  wmanIf2BsOfdmaExTprPower OBJECT-TYPE
20      SYNTAX      INTEGER (0..15)
21      UNITS       "dB"
22      MAX-ACCESS  read-write
23      STATUS      current
24      DESCRIPTION
25         "Tx power report interval = 2 ^ wmanIf2BsOfdmaExTprPower.
26         The unit of Tx power report interval is frame.
27         wmanIf2BsOfdmaExTprPower = 0b1111 means infinite."
28
29      REFERENCE
30         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
31         Std 802.16e-2005"
32
33      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 22 }
34
35
36  wmanIf2BsOfdmaExAlphaPavg OBJECT-TYPE
37      SYNTAX      INTEGER (0..15)
38      UNITS       "dB"
39      MAX-ACCESS  read-write
40      STATUS      current
41      DESCRIPTION
42         "Alpha p_avg parameter as shown in equation 138d in
43         IEEE 802.16e-2005 indicates the multiple of 1/16. For
44         example '0' means 1/16, 15 means 16/16. "
45
46      REFERENCE
47         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
48         Std 802.16e-2005"
49
50      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 23 }
51
52
53  wmanIf2BsOfdmaExCqichTxPwrRepThreshold OBJECT-TYPE
54      SYNTAX      INTEGER (0..15)
55      UNITS       "dB"
56      MAX-ACCESS  read-write
57      STATUS      current
58      DESCRIPTION
59         "Tx power report threshold.
60         wmanIf2BsOfdmaExTxPwrRepThreshold = 0b1111 means infinite.
61         It shall be used when CQICH is allocated to the SS."
62
63      REFERENCE
64         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
65

```

```

1      Std 802.16e-2005"
2      ::= { wmanIf2BsOfdmaExUplinkChannelEntry 24 }
3
4
5  wmanIf2BsOfdmaExCqichTprPower OBJECT-TYPE
6      SYNTAX      INTEGER (0..15)
7      UNITS       "dB"
8      MAX-ACCESS  read-write
9      STATUS      current
10     DESCRIPTION
11         "Tx power report interval = 2 ^ wmanIf2BsOfdmaExTprPower.
12         The unit of Tx power report interval is frame.
13         wmanIf2BsOfdmaExTprPower = 0b1111 means infinite.
14         It shall be used when CQICH is allocated to the SS."
15     REFERENCE
16         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
17         Std 802.16e-2005"
18     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 25 }
19
20
21
22
23  wmanIf2BsOfdmaExCqichAlphaPavg OBJECT-TYPE
24     SYNTAX      INTEGER (0..15)
25     UNITS       "dB"
26     MAX-ACCESS  read-write
27     STATUS      current
28     DESCRIPTION
29         "Alpha p_avg parameter as shown in equation 138d in
30         IEEE 802.16e-2005 indicates the multiple of 1/16. For
31         example '0' means 1/16, 15 means 16/16. It shall be
32         used when CQICH is allocated to the SS."
33     REFERENCE
34         "Subclause 11.3.1, Table 353, and 8.4.10.3.2.1 in IEEE
35         Std 802.16e-2005"
36     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 26 }
37
38
39
40
41  wmanIf2BsOfdmaExNormalizedCnChSounding OBJECT-TYPE
42     SYNTAX      INTEGER
43     MAX-ACCESS  read-write
44     STATUS      current
45     DESCRIPTION
46         "Signed integer for the required C/N (dB) for Channel
47         Sounding."
48     REFERENCE
49         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
50     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 27 }
51
52
53
54
55  wmanIf2BsOfdmaExInitialRngBackoffStart OBJECT-TYPE
56     SYNTAX      INTEGER (0..15)
57     MAX-ACCESS  read-write
58     STATUS      current
59     DESCRIPTION
60         "Initial backoff window size for initial ranging
61         contention, expressed as a power of 2."
62     REFERENCE
63         "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
64     ::= { wmanIf2BsOfdmaExUplinkChannelEntry 28 }
65

```

```

1
2
3 wmanIf2BsOfdmaExInitialRngBackoffEnd OBJECT-TYPE
4     SYNTAX      INTEGER (0..15)
5     MAX-ACCESS  read-write
6     STATUS      current
7     DESCRIPTION
8         "Final backoff window size for initial ranging
9         contention, expressed as a power of 2."
10
11    REFERENCE
12        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
13    ::= { wmanIf2BsOfdmaExUplinkChannelEntry 29 }
14
15 wmanIf2BsOfdmaExBwRequestBackoffStart OBJECT-TYPE
16     SYNTAX      INTEGER (0..15)
17     MAX-ACCESS  read-write
18     STATUS      current
19     DESCRIPTION
20
21        "Initial backoff window size for contention BW requests,
22        expressed as a power of 2."
23
24    REFERENCE
25        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
26    ::= { wmanIf2BsOfdmaExUplinkChannelEntry 30 }
27
28
29 wmanIf2BsOfdmaExBwRequestBackoffEnd OBJECT-TYPE
30     SYNTAX      INTEGER (0..15)
31     MAX-ACCESS  read-write
32     STATUS      current
33     DESCRIPTION
34
35        "Final backoff window size for contention BW requests,
36        expressed as a power of 2."
37
38    REFERENCE
39        "Subclause 11.3.1, Table 353, in IEEE Std 802.16e-2005"
40    ::= { wmanIf2BsOfdmaExUplinkChannelEntry 31 }
41
42 wmanIf2BsOfdmaExDownlinkChannelTable OBJECT-TYPE
43     SYNTAX      SEQUENCE OF WmanIf2BsOfdmaExDownlinkChannelEntry
44     MAX-ACCESS  not-accessible
45     STATUS      current
46     DESCRIPTION
47
48        "This table contains DCD channel attributes, defining the
49        transmission characteristics of uplink channels"
50
51    REFERENCE
52        "Table 358, in IEEE Std 802.16e-2005"
53    ::= { wmanIf2BsOfdmaPhy 10 }
54
55 wmanIf2BsOfdmaExDownlinkChannelEntry OBJECT-TYPE
56     SYNTAX      WmanIf2BsOfdmaExDownlinkChannelEntry
57     MAX-ACCESS  not-accessible
58     STATUS      current
59     DESCRIPTION
60
61        "This table provides one row for each downlink channel of
62        multi-sector BS, and is indexed by BS ifIndex. An entry
63        in this table exists for each ifEntry of BS with an
64        ifType of ieee80216WMAN."
65

```

```

1      AUGMENTS { wmanIf2BsOfdmaDownlinkChannelEntry }
2      ::= { wmanIf2BsOfdmaExDownlinkChannelTable 1 }
3
4
5      WmanIf2BsOfdmaExDownlinkChannelEntry ::= SEQUENCE {
6          wmanIf2BsOfdmaExHARQAackDelayULBurst      WmanIf2HarqAckDelay,
7          wmanIf2BsOfdmaExHarqZonePermutation      WmanIfPermutationType,
8          wmanIf2BsOfdmaExHMaxRetransmission      INTEGER,
9          wmanIf2BsOfdmaExCinrAlphaAvg            INTEGER,
10         wmanIf2BsOfdmaExRssiAlphaAvg            INTEGER,
11         wmanIf2BsOfdmaExDlAmcAlloPhyBandsBitmap OCTET STRING,
12         wmanIf2BsOfdmaExHandoverSupported      WmanIf2HoSupportType,
13         wmanIf2BsOfdmaExThresholdAddBsDivSet   INTEGER,
14         wmanIf2BsOfdmaExThresholdDelBsDivSet   INTEGER,
15         wmanIf2BsOfdmaExAsrSlotLength          INTEGER,
16         wmanIf2BsOfdmaExAsrSwitchingPeriod     INTEGER,
17         wmanIf2BsOfdmaExHytseresisMargin       INTEGER,
18         wmanIf2BsOfdmaExTimeToTrigger          INTEGER,
19         wmanIf2BsOfdmaExRetartCount            INTEGER}
20
21
22
23
24      wmanIf2BsOfdmaExHARQAackDelayULBurst OBJECT-TYPE
25          SYNTAX      WmanIf2HarqAckDelay
26          MAX-ACCESS  read-write
27          STATUS      current
28          DESCRIPTION
29              "This object defines the OFDMA H-ARQ ACK delay for UL
30              burst."
31          REFERENCE
32              "Table 358, in IEEE Std 802.16e-2005"
33              ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 1 }
34
35
36
37      wmanIf2BsOfdmaExHarqZonePermutation OBJECT-TYPE
38          SYNTAX      WmanIfPermutationType
39          MAX-ACCESS  read-write
40          STATUS      current
41          DESCRIPTION
42              "Permutation type for broadcast region in HARQ zone"
43          REFERENCE
44              "Table 358, in IEEE Std 802.16e-2005"
45              ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 2 }
46
47
48
49      wmanIf2BsOfdmaExHMaxRetransmission OBJECT-TYPE
50          SYNTAX      INTEGER (0..255)
51          MAX-ACCESS  read-write
52          STATUS      current
53          DESCRIPTION
54              "Maximum number of retransmission in DL HARQ."
55          REFERENCE
56              "Table 358, in IEEE Std 802.16e-2005"
57          DEFVAL     { 4 }
58          ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 3 }
59
60
61
62      wmanIf2BsOfdmaExCinrAlphaAvg OBJECT-TYPE
63          SYNTAX      INTEGER (0..15)
64          MAX-ACCESS  read-write
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "Bit 0..3 of Default RSSI and CINR averaging parameter
4          TLV.
5
6          Default averaging parameter Alpha Avg for physical
7          CINR measurements, in multiples of 1/16. For example
8          '0' means 1/16, 15 means 16/16."
9
10     REFERENCE
11         "Table 358, in IEEE Std 802.16e-2005"
12     DEFVAL      { 3 }
13     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 4 }
14
15
16     wmanIf2BsOfdmaExRssiAlphaAvg OBJECT-TYPE
17     SYNTAX      INTEGER (0..15)
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21         "Bit 0..3 of Default RSSI and CINR averaging parameter
22         TLV.
23
24         Default averaging parameter Alpha Avg for physical
25         RSSI measurements, in multiples of 1/16. For example
26         '0' means 1/16, 15 means 16/16."
27
28     REFERENCE
29         "Table 358, in IEEE Std 802.16e-2005"
30     DEFVAL      { 3 }
31     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 5 }
32
33
34     wmanIf2BsOfdmaExDlAmcAlloPhyBandsBitmap OBJECT-TYPE
35     SYNTAX      OCTET STRING (SIZE (6))
36     MAX-ACCESS  read-write
37     STATUS      current
38     DESCRIPTION
39         "A bitmap describing the physical bands allocated to the
40         segment in the DL, when allocating AMC subchannels
41         through the HARQ MAP, or through the Normal MAP, or for
42         Band-AMC CINR reports, or using the optional AMC
43         permutation (see 8.4.6.3). The LSB of the first byte
44         shall correspond to band 0. For any bit that is not set,
45         the corresponding band shall not be used by the SS on
46         that segment. When this TLV is not present, BS may
47         allocate any physical bands to an SS."
48
49     REFERENCE
50         "Table 358, in IEEE Std 802.16e-2005"
51     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 6 }
52
53
54     wmanIf2BsOfdmaExHandoverSupported OBJECT-TYPE
55     SYNTAX      WmanIf2HoSupportType
56     MAX-ACCESS  read-write
57     STATUS      current
58     DESCRIPTION
59         "Indicates the types of handover supported.
60         Bit #0 = HO
61
62
63
64
65

```

```

1         Bit #1 = MDHO
2         Bit #2 = FBSS HO."
3     REFERENCE
4         "Table 358, in IEEE Std 802.16e-2005"
5     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 7 }
6
7
8     wmanIf2BsOfdmaExThresholdAddBsDivSet OBJECT-TYPE
9     SYNTAX      INTEGER (0..255)
10    UNITS       "dB"
11    MAX-ACCESS  read-write
12    STATUS      current
13    DESCRIPTION
14        "Threshold used by the MS to add a neighbor BS to the
15         diversity set. When the CINR of a neighbor BS is higher
16         than H_Add_Threshold, the MS should send MOB_MSHO-REQ to
17         request adding this neighbor BS to the diversity set.
18         This threshold is used for the MS that is performing
19         MDHO/FBSS HO. If the BS does not support FBSS HO/MDHO,
20         this value is not set."
21    REFERENCE
22        "Table 358, in IEEE Std 802.16e-2005"
23    ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 8 }
24
25
26    wmanIf2BsOfdmaExThresholdDelBsDivSet OBJECT-TYPE
27    SYNTAX      INTEGER (0..255)
28    UNITS       "dB"
29    MAX-ACCESS  read-write
30    STATUS      current
31    DESCRIPTION
32        "Threshold used by the MS to delete a neighbor BS to the
33         diversity set. When the CINR of a neighbor BS is lower
34         than H_Add_Threshold, the MS should send MOB_MSHO-REQ to
35         request dropping this neighbor BS to the diversity set.
36         This threshold is used for the MS that is performing
37         MDHO/FBSS HO. If the BS does not support FBSS HO/MDHO,
38         this value is not set."
39    REFERENCE
40        "Table 358, in IEEE Std 802.16e-2005"
41    ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 9 }
42
43
44
45    wmanIf2BsOfdmaExAsrSlotLength OBJECT-TYPE
46    SYNTAX      INTEGER (0..15)
47    UNITS       "Frames"
48    MAX-ACCESS  read-write
49    STATUS      current
50    DESCRIPTION
51        "Bit 0..3 of ASR Slot Length and Switching Period.
52         For FBSS operation, the time axis is slotted by an ASR
53         (Anchor Switch Reporting) slot that is
54         wmanIf2BsOfdmaExAsrSlotLength frame long."
55    REFERENCE
56        "Table 358, in IEEE Std 802.16e-2005"
57    ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 10 }
58
59
60
61
62
63
64
65

```

```

1  wmanIf2BsOfdmaExAsrSwitchingPeriod OBJECT-TYPE
2      SYNTAX      INTEGER (0..15)
3      UNITS       "ASR slots"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "Bit 0..3 of ASR Slot Length and Switching Period.
8          A switching period is introduced whose duration is equals
9          to wmanIf2BsOfdmaExAsrSwitchingPeriod ASR slots that
10         should be long enough such that certain process (e.g.,
11         HARQ transmission, backhaul context transfer) can be
12         completed at the current anchor BS before the MS switches
13         to the new anchor BS."
14
15     REFERENCE
16         "Table 358, in IEEE Std 802.16e-2005"
17     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 11 }
18
19  wmanIf2BsOfdmaExHytseresisMargin OBJECT-TYPE
20      SYNTAX      INTEGER (0..57)
21      UNITS       "dB"
22      MAX-ACCESS  read-write
23      STATUS      current
24      DESCRIPTION
25          "When the CINR of a neighbor BS is larger than the sum of
26          the CINR of the current serving BS and
27          wmanIf2BsOfdmaExHytseresisMargin for the time-to-trigger
28          duration, then the neighbor BS is included in the list
29          of possible target BSs in MOB_MSHO-REQ."
30
31     REFERENCE
32         "Table 358, in IEEE Std 802.16e-2005"
33     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 12 }
34
35  wmanIf2BsOfdmaExTimeToTrigger OBJECT-TYPE
36      SYNTAX      INTEGER
37      UNITS       "milliseconds"
38      MAX-ACCESS  read-write
39      STATUS      current
40      DESCRIPTION
41          "Indicates the time duration for MS decides to select a
42          neighbor BS as a possible target BS. It is applicable
43          only for HHO."
44
45     REFERENCE
46         "Table 358, in IEEE Std 802.16e-2005"
47     ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 13 }
48
49  wmanIf2BsOfdmaExRetartCount OBJECT-TYPE
50      SYNTAX      INTEGER (0..255)
51      MAX-ACCESS  read-only
52      STATUS      current
53      DESCRIPTION
54          "The value is incremented by one whenever BS restarts
55          (see 6.3.9.11). The value rolls over from 0 to 255."
56
57     REFERENCE
58         "Table 358, in IEEE Std 802.16e-2005"
59
60
61
62
63
64
65

```



```

1      ::= { wmanIf2BsOfdmaExDownlinkChannelEntry 14}
2
3
4      --
5      -- SS object group - containing tables and objects to be implemented in
6      -- the Subscriber station
7
8      --
9      -- wmanIf2SsCps contain the SS Common Part Sublayer objects
10     --
11     wmanIf2SsCps OBJECT IDENTIFIER ::= { wmanIf2SsObjects 1 }
12
13
14     --
15     -- wmanIf2SsConfigurationTable contains global parameters for SS
16     --
17     wmanIf2SsConfigurationTable OBJECT-TYPE
18         SYNTAX      SEQUENCE OF WmanIf2SsConfigurationEntry
19         MAX-ACCESS  not-accessible
20         STATUS      current
21         DESCRIPTION
22             "This table contains one row for the SS system
23             parameters."
24         REFERENCE
25             "Subclause 10.1 in IEEE Std 802.16-2004"
26         ::= { wmanIf2SsCps 1 }
27
28
29     wmanIf2SsConfigurationEntry OBJECT-TYPE
30         SYNTAX      WmanIf2SsConfigurationEntry
31         MAX-ACCESS  not-accessible
32         STATUS      current
33         DESCRIPTION
34             "This table is indexed by ifIndex."
35         INDEX { ifIndex }
36         ::= { wmanIf2SsConfigurationTable 1 }
37
38
39     WmanIf2SsConfigurationEntry ::= SEQUENCE {
40         wmanIf2SsLostDLMapInterval      INTEGER,
41         wmanIf2SsLostULMapInterval      INTEGER,
42         wmanIf2SsContentionRangRetries   INTEGER,
43         wmanIf2SsRequestRetries         INTEGER,
44         wmanIf2SsRegRequestRetries      INTEGER,
45         wmanIf2SsTftpBackoffStart       INTEGER,
46         wmanIf2SsTftpBackoffEnd         INTEGER,
47         wmanIf2SsTftpRequestRetries     INTEGER,
48         wmanIf2SsTftpDownloadRetries    INTEGER,
49         wmanIf2SsTftpWait                INTEGER,
50         wmanIf2SsToDRetries             INTEGER,
51         wmanIf2SsToDRetryPeriod         INTEGER,
52         wmanIf2SsT1Timeout              INTEGER,
53         wmanIf2SsT2Timeout              INTEGER,
54         wmanIf2SsT3Timeout              INTEGER,
55         wmanIf2SsT4Timeout              INTEGER,
56         wmanIf2SsT6Timeout              INTEGER,
57         wmanIf2SsT12Timeout             INTEGER,
58         wmanIf2SsT14Timeout             INTEGER,
59
60
61
62
63
64
65

```

```

1      wmanIf2SsT16Timeout          INTEGER,
2      wmanIf2SsT18Timeout          INTEGER,
3      wmanIf2SsT19Timeout          INTEGER,
4      wmanIf2SsT20Timeout          INTEGER,
5      wmanIf2SsT21Timeout          INTEGER,
6      wmanIf2SsSBCRequestRetries  INTEGER,
7      wmanIf2SsTftpCpltRetries    INTEGER,
8      wmanIf2SsT26Timeout          INTEGER,
9      wmanIf2SsDLManagProcTime    INTEGER}
10
11
12
13  wmanIf2SsLostDLMapInterval OBJECT-TYPE
14      SYNTAX      INTEGER (0..600)
15      UNITS      "milliseconds"
16      MAX-ACCESS  read-write
17      STATUS      current
18      DESCRIPTION
19
20          "Time since last received DL-MAP message before downlink
21          synchronization is considered lost in ms."
22
23      ::= { wmanIf2SsConfigurationEntry 1 }
24
25  wmanIf2SsLostULMapInterval OBJECT-TYPE
26      SYNTAX      INTEGER (0..600)
27      UNITS      "milliseconds"
28      MAX-ACCESS  read-write
29      STATUS      current
30      DESCRIPTION
31
32          "Time since last received UL-MAP message before uplink
33          synchronization is considered lost in ms."
34
35      ::= { wmanIf2SsConfigurationEntry 2 }
36
37  wmanIf2SsContentionRangRetries OBJECT-TYPE
38      SYNTAX      INTEGER (16..65535)
39      MAX-ACCESS  read-write
40      STATUS      current
41      DESCRIPTION
42
43          "Number of retries on contention Ranging Requests."
44
45      ::= { wmanIf2SsConfigurationEntry 3 }
46
47  wmanIf2SsRequestRetries OBJECT-TYPE
48      SYNTAX      INTEGER (16..65535)
49      MAX-ACCESS  read-write
50      STATUS      current
51      DESCRIPTION
52
53          "Number of retries on bandwidth allocation requests."
54
55      ::= { wmanIf2SsConfigurationEntry 4 }
56
57  wmanIf2SsRegRequestRetries OBJECT-TYPE
58      SYNTAX      INTEGER (3..65535)
59      MAX-ACCESS  read-write
60      STATUS      current
61      DESCRIPTION
62
63          "Number of retries on registration requests."
64
65      ::= { wmanIf2SsConfigurationEntry 5 }

```

```

1  wmanIf2SsTftpBackoffStart OBJECT-TYPE
2      SYNTAX      INTEGER (1..65535)
3      UNITS       "seconds"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "Initial value for TFTP backoff in second."
8      ::= { wmanIf2SsConfigurationEntry 6 }
9
10
11
12  wmanIf2SsTftpBackoffEnd OBJECT-TYPE
13      SYNTAX      INTEGER (16..65535)
14      UNITS       "seconds"
15      MAX-ACCESS  read-write
16      STATUS      current
17      DESCRIPTION
18          "Last value for TFTP backoff in second."
19      ::= { wmanIf2SsConfigurationEntry 7 }
20
21
22
23  wmanIf2SsTftpRequestRetries OBJECT-TYPE
24      SYNTAX      INTEGER (16..65535)
25      MAX-ACCESS  read-write
26      STATUS      current
27      DESCRIPTION
28          "Number of retries on TFTP request."
29      ::= { wmanIf2SsConfigurationEntry 8 }
30
31
32
33  wmanIf2SsTftpDownloadRetries OBJECT-TYPE
34      SYNTAX      INTEGER (3..65535)
35      MAX-ACCESS  read-write
36      STATUS      current
37      DESCRIPTION
38          "Number of retries on entire TFTP downloads."
39      ::= { wmanIf2SsConfigurationEntry 9 }
40
41
42
43  wmanIf2SsTftpWait OBJECT-TYPE
44      SYNTAX      INTEGER (2..65535)
45      UNITS       "minutes"
46      MAX-ACCESS  read-write
47      STATUS      current
48      DESCRIPTION
49          "The duration between two consecutive Transfer
50          operational parameters (TFTP) retries in min."
51      ::= { wmanIf2SsConfigurationEntry 10 }
52
53
54
55  wmanIf2SsToDRetries OBJECT-TYPE
56      SYNTAX      INTEGER (3..65535)
57      MAX-ACCESS  read-write
58      STATUS      current
59      DESCRIPTION
60          "Number of Retries to establish the Time of Day."
61      ::= { wmanIf2SsConfigurationEntry 11 }
62
63
64
65  wmanIf2SsToDRetryPeriod OBJECT-TYPE
66      SYNTAX      INTEGER (5..65535)

```

```

1         UNITS          "minutes"
2         MAX-ACCESS    read-write
3         STATUS        current
4         DESCRIPTION
5             "The retry period to re-establishing the Time of Day, as
6             describe in the network entry procedure."
7         ::= { wmanIf2SsConfigurationEntry 12 }
8
9
10        wmanIf2SsT1Timeout OBJECT-TYPE
11            SYNTAX      INTEGER (0..50000)
12            UNITS        "milliseconds"
13            MAX-ACCESS  read-write
14            STATUS      current
15            DESCRIPTION
16                "Wait for DCD timeout in ms."
17            ::= { wmanIf2SsConfigurationEntry 13 }
18
19
20
21        wmanIf2SsT2Timeout OBJECT-TYPE
22            SYNTAX      INTEGER (0..10000)
23            UNITS        "milliseconds"
24            MAX-ACCESS  read-write
25            STATUS      current
26            DESCRIPTION
27                "Wait for broadcast ranging timeout in ms."
28            ::= { wmanIf2SsConfigurationEntry 14 }
29
30
31
32        wmanIf2SsT3Timeout OBJECT-TYPE
33            SYNTAX      INTEGER (0..200)
34            UNITS        "milliseconds"
35            MAX-ACCESS  read-write
36            STATUS      current
37            DESCRIPTION
38                "Ranging Response reception timeout following the
39                transmission of a Ranging Request in ms."
40            ::= { wmanIf2SsConfigurationEntry 15 }
41
42
43
44        wmanIf2SsT4Timeout OBJECT-TYPE
45            SYNTAX      INTEGER (30..35)
46            UNITS        "seconds"
47            MAX-ACCESS  read-write
48            STATUS      current
49            DESCRIPTION
50                "Wait for unicast ranging opportunity. If the pending until
51                complete field was used earlier by this SS, then the value
52                of that field shall be added to this interval in second."
53            ::= { wmanIf2SsConfigurationEntry 16 }
54
55
56
57        wmanIf2SsT6Timeout OBJECT-TYPE
58            SYNTAX      INTEGER (0..3000)
59            UNITS        "milliseconds"
60            MAX-ACCESS  read-write
61            STATUS      current
62            DESCRIPTION
63                "Wait for registration response in ms."
64
65

```

```

1      ::= { wmanIf2SsConfigurationEntry 17 }
2
3
4      wmanIf2SsT12Timeout OBJECT-TYPE
5          SYNTAX      INTEGER (0..50000)
6          UNITS       "milliseconds"
7          MAX-ACCESS  read-write
8          STATUS      current
9          DESCRIPTION
10             "Wait for UCD descriptor in ms."
11      ::= { wmanIf2SsConfigurationEntry 18 }
12
13
14      wmanIf2SsT14Timeout OBJECT-TYPE
15          SYNTAX      INTEGER (0..200)
16          UNITS       "milliseconds"
17          MAX-ACCESS  read-write
18          STATUS      current
19          DESCRIPTION
20             "Wait for DSX-RVD Timeout in ms."
21      ::= { wmanIf2SsConfigurationEntry 19 }
22
23
24
25      wmanIf2SsT16Timeout OBJECT-TYPE
26          SYNTAX      INTEGER (10..65535)
27          UNITS       "milliseconds"
28          MAX-ACCESS  read-write
29          STATUS      current
30          DESCRIPTION
31             "wait for bandwidth request grant in ms."
32      ::= { wmanIf2SsConfigurationEntry 20 }
33
34
35
36      wmanIf2SsT18Timeout OBJECT-TYPE
37          SYNTAX      INTEGER (0..65535)
38          UNITS       "milliseconds"
39          MAX-ACCESS  read-write
40          STATUS      current
41          DESCRIPTION
42             "wait for SBC-RSP timeout in ms."
43      ::= { wmanIf2SsConfigurationEntry 21 }
44
45
46
47      wmanIf2SsT19Timeout OBJECT-TYPE
48          SYNTAX      INTEGER (0..1048575)
49          UNITS       "milliseconds"
50          MAX-ACCESS  read-write
51          STATUS      current
52          DESCRIPTION
53             "Time DL-channel remains unusable in ms."
54      ::= { wmanIf2SsConfigurationEntry 22 }
55
56
57
58      wmanIf2SsT20Timeout OBJECT-TYPE
59          SYNTAX      INTEGER (0..65535)
60          UNITS       "milliseconds"
61          MAX-ACCESS  read-write
62          STATUS      current
63          DESCRIPTION
64             "Time SS searches for preambles on a given channel in ms."
65

```

```

1      ::= { wmanIf2SsConfigurationEntry 23 }
2
3
4      wmanIf2SsT21Timeout OBJECT-TYPE
5          SYNTAX      INTEGER (0..10000)
6          UNITS        "milliseconds"
7          MAX-ACCESS  read-write
8          STATUS      current
9          DESCRIPTION
10             "Time SS searches for DL-MAP on a given channel in ms."
11      ::= { wmanIf2SsConfigurationEntry 24 }
12
13
14      wmanIf2SsSBCRequestRetries OBJECT-TYPE
15          SYNTAX      INTEGER (3..16)
16          MAX-ACCESS  read-write
17          STATUS      current
18          DESCRIPTION
19             "Number of retries on SBC Request."
20      ::= { wmanIf2SsConfigurationEntry 25 }
21
22
23
24      wmanIf2SsTftpCpltRetries OBJECT-TYPE
25          SYNTAX      INTEGER (3..16)
26          MAX-ACCESS  read-write
27          STATUS      current
28          DESCRIPTION
29             "Number of retries on TFTP-CPLT."
30      ::= { wmanIf2SsConfigurationEntry 26 }
31
32
33
34      wmanIf2SsT26Timeout OBJECT-TYPE
35          SYNTAX      INTEGER (10..200)
36          UNITS        "milliseconds"
37          MAX-ACCESS  read-write
38          STATUS      current
39          DESCRIPTION
40             "Wait for TFTP-RSP in ms."
41      ::= { wmanIf2SsConfigurationEntry 27 }
42
43
44
45      wmanIf2SsDLManagProcTime OBJECT-TYPE
46          SYNTAX      INTEGER (0..200)
47          UNITS        "micro seconds"
48          MAX-ACCESS  read-write
49          STATUS      current
50          DESCRIPTION
51             "Max. time between reception of Fast Power Control
52             management message and compliance to its instructions
53             by SS in us."
54      ::= { wmanIf2SsConfigurationEntry 28 }
55
56
57      --
58      -- Subscriber Channel Measurement Table
59      --
60
61      wmanIf2SsChannelMeasurementTable OBJECT-TYPE
62          SYNTAX      SEQUENCE OF WmanIf2SsChannelMeasurementEntry
63          MAX-ACCESS  not-accessible
64          STATUS      current
65

```

```

1      DESCRIPTION
2          "This table contains downlink channel measurement
3          information for each SS."
4
5      REFERENCE
6          "6.3.2.3.33 in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsCps 2 }
8
9
10     wmanIf2SsChannelMeasurementEntry OBJECT-TYPE
11         SYNTAX      WmanIf2SsChannelMeasurementEntry
12         MAX-ACCESS  not-accessible
13         STATUS      current
14         DESCRIPTION
15             "Each entry in the table contains RSSI and CINR
16             signal quality measurement taken from the SS. The primary
17             index is the ifIndex pointing to SS.
18             wmanIf2CmnHistogramIndex is the index to histogram
19             samples. Since there is no time stamp in the table,
20             wmanIf2SsHistogramIndex should be increased monotonically,
21             and wraps around when it reaches the limit.
22             When the measurement entry for a SS reaches the limit,
23             the oldest entry shall be deleted as the new entry is
24             added to the table."
25         INDEX      { ifIndex, wmanIf2SsHistogramIndex }
26         ::= { wmanIf2SsChannelMeasurementTable 1 }
27
28
29
30
31     WmanIf2SsChannelMeasurementEntry ::= SEQUENCE {
32         wmanIf2SsHistogramIndex      Unsigned32,
33         wmanIf2SsChannelNumber       WmanIf2ChannelNumber,
34         wmanIf2SsStartFrame          INTEGER,
35         wmanIf2SsDuration             INTEGER,
36         wmanIf2SsBasicReport         BITS,
37         wmanIf2SsMeanCinrReport      INTEGER,
38         wmanIf2SsStdDeviationCinrReport  INTEGER,
39         wmanIf2SsMeanRssiReport      INTEGER,
40         wmanIf2SsStdDeviationRssiReport  INTEGER}
41
42
43
44     wmanIf2SsHistogramIndex OBJECT-TYPE
45         SYNTAX      Unsigned32 (1 .. 4294967295)
46         MAX-ACCESS  not-accessible
47         STATUS      current
48         DESCRIPTION
49             "wmanIf2SsHistogramIndex identifies the histogram samples
50             in the table for each subscriber station."
51         ::= { wmanIf2SsChannelMeasurementEntry 1 }
52
53
54
55     wmanIf2SsChannelNumber OBJECT-TYPE
56         SYNTAX      WmanIf2ChannelNumber
57         MAX-ACCESS  read-only
58         STATUS      current
59         DESCRIPTION
60             "Physical channel number to be reported on."
61         REFERENCE
62             "Subclause 8.5.1 in IEEE Std 802.16-2004"
63         ::= { wmanIf2SsChannelMeasurementEntry 2 }
64
65

```

```

1
2
3 wmanIf2SsStartFrame OBJECT-TYPE
4     SYNTAX      INTEGER (0 .. 65535)
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "Frame number in which measurement for this channel
9         started."
10
11    REFERENCE
12        "Subclause 11.12 in IEEE Std 802.16-2004"
13    ::= { wmanIf2SsChannelMeasurementEntry 3 }
14
15 wmanIf2SsDuration OBJECT-TYPE
16     SYNTAX      INTEGER (0..16777215)
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20         "Cumulative measurement duration on the channel in
21         multiples of Ts. For any value exceeding 0xFFFFF,
22         report 0xFFFFF."
23
24    REFERENCE
25        "Subclause 11.12 in IEEE Std 802.16-2004"
26    ::= { wmanIf2SsChannelMeasurementEntry 4 }
27
28
29
30 wmanIf2SsBasicReport OBJECT-TYPE
31     SYNTAX      BITS { wirelessHuman(0),
32                    unknownTransmission(1),
33                    primaryUser(2),
34                    channelNotMeasured(3) }
35
36     MAX-ACCESS  read-only
37     STATUS      current
38     DESCRIPTION
39         "Bit #0: WirelessHUMAN detected on the channel
40         Bit #1: Unknown transmissions detected on the channel
41         Bit #2: Primary User detected on the channel
42         Bit #3: Unmeasured. Channel not measured"
43
44    REFERENCE
45        "Subclause 11.12 in IEEE Std 802.16-2004"
46    ::= { wmanIf2SsChannelMeasurementEntry 5 }
47
48
49 wmanIf2SsMeanCinrReport OBJECT-TYPE
50     SYNTAX      INTEGER (-20 .. 37)
51     UNITS        "dB"
52     MAX-ACCESS  read-only
53     STATUS      current
54     DESCRIPTION
55         "Mean CINR report."
56
57    REFERENCE
58        "Subclause 8.2.2, 8.3.9 in IEEE Std 802.16-2004"
59    ::= { wmanIf2SsChannelMeasurementEntry 6 }
60
61
62 wmanIf2SsStdDeviationCinrReport OBJECT-TYPE
63     SYNTAX      INTEGER (0 .. 29)
64     UNITS        "dB"
65

```



```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "Standard deviation CINR report."
5      REFERENCE
6          "Subclause 8.2.2 and Subclause 8.3.9 in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsChannelMeasurementEntry 7 }
8
9
10
11 wmanIf2SsMeanRssiReport OBJECT-TYPE
12     SYNTAX      INTEGER (-123 .. -40)
13     UNITS       "dBm"
14     MAX-ACCESS  read-only
15     STATUS      current
16     DESCRIPTION
17         "Mean RSSI report."
18     REFERENCE
19         "Subclause 8.2.2 and Subclause 8.3.9 in IEEE Std 802.16-2004"
20     ::= { wmanIf2SsChannelMeasurementEntry 8 }
21
22
23
24 wmanIf2SsStdDeviationRssiReport OBJECT-TYPE
25     SYNTAX      INTEGER (0 .. 42)
26     UNITS       "dB"
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "Standard deviation RSSI report."
31     REFERENCE
32         "Subclause 8.2.2 and Subclause 8.3.9 in IEEE Std 802.16-2004"
33     ::= { wmanIf2SsChannelMeasurementEntry 9 }
34
35
36
37 --
38 -- Subscriber station PKM group
39 -- wmanIf2SsPkmObjects contain the Subscriber Station Privacy Sublayer
40 -- objects
41 --
42 --
43 wmanIf2SsPkmObjects OBJECT IDENTIFIER ::= { wmanIf2SsObjects 2 }
44
45 --
46 -- Table wmanIf2SsPkmAuthTable
47 --
48 --
49 wmanIf2SsPkmAuthTable OBJECT-TYPE
50     SYNTAX      SEQUENCE OF WmanIf2SsPkmAuthEntry
51     MAX-ACCESS  not-accessible
52     STATUS      current
53     DESCRIPTION
54         "This table describes the PKM attributes related
55         to the authorization for each SS wireless interface."
56     ::= { wmanIf2SsPkmObjects 1 }
57
58
59
60 wmanIf2SsPkmAuthEntry OBJECT-TYPE
61     SYNTAX      WmanIf2SsPkmAuthEntry
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65

```

```

1         "Each entry contains objects describing attributes of one
2         SS wireless interface."
3     INDEX      { ifIndex }
4     ::= { wmanIf2SsPkmAuthTable 1 }
5
6
7     WmanIf2SsPkmAuthEntry ::= SEQUENCE {
8         wmanIf2SsPkmAuthState          INTEGER,
9         wmanIf2SsPkmAuthKeySequenceNumber Integer32,
10        wmanIf2SsPkmAuthExpiresOld     DateAndTime,
11        wmanIf2SsPkmAuthExpiresNew     DateAndTime,
12        wmanIf2SsPkmAuthReset          TruthValue,
13        wmanIf2SsPkmAuthentInfos       Counter32,
14        wmanIf2SsPkmAuthRequests       Counter32,
15        wmanIf2SsPkmAuthReplies        Counter32,
16        wmanIf2SsPkmAuthRejects       Counter32,
17        wmanIf2SsPkmAuthInvalids      Counter32,
18        wmanIf2SsPkmAuthRejectErrorCode INTEGER,
19        wmanIf2SsPkmAuthRejectErrorString SnmpAdminString,
20        wmanIf2SsPkmAuthInvalidErrorCode INTEGER,
21        wmanIf2SsPkmAuthInvalidErrorString SnmpAdminString,
22        wmanIf2SsPkmAuthGraceTime      Integer32,
23        wmanIf2SsPkmTekGraceTime       Integer32,
24        wmanIf2SsPkmAuthWaitTimeout    Integer32,
25        wmanIf2SsPkmReauthWaitTimeout  Integer32,
26        wmanIf2SsPkmOpWaitTimeout      Integer32,
27        wmanIf2SsPkmRekeyWaitTimeout   Integer32,
28        wmanIf2SsPkmAuthRejectWaitTimeout Integer32 }
29
30
31
32
33
34
35     wmanIf2SsPkmAuthState OBJECT-TYPE
36     SYNTAX      INTEGER { start(1),
37                 authWait(2),
38                 authorized(3),
39                 reauthWait(4),
40                 authRejectWait(5),
41                 silent(6) }
42
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46         "The value of this object is the state of the SS
47         authorization FSM. The start state indicates that FSM is
48         in its initial state."
49     ::= { wmanIf2SsPkmAuthEntry 1 }
50
51
52
53     wmanIf2SsPkmAuthKeySequenceNumber OBJECT-TYPE
54     SYNTAX      Integer32 (0..15)
55     MAX-ACCESS  read-only
56     STATUS      current
57     DESCRIPTION
58         "The value of this object is the most recent authorization
59         key sequence number for this FSM."
60     ::= { wmanIf2SsPkmAuthEntry 2 }
61
62
63
64     wmanIf2SsPkmAuthExpiresOld OBJECT-TYPE
65     SYNTAX      DateAndTime

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the actual clock time for
5          expiration of the immediate predecessor of the most recent
6          authorization key for this FSM.  If this FSM has only one
7          authorization key, then the value is the time of activation
8          of this FSM."
9      ::= { wmanIf2SsPkmAuthEntry 3 }
10
11
12
13 wmanIf2SsPkmAuthExpiresNew OBJECT-TYPE
14     SYNTAX      DateAndTime
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is the actual clock time for
19         expiration of the most recent authorization key for this
20         FSM."
21     ::= { wmanIf2SsPkmAuthEntry 4 }
22
23
24
25 wmanIf2SsPkmAuthReset OBJECT-TYPE
26     SYNTAX      TruthValue
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "Setting this object to TRUE generates a Reauthorize event
31         in the authorization FSM.  Reading this object always
32         returns FALSE."
33     ::= { wmanIf2SsPkmAuthEntry 5 }
34
35
36
37 wmanIf2SsPkmAuthentInfos OBJECT-TYPE
38     SYNTAX      Counter32
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "The value of this object is the count of times the SS has
43         transmitted an Authentication Information message."
44     ::= { wmanIf2SsPkmAuthEntry 6 }
45
46
47
48 wmanIf2SsPkmAuthRequests OBJECT-TYPE
49     SYNTAX      Counter32
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The value of this object is the count of times the SS has
54         transmitted an Authorization Request message."
55     ::= { wmanIf2SsPkmAuthEntry 7 }
56
57
58
59 wmanIf2SsPkmAuthReplies OBJECT-TYPE
60     SYNTAX      Counter32
61     MAX-ACCESS  read-only
62     STATUS      current
63     DESCRIPTION
64         "The value of this object is the count of times the SS has
65

```

```

1         received an Authorization Reply message."
2     ::= { wmanIf2SsPkmAuthEntry 8 }
3
4
5 wmanIf2SsPkmAuthRejects OBJECT-TYPE
6     SYNTAX      Counter32
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10
11         "The value of this object is the count of times the SS has
12         received an Authorization Reject message."
13     ::= { wmanIf2SsPkmAuthEntry 9 }
14
15 wmanIf2SsPkmAuthInvalids OBJECT-TYPE
16     SYNTAX      Counter32
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20
21         "The value of this object is the count of times the SS has
22         received an Authorization Invalid message."
23     ::= { wmanIf2SsPkmAuthEntry 10 }
24
25
26 wmanIf2SsPkmAuthRejectErrorCode OBJECT-TYPE
27     SYNTAX      INTEGER { none(1),
28                 unknown(2),
29                 unauthorizedSs(3),
30                 unauthorizedSaid(4),
31                 permanentAuthorizationFailure(8),
32                 timeOfDayNotAcquired(11) }
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36
37         "The value of this object is the enumerated description of
38         the Error-Code in most recent Authorization Reject message
39         received by the SS.  This has value unknown(2) if the last
40         Error-Code value was 0, and none(1) if no Authorization
41         Reject message has been received since reboot."
42     ::= { wmanIf2SsPkmAuthEntry 11 }
43
44
45 wmanIf2SsPkmAuthRejectErrorString OBJECT-TYPE
46     SYNTAX      SnmpAdminString (SIZE (0..128))
47     MAX-ACCESS  read-only
48     STATUS      current
49     DESCRIPTION
50
51         "The value of this object is the Display-String in most
52         recent Authorization Reject message received by the SS.
53         This is a zero length string if no Authorization Reject
54         message has been received since reboot."
55     ::= { wmanIf2SsPkmAuthEntry 12 }
56
57
58 wmanIf2SsPkmAuthInvalidErrorCode OBJECT-TYPE
59     SYNTAX      INTEGER { none(1),
60                 unknown(2),
61                 unauthorizedSs(3),
62                 unsolicited(5),
63
64
65

```

```

1         invalidKeySequence(6),
2         keyRequestAuthenticationFailure(7) }
3
4     MAX-ACCESS    read-only
5     STATUS       current
6     DESCRIPTION
7         "The value of this object is the enumerated description of
8         the Error-Code in most recent Authorization Invalid message
9         received by the SS. This has value unknown(2) if the last
10        Error-Code value was 0, and none(1) if no Authorization
11        Invalid message has been received since reboot."
12
13    ::= { wmanIf2SsPkmAuthEntry 13 }
14
15    wmanIf2SsPkmAuthInvalidErrorString OBJECT-TYPE
16        SYNTAX      SnmpAdminString (SIZE (0..128))
17        MAX-ACCESS  read-only
18        STATUS      current
19        DESCRIPTION
20            "The value of this object is the Display-String in most
21            recent Authorization Invalid message received by the SS.
22            This is a zero length string if no Authorization Invalid
23            message has been received since reboot."
24
25        ::= { wmanIf2SsPkmAuthEntry 14 }
26
27
28    wmanIf2SsPkmAuthGraceTime OBJECT-TYPE
29        SYNTAX      Integer32 (300..3024000)
30        UNITS       "seconds"
31        MAX-ACCESS  read-only
32        STATUS      current
33        DESCRIPTION
34            "The value of this object is the grace time for an
35            authorization key. A SS is expected to start trying to get
36            a new authorization key beginning AuthGraceTime seconds
37            before the authorization key actually expires."
38
39        REFERENCE
40            "Table 343 in IEEE Std 802.16-2004"
41
42        DEFVAL      { 600 }
43
44        ::= { wmanIf2SsPkmAuthEntry 15 }
45
46
47    wmanIf2SsPkmTekGraceTime OBJECT-TYPE
48        SYNTAX      Integer32 (300..3024000)
49        UNITS       "seconds"
50        MAX-ACCESS  read-only
51        STATUS      current
52        DESCRIPTION
53            "The value of this object is the grace time for the TEK in
54            seconds. The SS is expected to start trying to acquire a
55            new TEK beginning TEK GraceTime seconds before the
56            expiration of the most recent TEK."
57
58        REFERENCE
59            "Table 343 in IEEE Std 802.16-2004"
60
61        DEFVAL      { 3600 }
62
63        ::= { wmanIf2SsPkmAuthEntry 16 }
64
65    wmanIf2SsPkmAuthWaitTimeout OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (2..30)
2      UNITS       "seconds"
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The value of this object is the Authorize Wait Timeout."
7      REFERENCE
8          "Table 343 in IEEE Std 802.16-2004"
9      DEFVAL      { 10 }
10     ::= { wmanIf2SsPkmAuthEntry 17 }
11
12
13
14     wmanIf2SsPkmReauthWaitTimeout OBJECT-TYPE
15         SYNTAX      Integer32 (2..30)
16         UNITS       "seconds"
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "The value of this object is the Reauthorize Wait Timeout
21             in seconds."
22         REFERENCE
23             "Table 343 in IEEE Std 802.16-2004"
24         DEFVAL      { 10 }
25         ::= { wmanIf2SsPkmAuthEntry 18 }
26
27
28
29
30     wmanIf2SsPkmOpWaitTimeout OBJECT-TYPE
31         SYNTAX      Integer32 (1..10)
32         UNITS       "seconds"
33         MAX-ACCESS  read-only
34         STATUS      current
35         DESCRIPTION
36             "The value of this object is the Operational Wait Timeout
37             in seconds."
38         REFERENCE
39             "Table 343 in IEEE Std 802.16-2004"
40         DEFVAL      { 1 }
41         ::= { wmanIf2SsPkmAuthEntry 19 }
42
43
44
45
46     wmanIf2SsPkmRekeyWaitTimeout OBJECT-TYPE
47         SYNTAX      Integer32 (1..10)
48         UNITS       "seconds"
49         MAX-ACCESS  read-only
50         STATUS      current
51         DESCRIPTION
52             "The value of this object is the Rekey Wait Timeout in
53             seconds."
54         REFERENCE
55             "Table 343 in IEEE Std 802.16-2004"
56         DEFVAL      { 1 }
57         ::= { wmanIf2SsPkmAuthEntry 20 }
58
59
60
61
62     wmanIf2SsPkmAuthRejectWaitTimeout OBJECT-TYPE
63         SYNTAX      Integer32 (10..600)
64         UNITS       "seconds"
65         MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object is the Authorization Reject Wait
4          Timeout in seconds."
5      REFERENCE
6          "Table 343 in IEEE Std 802.16-2004"
7      DEFVAL      { 60 }
8      ::= { wmanIf2SsPkmAuthEntry 21 }
9
10
11
12  --
13  -- Table wmanIf2SsPkmTekTable
14  --
15  wmanIf2SsPkmTekTable OBJECT-TYPE
16      SYNTAX      SEQUENCE OF WmanIf2SsPkmTekEntry
17      MAX-ACCESS  not-accessible
18      STATUS      current
19      DESCRIPTION
20          "This table describes the attributes of each SS Traffic
21          Encryption Key (TEK) association. The SS maintains (no more
22          than) one TEK association per SAID per SS wireless
23          interface."
24      ::= { wmanIf2SsPkmObjects 2 }
25
26
27
28  wmanIf2SsPkmTekEntry OBJECT-TYPE
29      SYNTAX      WmanIf2SsPkmTekEntry
30      MAX-ACCESS  not-accessible
31      STATUS      current
32      DESCRIPTION
33          "Each entry contains objects describing the TEK association
34          attributes of one SAID. The SS MUST create one entry per
35          SAID, regardless of whether the SAID was obtained from a
36          Registration Response message, from an Authorization Reply
37          message, or from any dynamic SAID establishment
38          mechanisms."
39      INDEX      { ifIndex, wmanIf2SsPkmTekSAID }
40      ::= { wmanIf2SsPkmTekTable 1 }
41
42
43
44  WmanIf2SsPkmTekEntry ::= SEQUENCE {
45      wmanIf2SsPkmTekSAID          INTEGER,
46      wmanIf2SsPkmTekSAType        INTEGER,
47      wmanIf2SsPkmTekDataEncryptAlg WmanIf2DataEncryptAlgId,
48      wmanIf2SsPkmTekDataAuthAlg   WmanIf2DataAuthAlgId,
49      wmanIf2SsPkmTekEncryptAlg    WmanIf2TekEncryptAlgId,
50      wmanIf2SsPkmTekState         INTEGER,
51      wmanIf2SsPkmTekKeySequenceNumber Integer32,
52      wmanIf2SsPkmTekExpiresOld    DateAndTime,
53      wmanIf2SsPkmTekExpiresNew    DateAndTime,
54      wmanIf2SsPkmTekKeyRequests   Counter32,
55      wmanIf2SsPkmTekKeyReplies    Counter32,
56      wmanIf2SsPkmTekKeyRejects    Counter32,
57      wmanIf2SsPkmTekInvalids      Counter32,
58      wmanIf2SsPkmTekAuthPends     Counter32,
59      wmanIf2SsPkmTekKeyRejectErrorCode INTEGER,
60      wmanIf2SsPkmTekKeyRejectErrorString SnmpAdminString,
61
62
63
64
65

```

```

1      wmanIf2SsPkmTekInvalidErrorCode      INTEGER,
2      wmanIf2SsPkmTekInvalidErrorString   SnmpAdminString}
3
4
5  wmanIf2SsPkmTekSAId OBJECT-TYPE
6      SYNTAX      INTEGER (0..65535)
7      MAX-ACCESS  not-accessible
8      STATUS      current
9      DESCRIPTION
10         "The value of this object is the Security Association
11         ID (SAID)."
```

```

12     ::= { wmanIf2SsPkmTekEntry 1 }
13
14
15  wmanIf2SsPkmTekSAType OBJECT-TYPE
16      SYNTAX      INTEGER {primarySA(0),
17                    staticSA(1),
18                    dynamicSA(2)}
19      MAX-ACCESS  read-only
20      STATUS      current
21      DESCRIPTION
22         "The value of this object is the type of security
23         association."
```

```

24     REFERENCE
25         "IEEE Std 802.16-2004; 11.9.18"
```

```

26     ::= { wmanIf2SsPkmTekEntry 2 }
27
28
29
30
31  wmanIf2SsPkmTekDataEncryptAlg OBJECT-TYPE
32      SYNTAX      WmanIf2DataEncryptAlgId
33      MAX-ACCESS  read-only
34      STATUS      current
35      DESCRIPTION
36         "The value of this object is the data encryption algorithm
37         being utilized."
```

```

38     REFERENCE
39         "Table 375, IEEE Std 802.16-2004"
```

```

40     ::= { wmanIf2SsPkmTekEntry 3 }
41
42
43
44  wmanIf2SsPkmTekDataAuthentAlg OBJECT-TYPE
45      SYNTAX      WmanIf2DataAuthAlgId
46      MAX-ACCESS  read-only
47      STATUS      current
48      DESCRIPTION
49         "The value of this object is the data authentication
50         algorithm being utilized."
```

```

51     REFERENCE
52         "Table 376, IEEE Std 802.16-2004"
```

```

53     ::= { wmanIf2SsPkmTekEntry 4 }
54
55
56
57  wmanIf2SsPkmTekEncryptAlg OBJECT-TYPE
58      SYNTAX      WmanIf2TekEncryptAlgId
59      MAX-ACCESS  read-only
60      STATUS      current
61      DESCRIPTION
62         "The value of this object is the TEK key encryption
63         algorithm for this cryptographic suite capability."
```



```

1      REFERENCE
2          "Table 377, IEEE Std 802.16-2004"
3      ::= { wmanIf2SsPkmTekEntry 5 }
4
5
6      wmanIf2SsPkmTekState OBJECT-TYPE
7          SYNTAX      INTEGER {start(1),
8                      opWait(2),
9                      opReauthWait(3),
10                     operational(4),
11                     rekeyWait(5),
12                     rekeyReauthWait(6)}
13
14          MAX-ACCESS  read-only
15          STATUS      current
16          DESCRIPTION
17              "The value of this object is the state of the indicated TEK
18              FSM. The start(1) state indicates that FSM is in its
19              initial state."
20          ::= { wmanIf2SsPkmTekEntry 6 }
21
22
23
24      wmanIf2SsPkmTekKeySequenceNumber OBJECT-TYPE
25          SYNTAX      Integer32 (0..3)
26          MAX-ACCESS  read-only
27          STATUS      current
28          DESCRIPTION
29              "The value of this object is the most recent TEK key
30              sequence number for this TEK FSM."
31          REFERENCE
32              "IEEE Std 802.16-2004; 11.9.5"
33          ::= { wmanIf2SsPkmTekEntry 7 }
34
35
36
37      wmanIf2SsPkmTekExpiresOld OBJECT-TYPE
38          SYNTAX      DateAndTime
39          MAX-ACCESS  read-only
40          STATUS      current
41          DESCRIPTION
42              "The value of this object is the actual clock time for
43              expiration of the immediate predecessor of the most recent
44              TEK for this FSM. If this FSM has only one TEK, then the
45              value is the time of activation of this FSM."
46          ::= { wmanIf2SsPkmTekEntry 8 }
47
48
49
50      wmanIf2SsPkmTekExpiresNew OBJECT-TYPE
51          SYNTAX      DateAndTime
52          MAX-ACCESS  read-only
53          STATUS      current
54          DESCRIPTION
55              "The value of this object is the actual clock time for
56              expiration of the most recent TEK for this FSM."
57          ::= { wmanIf2SsPkmTekEntry 9 }
58
59
60
61      wmanIf2SsPkmTekKeyRequests OBJECT-TYPE
62          SYNTAX      Counter32
63          MAX-ACCESS  read-only
64          STATUS      current
65

```

```

1      DESCRIPTION
2          "The value of this object is the count of times the SS has
3          transmitted a Key Request message."
4      ::= { wmanIf2SsPkmTekEntry 10 }
5
6
7      wmanIf2SsPkmTekKeyReplies OBJECT-TYPE
8          SYNTAX      Counter32
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "The value of this object is the count of times the SS has
13             received a Key Reply message, including a message whose
14             authentication failed."
15         ::= { wmanIf2SsPkmTekEntry 11 }
16
17
18
19         wmanIf2SsPkmTekKeyRejects OBJECT-TYPE
20             SYNTAX      Counter32
21             MAX-ACCESS  read-only
22             STATUS      current
23             DESCRIPTION
24                 "The value of this object is the count of times the SS has
25                 received a Key Reject message, including a message whose
26                 authentication failed."
27             ::= { wmanIf2SsPkmTekEntry 12 }
28
29
30
31         wmanIf2SsPkmTekInvalids OBJECT-TYPE
32             SYNTAX      Counter32
33             MAX-ACCESS  read-only
34             STATUS      current
35             DESCRIPTION
36                 "The value of this object is the count of times the SS has
37                 received a TEK Invalid message, including a message whose
38                 authentication failed."
39             ::= { wmanIf2SsPkmTekEntry 13 }
40
41
42
43         wmanIf2SsPkmTekAuthPends OBJECT-TYPE
44             SYNTAX      Counter32
45             MAX-ACCESS  read-only
46             STATUS      current
47             DESCRIPTION
48                 "The value of this object is the count of times an
49                 Authorization Pending (Auth Pend) event occurred in this
50                 FSM."
51             ::= { wmanIf2SsPkmTekEntry 14 }
52
53
54
55         wmanIf2SsPkmTekKeyRejectErrorCode OBJECT-TYPE
56             SYNTAX      INTEGER { none(1),
57                         unknown(2),
58                         unauthorizedSaid(4) }
59             MAX-ACCESS  read-only
60             STATUS      current
61             DESCRIPTION
62                 "The value of this object is the enumerated description of
63                 the Error-Code in most recent Key Reject message received
64
65

```

```

1         by the SS. This has value unknown(2) if the last Error-Code
2         value was 0, and none(1) if no Key Reject message has been
3         received since reboot."
4     ::= { wmanIf2SsPkmTekEntry 15 }
5
6
7 wmanIf2SsPkmTekKeyRejectErrorString OBJECT-TYPE
8     SYNTAX      SnmpAdminString (SIZE (0..128))
9     MAX-ACCESS  read-only
10    STATUS      current
11    DESCRIPTION
12        "The value of this object is the Display-String in most
13        recent Key Reject message received by the SS. This is a
14        zero length string if no Key Reject message has been
15        received since reboot."
16    ::= { wmanIf2SsPkmTekEntry 16 }
17
18
19
20 wmanIf2SsPkmTekInvalidErrorCode OBJECT-TYPE
21     SYNTAX      INTEGER {none(1),
22                 unknown(2),
23                 invalidKeySequence(6)}
24     MAX-ACCESS  read-only
25     STATUS      current
26     DESCRIPTION
27         "The value of this object is the enumerated description of
28         the Error-Code in most recent TEK Invalid message received
29         by the SS. This has value unknown(2) if the last
30         Error-Code value was 0, and none(1) if no TEK Invalid
31         message has been received since reboot."
32     ::= { wmanIf2SsPkmTekEntry 17 }
33
34
35
36
37 wmanIf2SsPkmTekInvalidErrorString OBJECT-TYPE
38     SYNTAX      SnmpAdminString (SIZE (0..128))
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "The value of this object is the Display-String in most
43         recent TEK Invalid message received by the SS. This is a
44         zero length string if no TEK Invalid message has been
45         received since reboot."
46     ::= { wmanIf2SsPkmTekEntry 18 }
47
48
49
50 --
51 -- Table wmanIf2SsDeviceCertTable
52 --
53
54 wmanIf2SsDeviceCertTable OBJECT-TYPE
55     SYNTAX      SEQUENCE OF WmanIf2SsDeviceCertEntry
56     MAX-ACCESS  not-accessible
57     STATUS      current
58     DESCRIPTION
59         "This table describes the PKM device certificates for each
60         SS wireless interface."
61     ::= { wmanIf2SsPkmObjects 3 }
62
63
64
65 wmanIf2SsDeviceCertEntry OBJECT-TYPE

```

```

1      SYNTAX      WmanIf2SsDeviceCertEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "Each entry contains the device certificate of one SS."
6      INDEX      { ifIndex }
7      ::= { wmanIf2SsDeviceCertTable 1 }
8
9
10
11 WmanIf2SsDeviceCertEntry ::= SEQUENCE {
12     wmanIf2SsDeviceCert          OCTET STRING,
13     wmanIf2SsDeviceManufCert     OCTET STRING}
14
15 wmanIf2SsDeviceCert OBJECT-TYPE
16     SYNTAX      OCTET STRING (SIZE(0..65535))
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20         "The X509 DER-encoded subscriber station certificate."
21     ::= { wmanIf2SsDeviceCertEntry 1 }
22
23
24
25 wmanIf2SsDeviceManufCert OBJECT-TYPE
26     SYNTAX      OCTET STRING (SIZE(0..65535))
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "The X509 DER-encoded manufacturer certificate which is
31         signed by the CA root authority certificate."
32     ::= { wmanIf2SsDeviceCertEntry 2 }
33
34
35 --
36 --
37 -- Subscriber station Notification Group
38 -- wmanIf2SsNotificationObjects contains the SS SNMP Trap objects
39 --
40
41 wmanIf2SsNotification OBJECT IDENTIFIER ::= { wmanIf2SsObjects 3 }
42 wmanIf2SsTrapControl OBJECT IDENTIFIER ::= { wmanIf2SsNotification 1 }
43 wmanIf2SsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIf2SsNotification 2
44 }
45
46
47 -- This object groups all NOTIFICATION-TYPE objects for SS.
48 -- It is defined following RFC2758 sections 8.5 and 8.6
49 -- for the compatibility with SNMPv1.
50 wmanIf2SsTrapPrefix OBJECT IDENTIFIER ::= { wmanIf2SsTrapDefinitions 0 }
51
52
53 wmanIf2SsTrapControlRegister OBJECT-TYPE
54     SYNTAX      BITS {wmanIf2SsTlvUnknown(0),
55                    wmanIf2SsDynamicServiceFail(1),
56                    wmanIf2SsDhcpSuccess(2),
57                    wmanIf2SsRssiStatusChange(3)}
58     MAX-ACCESS  read-write
59     STATUS      current
60     DESCRIPTION
61         "The object is used to enable Subscriber Station traps.
62         From left to right, the set bit indicates the corresponding
63         Subscriber Station trap is enabled."
64
65

```

```

1      ::= { wmanIf2SsTrapControl 1 }
2
3
4  wmanIf2SsThresholdConfigTable OBJECT-TYPE
5      SYNTAX      SEQUENCE OF WmanIf2SsThresholdConfigEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "This table contains threshold objects that can be set to
10         detect the threshold crossing events."
11      ::= { wmanIf2SsTrapControl 2 }
12
13
14  wmanIf2SsThresholdConfigEntry OBJECT-TYPE
15      SYNTAX      WmanIf2SsThresholdConfigEntry
16      MAX-ACCESS  not-accessible
17      STATUS      current
18      DESCRIPTION
19          "This table provides one row for each Ss, and is indexed
20         by ifIndex."
21      INDEX      { ifIndex }
22      ::= { wmanIf2SsThresholdConfigTable 1 }
23
24
25
26  WmanIf2SsThresholdConfigEntry ::= SEQUENCE {
27      wmanIf2SsRssiLowThreshold      Integer32,
28      wmanIf2SsRssiHighThreshold     Integer32}
29
30
31  wmanIf2SsRssiLowThreshold OBJECT-TYPE
32      SYNTAX      Integer32
33      UNITS       "dBm"
34      MAX-ACCESS  read-write
35      STATUS      current
36      DESCRIPTION
37          "Low RSSI threshold for generating the RSSI alarm trap."
38      ::= { wmanIf2SsThresholdConfigEntry 1 }
39
40
41
42  wmanIf2SsRssiHighThreshold OBJECT-TYPE
43      SYNTAX      Integer32
44      UNITS       "dBm"
45      MAX-ACCESS  read-write
46      STATUS      current
47      DESCRIPTION
48          "High RSSI threshold for generating a trap to indicate
49         the RSSI is restored."
50      ::= { wmanIf2SsThresholdConfigEntry 2 }
51
52
53
54  wmanIf2SsTlvUnknownTrap NOTIFICATION-TYPE
55      OBJECTS     {ifIndex,
56                 wmanIf2SsMacAddress,
57                 wmanIf2SsUnknownTlv}
58      STATUS      current
59      DESCRIPTION
60          "Event that notifies detection of unknown TLV during
61         the TLV parsing process."
62      ::= { wmanIf2SsTrapPrefix 1 }
63
64
65

```

```

1  wmanIf2SsDynamicServiceFailTrap NOTIFICATION-TYPE
2      OBJECTS      {ifIndex,
3                    wmanIf2SsMacAddress,
4                    wmanIf2SsDynamicServiceType,
5                    wmanIf2SsDynamicServiceFailReason}
6
7      STATUS      current
8
9      DESCRIPTION
10         "An event to report the failure of a dynamic service
11         operation happened during the dynamic services process
12         and detected in the BS side."
13         ::= { wmanIf2SsTrapPrefix 2 }
14
15  wmanIf2SsDhcpSuccessTrap NOTIFICATION-TYPE
16      OBJECTS      {ifIndex,
17                    wmanIf2SsMacAddress}
18
19      STATUS      current
20
21      DESCRIPTION
22         "An event to report a successful Handshake to establish IP
23         connectivity."
24         ::= { wmanIf2SsTrapPrefix 3 }
25
26  wmanIf2SsRssiStatusChangeTrap NOTIFICATION-TYPE
27      OBJECTS      {ifIndex,
28                    wmanIf2SsMacAddress,
29                    wmanIf2SsRssiStatus,
30                    wmanIf2SsRssiStatusInfo}
31
32      STATUS      current
33
34      DESCRIPTION
35         "An event to report that the downlink RSSI is below
36         wmanIf2SsRssiLowThreshold, or above
37         wmanIf2SsRssiHighThreshold after restore."
38         ::= { wmanIf2SsTrapPrefix 4 }
39
40
41  wmanIf2SsNotificationObjectsTable OBJECT-TYPE
42      SYNTAX      SEQUENCE OF WmanIf2SsNotificationObjectsEntry
43      MAX-ACCESS  not-accessible
44      STATUS      current
45      DESCRIPTION
46         "This table contains SS notification objects that have been
47         reported by the trap."
48         ::= { wmanIf2SsTrapDefinitions 1 }
49
50
51  wmanIf2SsNotificationObjectsEntry OBJECT-TYPE
52      SYNTAX      WmanIf2SsNotificationObjectsEntry
53      MAX-ACCESS  not-accessible
54      STATUS      current
55      DESCRIPTION
56         "This table provides one row for each SS that has
57         generated traps, and is indexed by ifIndex."
58      INDEX      { ifIndex }
59      ::= { wmanIf2SsNotificationObjectsTable 1 }
60
61
62
63  WmanIf2SsNotificationObjectsEntry ::= SEQUENCE {
64      wmanIf2SsMacAddress      MacAddress,
65

```

```

1         wmanIf2SsUnknownTlv          OCTET STRING,
2         wmanIf2SsDynamicServiceType  INTEGER,
3         wmanIf2SsDynamicServiceFailReason  OCTET STRING,
4         wmanIf2SsRssiStatus           INTEGER,
5         wmanIf2SsRssiStatusInfo       OCTET STRING}
6
7
8     wmanIf2SsMacAddress  OBJECT-TYPE
9         SYNTAX          MacAddress
10        MAX-ACCESS      read-only
11        STATUS          current
12        DESCRIPTION
13            "The MAC address of the SS generating the trap."
14        ::= { wmanIf2SsNotificationObjectsEntry 1 }
15
16
17
18     wmanIf2SsUnknownTlv  OBJECT-TYPE
19         SYNTAX          OCTET STRING (SIZE(0..65535))
20         MAX-ACCESS      read-only
21         STATUS          current
22         DESCRIPTION
23             "Indicating the value of the unknown TLV."
24         ::= { wmanIf2SsNotificationObjectsEntry 2 }
25
26
27
28     wmanIf2SsDynamicServiceType  OBJECT-TYPE
29         SYNTAX          INTEGER {ssSfCreationReq(1),
30                             ssSfCreationRsp(2),
31                             ssSfCreationAck(3)}
32         MAX-ACCESS      read-only
33         STATUS          current
34         DESCRIPTION
35             "This object indicates the dynamic service flow
36             creation command type."
37         ::= { wmanIf2SsNotificationObjectsEntry 3 }
38
39
40
41     wmanIf2SsDynamicServiceFailReason  OBJECT-TYPE
42         SYNTAX          OCTET STRING (SIZE(0..255))
43         MAX-ACCESS      read-only
44         STATUS          current
45         DESCRIPTION
46             "This object indicates the reason why the service flow
47             creation has failed."
48         ::= { wmanIf2SsNotificationObjectsEntry 4 }
49
50
51
52     wmanIf2SsRssiStatus  OBJECT-TYPE
53         SYNTAX          INTEGER {ssRssiAlarm(1),
54                             ssRssiNoAlarm(2)}
55         MAX-ACCESS      read-only
56         STATUS          current
57         DESCRIPTION
58             "A RSSI alarm is generated if the RSSI is lower than
59             wmanIf2SsRssiLowThreshold, or above
60             wmanIf2SsRssiHighThreshold after alarm is restored."
61         ::= { wmanIf2SsNotificationObjectsEntry 5 }
62
63
64
65     wmanIf2SsRssiStatusInfo  OBJECT-TYPE

```

```

1      SYNTAX      OCTET STRING (SIZE(0..255))
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "This object provides additional information about RSSI
6          alarm. It is implementation specific"
7      ::= { wmanIf2SsNotificationObjectsEntry 6 }
8
9
10     --
11     -- Subscriber station PHY Group
12     --
13     wmanIf2SsPhy OBJECT IDENTIFIER ::= { wmanIf2SsObjects 5 }
14
15     --
16     -- SS OFDM PHY objects
17     --
18     wmanIf2SsOfdmPhy OBJECT IDENTIFIER ::= { wmanIf2SsPhy 1 }
19
20     wmanIf2SsOfdmUplinkChannelTable OBJECT-TYPE
21         SYNTAX      SEQUENCE OF WmanIf2SsOfdmUplinkChannelEntry
22         MAX-ACCESS  not-accessible
23         STATUS      current
24         DESCRIPTION
25             "This table contains UCD channel attributes, defining the
26             transmission characteristics of uplink channels"
27         REFERENCE
28             "Table 349 and Table 352, in IEEE Std 802.16-2004"
29         ::= { wmanIf2SsOfdmPhy 1 }
30
31     wmanIf2SsOfdmUplinkChannelEntry OBJECT-TYPE
32         SYNTAX      WmanIf2SsOfdmUplinkChannelEntry
33         MAX-ACCESS  not-accessible
34         STATUS      current
35         DESCRIPTION
36             "This table provides one row for each uplink channel of
37             multi-sector BS, and is indexed by BS ifIndex. An entry
38             in this table exists for each ifEntry of BS with an
39             ifType of ieee80216WMAN."
40         INDEX { ifIndex }
41         ::= { wmanIf2SsOfdmUplinkChannelTable 1 }
42
43     WmanIf2SsOfdmUplinkChannelEntry ::= SEQUENCE {
44         wmanIf2SsOfdmCtBasedResvTimeout      INTEGER,
45         wmanIf2SsOfdmBwReqOppSize           INTEGER,
46         wmanIf2SsOfdmRangReqOppSize        INTEGER,
47         wmanIf2SsOfdmUplinkCenterFreq     Unsigned32,
48         wmanIf2SsOfdmNumSubChReqRegionFull INTEGER,
49         wmanIf2SsOfdmNumSymbolsReqRegionFull INTEGER,
50         wmanIf2SsOfdmSubChFocusCtCode     INTEGER,
51         wmanIf2SsOfdmUplinkChannelId      INTEGER}
52
53     wmanIf2SsOfdmCtBasedResvTimeout OBJECT-TYPE
54         SYNTAX      INTEGER (1..255)
55         MAX-ACCESS  read-only
56
57
58
59
60
61
62
63
64
65

```



```

1      STATUS      current
2      DESCRIPTION
3          "The number of UL-MAPs to receive before contention-based
4          reservation is attempted again for the same connection."
5      REFERENCE
6          "Table 349, in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsOfdmUplinkChannelEntry 1 }
8
9
10     wmanIf2SsOfdmBwReqOppSize OBJECT-TYPE
11     SYNTAX      INTEGER (1..65535)
12     UNITS       "PS"
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "Size (in units of PS) of PHY payload that SS may use to
17         format and transmit a bandwidth request message in a
18         contention request opportunity. The value includes all
19         PHY overhead as well as allowance for the MAC data the
20         message may hold."
21     REFERENCE
22         "Table 349, in IEEE Std 802.16-2004"
23     ::= { wmanIf2SsOfdmUplinkChannelEntry 2 }
24
25
26     wmanIf2SsOfdmRangReqOppSize OBJECT-TYPE
27     SYNTAX      INTEGER (1..65535)
28     UNITS       "PS"
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32         "Size (in units of PS) of PHY payload that SS may use to
33         format and transmit a RNG-REQ message in a contention
34         request opportunity. The value includes all PHY overhead
35         as well as allowance for the MAC data the message may
36         hold and the maximum SS/BS roundtrip propagation delay."
37     REFERENCE
38         "Table 349, in IEEE Std 802.16-2004"
39     ::= { wmanIf2SsOfdmUplinkChannelEntry 3 }
40
41
42     wmanIf2SsOfdmUplinkCenterFreq OBJECT-TYPE
43     SYNTAX      Unsigned32
44     UNITS       "kHz"
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         " Uplink center frequency (kHz) "
49     REFERENCE
50         "Table 349, in IEEE Std 802.16-2004"
51     ::= { wmanIf2SsOfdmUplinkChannelEntry 4 }
52
53
54     wmanIf2SsOfdmNumSubChReqRegionFull OBJECT-TYPE
55     SYNTAX      INTEGER { oneSubchannel(0),
56                    twoSubchannels(1),
57                    fourSubchannels(2),
58                    eightSubchannels(3) }
59
60
61
62
63
64
65

```

```

1         sixteenSubchannels(4) }
2     MAX-ACCESS    read-only
3     STATUS        current
4     DESCRIPTION
5         "Number of subchannels used by each transmit
6         opportunity when REQ Region-Full is allocated in
7         subchannelization region."
8     REFERENCE
9         "Table 352, in IEEE Std 802.16-2004"
10    ::= { wmanIf2SsOfdmUplinkChannelEntry 5 }
11
12 wmanIf2SsOfdmNumSymbolsReqRegionFull OBJECT-TYPE
13     SYNTAX        INTEGER (0..31)
14     MAX-ACCESS    read-only
15     STATUS        current
16     DESCRIPTION
17         "Number of OFDM symbols used by each transmit
18         opportunity when REQ Region-Full is allocated in
19         subchannelization region."
20     REFERENCE
21         "Table 352, in IEEE Std 802.16-2004"
22    ::= { wmanIf2SsOfdmUplinkChannelEntry 6 }
23
24 wmanIf2SsOfdmSubChFocusCtCode OBJECT-TYPE
25     SYNTAX        INTEGER (0..8)
26     MAX-ACCESS    read-only
27     STATUS        current
28     DESCRIPTION
29         "Number of contention codes (CSE) that shall only be used to
30         request a subchannelized allocation. Default value 0.
31         Allowed values 0-8."
32     REFERENCE
33         "Table 352, in IEEE Std 802.16-2004"
34     DEFVAL        { 0 }
35    ::= { wmanIf2SsOfdmUplinkChannelEntry 7 }
36
37 wmanIf2SsOfdmUpLinkChannelId OBJECT-TYPE
38     SYNTAX        INTEGER (0..255)
39     MAX-ACCESS    read-only
40     STATUS        current
41     DESCRIPTION
42         "The identifier of the uplink channel to which this
43         message refers."
44     REFERENCE
45         "Subclause 6.3.2.3.4. Table 16, in IEEE Std 802.16-2004"
46    ::= { wmanIf2SsOfdmUplinkChannelEntry 8 }
47
48 wmanIf2SsOfdmDownlinkChannelTable OBJECT-TYPE
49     SYNTAX        SEQUENCE OF WmanIf2SsOfdmDownlinkChannelEntry
50     MAX-ACCESS    not-accessible
51     STATUS        current
52     DESCRIPTION
53         "This table contains DCD channel attributes, defining the
54         transmission characteristics of downlink channels"
55
56
57
58
59
60
61
62
63
64
65

```

```

1      REFERENCE
2          "Table 358, in IEEE Std 802.16-2004"
3      ::= { wmanIf2SsOfdmPhy 2 }
4
5
6      wmanIf2SsOfdmDownlinkChannelEntry OBJECT-TYPE
7          SYNTAX      WmanIf2SsOfdmDownlinkChannelEntry
8          MAX-ACCESS  not-accessible
9          STATUS      current
10         DESCRIPTION
11             "This table provides one row for each downlink channel of
12             multi-sector BS, and is indexed by BS ifIndex. An entry
13             in this table exists for each ifEntry of BS with an
14             ifType of ieee80216WMAN."
15         INDEX { ifIndex }
16     ::= { wmanIf2SsOfdmDownlinkChannelTable 1 }
17
18
19
20     WmanIf2SsOfdmDownlinkChannelEntry ::= SEQUENCE {
21         wmanIf2SsOfdmBsEIRP          INTEGER,
22         wmanIf2SsOfdmChannelNumber   WmanIf2ChannelNumber,
23         wmanIf2SsOfdmTTG             INTEGER,
24         wmanIf2SsOfdmRTG             INTEGER,
25         wmanIf2SsOfdmInitRngMaxRSS   INTEGER,
26         wmanIf2SsOfdmDownlinkCenterFreq Unsigned32,
27         wmanIf2SsOfdmBsId            WmanIf2BsIdType,
28         wmanIf2SsOfdmMacVersion       WmanIf2MacVersion,
29         wmanIf2SsOfdmFrameDurationCode INTEGER,
30         wmanIf2SsOfdmDownLinkChannelId INTEGER}
31
32
33
34
35     wmanIf2SsOfdmBsEIRP OBJECT-TYPE
36         SYNTAX      INTEGER (-32768..32767)
37         UNITS       "dBm"
38         MAX-ACCESS  read-only
39         STATUS      current
40         DESCRIPTION
41             "The EIRP is the equivalent isotropic radiated power of
42             the base station, which is computed for a simple
43             single-antenna transmitter."
44         REFERENCE
45             "Table 358, in IEEE Std 802.16-2004"
46     ::= { wmanIf2SsOfdmDownlinkChannelEntry 1 }
47
48
49
50     wmanIf2SsOfdmChannelNumber OBJECT-TYPE
51         SYNTAX      WmanIf2ChannelNumber
52         MAX-ACCESS  read-only
53         STATUS      current
54         DESCRIPTION
55             "Downlink channel number as defined in 8.5.
56             Used for license-exempt operation only."
57         REFERENCE
58             "Table 358, in IEEE Std 802.16-2004"
59     ::= { wmanIf2SsOfdmDownlinkChannelEntry 2 }
60
61
62
63     wmanIf2SsOfdmTTG OBJECT-TYPE
64         SYNTAX      INTEGER (0..255)
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "Transmit / Receive Transition Gap."
5      REFERENCE
6          "Table 358, in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsOfdmDownlinkChannelEntry 3 }
8
9
10     wmanIf2SsOfdmRTG OBJECT-TYPE
11     SYNTAX      INTEGER (0..255)
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "Receive / Transmit Transition Gap."
16     REFERENCE
17         "Table 358, in IEEE Std 802.16-2004"
18     ::= { wmanIf2SsOfdmDownlinkChannelEntry 4 }
19
20
21     wmanIf2SsOfdmInitRngMaxRSS OBJECT-TYPE
22     SYNTAX      INTEGER (-32768..32767)
23     UNITS        "dBm"
24     MAX-ACCESS  read-only
25     STATUS      current
26     DESCRIPTION
27         "Initial Ranging Max. equivalent isotropic received power
28         at BS Signed in units of 1 dBm."
29     REFERENCE
30         "Table 358, in IEEE Std 802.16-2004"
31     ::= { wmanIf2SsOfdmDownlinkChannelEntry 5 }
32
33
34     wmanIf2SsOfdmDownlinkCenterFreq OBJECT-TYPE
35     SYNTAX      Unsigned32
36     UNITS        "kHz"
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40         "Downlink center frequency (kHz)."
```

```

41     REFERENCE
42         "Table 358, in IEEE Std 802.16-2004"
43     ::= { wmanIf2SsOfdmDownlinkChannelEntry 6 }
44
45
46     wmanIf2SsOfdmBsId OBJECT-TYPE
47     SYNTAX      WmanIf2BsIdType
48     MAX-ACCESS  read-only
49     STATUS      current
50     DESCRIPTION
51         "Base station ID."
52     REFERENCE
53         "Table 358, in IEEE Std 802.16-2004"
54     ::= { wmanIf2SsOfdmDownlinkChannelEntry 7 }
55
56
57     wmanIf2SsOfdmMacVersion OBJECT-TYPE
58     SYNTAX      WmanIf2MacVersion
59     MAX-ACCESS  read-only
60
61
62
63
64
65
```

```

1      STATUS      current
2      DESCRIPTION
3          "This parameter specifies the version of 802.16 to which
4          the message originator conforms."
5      REFERENCE
6          "Table 358, in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsOfdmDownlinkChannelEntry 8 }
8
9
10     wmanIf2SsOfdmFrameDurationCode OBJECT-TYPE
11     SYNTAX      INTEGER {duration2dot5ms(0),
12                 duration4ms(1),
13                 duration5ms(2),
14                 duration8ms(3),
15                 duration10ms(4),
16                 duration12dot5ms(5),
17                 duration20ms(6)}
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21         "The duration of the frame. The frame duration code
22         values are specified in Table 230."
23     REFERENCE
24         "Subclause 11.4.1, Table 230, in IEEE Std 802.16-2004"
25     ::= { wmanIf2SsOfdmDownlinkChannelEntry 9 }
26
27
28     wmanIf2SsOfdmDownLinkChannelId OBJECT-TYPE
29     SYNTAX      INTEGER (0..255)
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33         "The identifier of the downlink channel to which this
34         message refers."
35     REFERENCE
36         "Subclause 6.3.2.3.1, Table 15, in IEEE Std 802.16-2004"
37     ::= { wmanIf2SsOfdmDownlinkChannelEntry 10 }
38
39
40     wmanIf2SsOfdmUcdBurstProfileTable OBJECT-TYPE
41     SYNTAX      SEQUENCE OF WmanIf2SsOfdmUcdBurstProfileEntry
42     MAX-ACCESS  not-accessible
43     STATUS      current
44     DESCRIPTION
45         "This table contains UCD burst profiles for each uplink
46         channel"
47     REFERENCE
48         "Table 356, in IEEE Std 802.16-2004"
49     ::= { wmanIf2SsOfdmPhy 3 }
50
51
52     wmanIf2SsOfdmUcdBurstProfileEntry OBJECT-TYPE
53     SYNTAX      WmanIf2SsOfdmUcdBurstProfileEntry
54     MAX-ACCESS  not-accessible
55     STATUS      current
56     DESCRIPTION
57         "This table provides one row for each UCD burst profile.
58         This table is double indexed. The primary index is an
59
60
61
62
63
64
65

```

```

1         ifIndex with an ifType of ieee80216WMAN. The secondary
2         index is wmanIf2SsOfdmOfdmUcdBurstProfIndex."
3     INDEX { ifIndex, wmanIf2SsOfdmUiucIndex }
4     ::= { wmanIf2SsOfdmUcdBurstProfileTable 1 }
5
6
7     WmanIf2SsOfdmUcdBurstProfileEntry ::= SEQUENCE {
8         wmanIf2SsOfdmUiucIndex          INTEGER,
9         wmanIf2SsOfdmUcdFecCodeType    WmanIf2OfdmFecCodeType,
10        wmanIf2SsOfdmFocusCtPowerBoost  INTEGER,
11        wmanIf2SsOfdmUcdTcsEnable      INTEGER}
12
13
14     wmanIf2SsOfdmUiucIndex OBJECT-TYPE
15     SYNTAX      INTEGER (5 .. 12)
16     MAX-ACCESS  not-accessible
17     STATUS      current
18     DESCRIPTION
19         "The Uplink Interval Usage Code indicates the uplink burst
20         profile in the UCD message, and is used along with ifIndex
21         to identify an entry in the
22         wmanIf2SsOfdmUcdBurstProfileTable."
23     REFERENCE
24         "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
25     ::= { wmanIf2SsOfdmUcdBurstProfileEntry 1 }
26
27
28     wmanIf2SsOfdmUcdFecCodeType OBJECT-TYPE
29     SYNTAX      WmanIf2OfdmFecCodeType
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33         "Uplink FEC code type and modulation type"
34     REFERENCE
35         "Table 356, in IEEE Std 802.16-2004"
36     ::= { wmanIf2SsOfdmUcdBurstProfileEntry 2 }
37
38
39     wmanIf2SsOfdmFocusCtPowerBoost OBJECT-TYPE
40     SYNTAX      INTEGER (0 .. 255)
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44         "The power boost in dB of focused contention carriers, as
45         described in 8.3.6.3.3."
46     REFERENCE
47         "Table 356, in IEEE Std 802.16-2004"
48     ::= { wmanIf2SsOfdmUcdBurstProfileEntry 3 }
49
50
51     wmanIf2SsOfdmUcdTcsEnable OBJECT-TYPE
52     SYNTAX      INTEGER {tcsDisabled(0),
53                 tcsEnabled(1)}
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57         "This parameter determines the transmission convergence
58         sublayer, as described in 8.1.4.3, can be enabled on a
59         per-burst basis for both uplink and downlink. through
60
61
62
63
64
65

```

```

1         DIUC/UIUC messages."
2     REFERENCE
3         "Table 356, in IEEE Std 802.16-2004"
4     ::= { wmanIf2SsOfdmUcdBurstProfileEntry 4 }
5
6
7     wmanIf2SsOfdmDcdBurstProfileTable OBJECT-TYPE
8         SYNTAX      SEQUENCE OF WmanIf2SsOfdmDcdBurstProfileEntry
9         MAX-ACCESS  not-accessible
10        STATUS      current
11        DESCRIPTION
12            "This table provides one row for each DCD burst profile.
13            This table is double indexed. The primary index is an
14            ifIndex with an ifType of ieee80216WMAN. The secondary
15            index is wmanIf2SsOfdmDiucIndex."
16        REFERENCE
17            "Table 362, in IEEE Std 802.16-2004"
18        ::= { wmanIf2SsOfdmPhy 4 }
19
20
21
22
23     wmanIf2SsOfdmDcdBurstProfileEntry OBJECT-TYPE
24         SYNTAX      WmanIf2SsOfdmDcdBurstProfileEntry
25         MAX-ACCESS  not-accessible
26         STATUS      current
27         DESCRIPTION
28            "This table provides one row for each DCD burst profile.
29            This table is double indexed. The primary index is an
30            ifIndex with an ifType of ieee80216WMAN. The secondary
31            index is wmanIf2SsOfdmDcdBurstProfIndex."
32         INDEX { ifIndex, wmanIf2SsOfdmDiucIndex }
33         ::= { wmanIf2SsOfdmDcdBurstProfileTable 1 }
34
35
36
37     WmanIf2SsOfdmDcdBurstProfileEntry ::= SEQUENCE {
38         wmanIf2SsOfdmDiucIndex          INTEGER,
39         wmanIf2SsOfdmDownlinkFrequency  Unsigned32,
40         wmanIf2SsOfdmDcdFecCodeType     WmanIf2OfdmFecCodeType,
41         wmanIf2SsOfdmDiucMandatoryExitThresh  INTEGER,
42         wmanIf2SsOfdmDiucMinEntryThresh  INTEGER,
43         wmanIf2SsOfdmTcsEnable          INTEGER}
44
45
46
47     wmanIf2SsOfdmDiucIndex OBJECT-TYPE
48         SYNTAX      INTEGER (1..11)
49         MAX-ACCESS  not-accessible
50         STATUS      current
51         DESCRIPTION
52            "The Downlink Interval Usage Code indicates the downlink
53            burst profile in the DCD message, and is used along with
54            ifIndex to identify an entry in the
55            wmanIf2SsOfdmDcdBurstProfileTable."
56         REFERENCE
57            "Subclause 8.3.6.3.1, in IEEE Std 802.16-2004"
58         ::= { wmanIf2SsOfdmDcdBurstProfileEntry 1 }
59
60
61
62     wmanIf2SsOfdmDownlinkFrequency OBJECT-TYPE
63         SYNTAX      Unsigned32
64         UNITS       "kHz"
65

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "Downlink Frequency (kHz)."

```



```

1  --
2  -- SS OFDMA PHY objects
3  --
4
5  wmanIf2SsOfdmaPhy OBJECT IDENTIFIER ::= { wmanIf2SsPhy 2 }
6
7  wmanIf2SsOfdmaUplinkChannelTable OBJECT-TYPE
8      SYNTAX      SEQUENCE OF WmanIf2SsOfdmaUplinkChannelEntry
9      MAX-ACCESS  not-accessible
10     STATUS      current
11     DESCRIPTION
12         "This table contains UCD channel attributes, defining the
13         transmission characteristics of uplink channels"
14     REFERENCE
15         "Subclause 11.3.1, Table 349 and Table 353, in IEEE Std
16         802.16-2004"
17     ::= { wmanIf2SsOfdmaPhy 1 }
18
19
20
21  wmanIf2SsOfdmaUplinkChannelEntry OBJECT-TYPE
22     SYNTAX      WmanIf2SsOfdmaUplinkChannelEntry
23     MAX-ACCESS  not-accessible
24     STATUS      current
25     DESCRIPTION
26         "This table provides one row for each uplink channel of
27         multi-sector BS, and is indexed by BS ifIndex. An entry
28         in this table exists for each ifEntry of BS with an
29         ifType of ieee80216WMAN."
30     INDEX      { ifIndex }
31     ::= { wmanIf2SsOfdmaUplinkChannelTable 1 }
32
33
34
35
36  WmanIf2SsOfdmaUplinkChannelEntry ::= SEQUENCE {
37     wmanIf2SsOfdmaCtBasedResvTimeout      INTEGER,
38     wmanIf2SsOfdmaBwReqOppSize           INTEGER,
39     wmanIf2SsOfdmaRangReqOppSize         INTEGER,
40     wmanIf2SsOfdmaUplinkCenterFreq       Unsigned32,
41     wmanIf2SsOfdmaInitRngCodes           INTEGER,
42     wmanIf2SsOfdmaPeriodicRngCodes       INTEGER,
43     wmanIf2SsOfdmaBWRngCodes             INTEGER,
44     wmanIf2SsOfdmaPerRngBackoffStart     INTEGER,
45     wmanIf2SsOfdmaPerRngBackoffEnd       INTEGER,
46     wmanIf2SsOfdmaStartOfRngCodes        INTEGER,
47     wmanIf2SsOfdmaPermutationBase        INTEGER,
48     wmanIf2SsOfdmaULAllocSubchBitmap     OCTET STRING,
49     wmanIf2SsOfdmaOptPermULAllocSubchBitmap OCTET STRING,
50     wmanIf2SsOfdmaBandAMCAallocThreshold INTEGER,
51     wmanIf2SsOfdmaBandAMCReleaseThreshold INTEGER,
52     wmanIf2SsOfdmaBandAMCAallocTimer     INTEGER,
53     wmanIf2SsOfdmaBandAMCReleaseTimer    INTEGER,
54     wmanIf2SsOfdmaBandStatRepMAXPeriod   INTEGER,
55     wmanIf2SsOfdmaBandAMCRetryTimer      INTEGER,
56     wmanIf2SsOfdmaSafetyChAllocThreshold INTEGER,
57     wmanIf2SsOfdmaSafetyChReleaseThreshold INTEGER,
58     wmanIf2SsOfdmaSafetyChAllocTimer     INTEGER,
59     wmanIf2SsOfdmaSafetyChReleaseTimer    INTEGER,
60     wmanIf2SsOfdmaBinStatRepMAXPeriod    INTEGER,
61
62
63
64
65

```

```

1      wmanIf2SsOfdmaSafetyChaRetryTimer      INTEGER,
2      wmanIf2SsOfdmaHARQAackDelayULBurst     INTEGER,
3      wmanIf2SsOfdmaCQICHBandAMCTranaDelay   INTEGER}
4
5
6      wmanIf2SsOfdmaCtBasedResvTimeout OBJECT-TYPE
7          SYNTAX      INTEGER (1..255)
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "The number of UL-MAPs to receive before contention-based
12             reservation is attempted again for the same connection."
13         REFERENCE
14             "Table 349, in IEEE Std 802.16-2004"
15         ::= { wmanIf2SsOfdmaUplinkChannelEntry 1 }
16
17
18
19      wmanIf2SsOfdmaBwReqOppSize OBJECT-TYPE
20          SYNTAX      INTEGER (1..65535)
21          UNITS       "PS"
22          MAX-ACCESS  read-only
23          STATUS      current
24          DESCRIPTION
25             "Size (in units of PS) of PHY payload that SS may use to
26             format and transmit a bandwidth request message in a
27             contention request opportunity. The value includes all
28             PHY overhead as well as allowance for the MAC data the
29             message may hold."
30         REFERENCE
31             "Table 349, in IEEE Std 802.16-2004"
32         ::= { wmanIf2SsOfdmaUplinkChannelEntry 2 }
33
34
35
36
37      wmanIf2SsOfdmaRangReqOppSize OBJECT-TYPE
38          SYNTAX      INTEGER (1..65535)
39          UNITS       "PS"
40          MAX-ACCESS  read-only
41          STATUS      current
42          DESCRIPTION
43             "Size (in units of PS) of PHY payload that SS may use to
44             format and transmit a RNG-REQ message in a contention
45             request opportunity. The value includes all PHY overhead
46             as well as allowance for the MAC data the message may
47             hold and the maximum SS/BS roundtrip propagation delay."
48         REFERENCE
49             "Table 349, in IEEE Std 802.16-2004"
50         ::= { wmanIf2SsOfdmaUplinkChannelEntry 3 }
51
52
53
54
55      wmanIf2SsOfdmaUplinkCenterFreq OBJECT-TYPE
56          SYNTAX      Unsigned32
57          UNITS       "kHz"
58          MAX-ACCESS  read-only
59          STATUS      current
60          DESCRIPTION
61             " Uplink center frequency (kHz) "
62         REFERENCE
63             "Table 349, in IEEE Std 802.16-2004"
64
65

```

```

1      ::= { wmanIf2SsOfdmaUplinkChannelEntry 4 }
2
3
4  wmanIf2SsOfdmaInitRngCodes OBJECT-TYPE
5      SYNTAX      INTEGER (0..255)
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "Number of initial ranging CDMA codes. Possible values are
10         0..255. The total number of wmanIf2SsOfdmaInitRngCodes,
11         wmanIf2SsOfdmaPeriodicRngCodes and wmanIf2SsOfdmaBWReqCodes
12         shall be equal or less than 256."
13
14      REFERENCE
15          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
16      DEFVAL      { 30 }
17      ::= { wmanIf2SsOfdmaUplinkChannelEntry 5 }
18
19
20  wmanIf2SsOfdmaPeriodicRngCodes OBJECT-TYPE
21      SYNTAX      INTEGER (0..255)
22      MAX-ACCESS  read-only
23      STATUS      current
24      DESCRIPTION
25          "Number of periodic ranging CDMA codes. Possible values are
26         0..255. The total number of wmanIf2SsOfdmaInitRngCodes,
27         wmanIf2SsOfdmaPeriodicRngCodes and wmanIf2SsOfdmaBWReqCodes
28         shall be equal or less than 256."
29
30      REFERENCE
31          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
32      DEFVAL      { 30 }
33      ::= { wmanIf2SsOfdmaUplinkChannelEntry 6 }
34
35
36
37  wmanIf2SsOfdmaBWReqCodes OBJECT-TYPE
38      SYNTAX      INTEGER (0..255)
39      MAX-ACCESS  read-only
40      STATUS      current
41      DESCRIPTION
42          "Number of bandwidth request codes. Possible values are
43         0..255. The total number of wmanIf2SsOfdmaInitRngCodes,
44         wmanIf2SsOfdmaPeriodicRngCodes and wmanIf2SsOfdmaBWReqCodes
45         shall be equal or less than 256."
46
47      REFERENCE
48          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
49      DEFVAL      { 30 }
50      ::= { wmanIf2SsOfdmaUplinkChannelEntry 7 }
51
52
53
54  wmanIf2SsOfdmaPerRngBackoffStart OBJECT-TYPE
55      SYNTAX      INTEGER (0..15)
56      MAX-ACCESS  read-only
57      STATUS      current
58      DESCRIPTION
59          "Initial backoff window size for periodic ranging
60         contention, expressed as a power of 2."
61
62      REFERENCE
63          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
64      DEFVAL      { 0 }
65

```

```

1      ::= { wmanIf2SsOfdmaUplinkChannelEntry 8 }
2
3
4  wmanIf2SsOfdmaPerRngBackoffEnd OBJECT-TYPE
5      SYNTAX      INTEGER (0 .. 15)
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "Final backoff window size for periodic ranging contention,
10         expressed as a power of 2."
11
12     REFERENCE
13         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
14     DEFVAL      { 15 }
15     ::= { wmanIf2SsOfdmaUplinkChannelEntry 9 }
16
17
18  wmanIf2SsOfdmaStartOfRngCodes OBJECT-TYPE
19     SYNTAX      INTEGER (0..255)
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "Indicates the starting number, S, of the group of codes
24         used for this uplink.All the ranging codes used on this
25         uplink will be between S and ((S+N+M+L) mod 256). Where,
26         N is the number of initial-ranging codes M is the number
27         of periodic-ranging codes L is the number of
28         bandwidth-request codes The range of values is 0 S255"
29
30     REFERENCE
31         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
32     DEFVAL      { 0 }
33     ::= { wmanIf2SsOfdmaUplinkChannelEntry 10 }
34
35
36
37  wmanIf2SsOfdmaPermutationBase OBJECT-TYPE
38     SYNTAX      INTEGER (0..255)
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "Determines the UL_IDcell parameter for the subcarrier
43         permutation to be used on this uplink channel"
44
45     REFERENCE
46         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
47     DEFVAL      { 0 }
48     ::= { wmanIf2SsOfdmaUplinkChannelEntry 11 }
49
50
51
52  wmanIf2SsOfdmaULAllocSubchBitmap OBJECT-TYPE
53     SYNTAX      OCTET STRING (SIZE (9))
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57         "This is a bitmap describing the sub-channels allocated
58         to the segment in the UL, when using the uplink PUSC
59         permutation. The LSB of the first byte shall correspond to
60         subchannel 0. For any bit that is not set,
61         the corresponding subchannel shall not be used by the SS
62         on that segment"
63
64     REFERENCE
65

```

```

1      "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
2      ::= { wmanIf2SsOfdmaUplinkChannelEntry 12 }
3
4
5  wmanIf2SsOfdmaOptPermULAllocSubchBitmap OBJECT-TYPE
6      SYNTAX      OCTET STRING (SIZE (13))
7      MAX-ACCESS  read-only
8      STATUS      current
9      DESCRIPTION
10         "This is a bitmap describing the sub-channels allocated to
11         the segment in the UL, when using the uplink optional PUSC
12         permutation (see 8.4.6.2.5 in IEEE Std 802.16-2004). The
13         LSB of the first byte shall correspond to subchannel 0.
14         For any bit that is not set, the corresponding subchannel
15         shall not be used by the SS on that segment"
16
17     REFERENCE
18         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
19     ::= { wmanIf2SsOfdmaUplinkChannelEntry 13 }
20
21
22
23  wmanIf2SsOfdmaBandAMCAllocThreshold OBJECT-TYPE
24      SYNTAX      INTEGER (0 .. 255)
25      UNITS      "dB"
26      MAX-ACCESS  read-only
27      STATUS      current
28      DESCRIPTION
29         "This object defines the OFDMA band AMC allocation
30         threshold."
31
32     REFERENCE
33         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
34     ::= { wmanIf2SsOfdmaUplinkChannelEntry 14 }
35
36
37  wmanIf2SsOfdmaBandAMCReleaseThreshold OBJECT-TYPE
38      SYNTAX      INTEGER (0 .. 255)
39      UNITS      "dB"
40      MAX-ACCESS  read-only
41      STATUS      current
42      DESCRIPTION
43         "This object defines the OFDMA band AMC release
44         threshold."
45
46     REFERENCE
47         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
48     ::= { wmanIf2SsOfdmaUplinkChannelEntry 15 }
49
50
51
52  wmanIf2SsOfdmaBandAMCAllocTimer OBJECT-TYPE
53      SYNTAX      INTEGER (0 .. 255)
54      UNITS      "Frame"
55      MAX-ACCESS  read-only
56      STATUS      current
57      DESCRIPTION
58         "This object defines the OFDMA band AMC allocation
59         timer."
60
61     REFERENCE
62         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
63     ::= { wmanIf2SsOfdmaUplinkChannelEntry 16 }
64
65

```

```

1  wmanIf2SsOfdmaBandAMCReleaseTimer OBJECT-TYPE
2      SYNTAX      INTEGER (0 .. 255)
3      UNITS       "Frame"
4      MAX-ACCESS  read-only
5      STATUS      current
6      DESCRIPTION
7          "This object defines the OFDMA band AMC release
8          timer."
9      REFERENCE
10         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
11         ::= { wmanIf2SsOfdmaUplinkChannelEntry 17 }
12
13  wmanIf2SsOfdmaBandStatRepMAXPeriod OBJECT-TYPE
14      SYNTAX      INTEGER (0 .. 255)
15      UNITS       "Frame"
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19         "This object defines the OFDMA band status reporting
20         maximum period."
21      REFERENCE
22         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
23         ::= { wmanIf2SsOfdmaUplinkChannelEntry 18 }
24
25  wmanIf2SsOfdmaBandAMCRetryTimer OBJECT-TYPE
26      SYNTAX      INTEGER (0 .. 255)
27      UNITS       "Frame"
28      MAX-ACCESS  read-only
29      STATUS      current
30      DESCRIPTION
31         "This object defines the OFDMA band AMC retry
32         timer."
33      REFERENCE
34         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
35         ::= { wmanIf2SsOfdmaUplinkChannelEntry 19 }
36
37  wmanIf2SsOfdmaSafetyChAllocThreshold OBJECT-TYPE
38      SYNTAX      INTEGER (0 .. 255)
39      UNITS       "dB"
40      MAX-ACCESS  read-only
41      STATUS      current
42      DESCRIPTION
43         "This object defines the OFDMA safety channel allocation
44         threshold."
45      REFERENCE
46         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
47         ::= { wmanIf2SsOfdmaUplinkChannelEntry 20 }
48
49  wmanIf2SsOfdmaSafetyChReleaseThreshold OBJECT-TYPE
50      SYNTAX      INTEGER (0 .. 255)
51      UNITS       "dB"
52      MAX-ACCESS  read-only
53      STATUS      current
54      DESCRIPTION
55

```

```

1         "This object defines the OFDMA safety channel release
2         threshold."
3     REFERENCE
4         "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
5     ::= { wmanIf2SsOfdmaUplinkChannelEntry 21 }
6
7
8     wmanIf2SsOfdmaSafetyChAllocTimer OBJECT-TYPE
9         SYNTAX      INTEGER (0 .. 255)
10        UNITS       "Frame"
11        MAX-ACCESS  read-only
12        STATUS      current
13        DESCRIPTION
14            "This object defines the OFDMA safety channel allocation
15            timer."
16        REFERENCE
17            "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
18        ::= { wmanIf2SsOfdmaUplinkChannelEntry 22 }
19
20
21
22
23     wmanIf2SsOfdmaSafetyChReleaseTimer OBJECT-TYPE
24        SYNTAX      INTEGER (0 .. 255)
25        UNITS       "Frame"
26        MAX-ACCESS  read-only
27        STATUS      current
28        DESCRIPTION
29            "This object defines the OFDMA safety channel release
30            timer."
31        REFERENCE
32            "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
33        ::= { wmanIf2SsOfdmaUplinkChannelEntry 23 }
34
35
36
37
38     wmanIf2SsOfdmaBinStatRepMAXPeriod OBJECT-TYPE
39        SYNTAX      INTEGER (0 .. 255)
40        UNITS       "Frame"
41        MAX-ACCESS  read-only
42        STATUS      current
43        DESCRIPTION
44            "This object defines the OFDMA bin status reporting
45            maximum period."
46        REFERENCE
47            "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
48        ::= { wmanIf2SsOfdmaUplinkChannelEntry 24 }
49
50
51
52
53     wmanIf2SsOfdmaSafetyChaRetryTimer OBJECT-TYPE
54        SYNTAX      INTEGER (0 .. 255)
55        UNITS       "Frame"
56        MAX-ACCESS  read-only
57        STATUS      current
58        DESCRIPTION
59            "This object defines the OFDMA safety channel retry
60            timer."
61        REFERENCE
62            "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
63        ::= { wmanIf2SsOfdmaUplinkChannelEntry 25 }
64
65

```

```

1  wmanIf2SsOfdmaHARQAackDelayULBurst OBJECT-TYPE
2      SYNTAX          INTEGER {oneframeoffset(1),
3                          twoframesoffset(2),
4                          threeframesoffset(3)}
5
6      MAX-ACCESS      read-only
7      STATUS          current
8      DESCRIPTION
9          "This object defines the OFDMA H-ARQ ACK delay for UL burst.
10         1 = one frame offset
11         2 = two frames offset
12         3 = three frames offset"
13
14      REFERENCE
15          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
16          ::= { wmanIf2SsOfdmaUplinkChannelEntry 26 }
17
18
19  wmanIf2SsOfdmaCQICHBandAMCTranaDelay OBJECT-TYPE
20      SYNTAX          INTEGER (0 .. 255)
21      UNITS           "Frame"
22      MAX-ACCESS      read-only
23      STATUS          current
24      DESCRIPTION
25          "This object defines the OFDMA CQICH band AMC transition
26          delay."
27
28      REFERENCE
29          "Subclause 11.3.1, Table 353, in IEEE Std 802.16-2004"
30          ::= { wmanIf2SsOfdmaUplinkChannelEntry 27 }
31
32
33  wmanIf2SsOfdmaDownlinkChannelTable OBJECT-TYPE
34      SYNTAX          SEQUENCE OF WmanIf2SsOfdmaDownlinkChannelEntry
35      MAX-ACCESS      not-accessible
36      STATUS          current
37      DESCRIPTION
38          "This table contains DCD channel attributes, defining the
39          transmission characteristics of downlink channels"
40
41      REFERENCE
42          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
43          ::= { wmanIf2SsOfdmaPhy 2 }
44
45
46  wmanIf2SsOfdmaDownlinkChannelEntry OBJECT-TYPE
47      SYNTAX          WmanIf2SsOfdmaDownlinkChannelEntry
48      MAX-ACCESS      not-accessible
49      STATUS          current
50      DESCRIPTION
51          "This table provides one row for each downlink channel of
52          multi-sector BS, and is indexed by BS ifIndex. An entry in
53          this table exists for each ifEntry of BS with an ifType of
54          ieee80216WMAN."
55
56      INDEX           { ifIndex }
57      ::= { wmanIf2SsOfdmaDownlinkChannelTable 1 }
58
59
60  WmanIf2SsOfdmaDownlinkChannelEntry ::= SEQUENCE {
61      wmanIf2SsOfdmaBsEIRP          INTEGER,
62      wmanIf2SsOfdmaChannelNumber   WmanIf2ChannelNumber,
63      wmanIf2SsOfdmaTTG             INTEGER,
64
65

```



```

1      wmanIf2SsOfdmaRTG                INTEGER,
2      wmanIf2SsOfdmaInitRngMaxRSS     INTEGER,
3      wmanIf2SsOfdmaDownlinkCenterFreq Unsigned32,
4      wmanIf2SsOfdmaBsId              WmanIf2BsIdType,
5      wmanIf2SsOfdmaMacVersion        WmanIf2MacVersion,
6      wmanIf2SsOfdmaFrameDurationCode INTEGER,
7      wmanIf2SsOfdmaSizeCqichIdField  INTEGER,
8      wmanIf2SsOfdmaHARQAackDelayBurst INTEGER}
9
10
11
12  wmanIf2SsOfdmaBsEIRP OBJECT-TYPE
13      SYNTAX      INTEGER (-32768..32767)
14      UNITS       "dBm"
15      MAX-ACCESS  read-only
16      STATUS      current
17      DESCRIPTION
18          "The EIRP is the equivalent isotropic radiated power of
19           the base station, which is computed for a simple
20           single-antenna transmitter."
21      REFERENCE
22          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
23      ::= { wmanIf2SsOfdmaDownlinkChannelEntry 1 }
24
25
26
27  wmanIf2SsOfdmaChannelNumber OBJECT-TYPE
28      SYNTAX      WmanIf2ChannelNumber
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "Downlink channel number as defined in 8.5. Used for
33           license-exempt operation only."
34      REFERENCE
35          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
36      ::= { wmanIf2SsOfdmaDownlinkChannelEntry 2 }
37
38
39
40
41  wmanIf2SsOfdmaTTG OBJECT-TYPE
42      SYNTAX      INTEGER (0..255)
43      MAX-ACCESS  read-only
44      STATUS      current
45      DESCRIPTION
46          "Transmit / Receive Transition Gap."
47      REFERENCE
48          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
49      ::= { wmanIf2SsOfdmaDownlinkChannelEntry 3 }
50
51
52
53  wmanIf2SsOfdmaRTG OBJECT-TYPE
54      SYNTAX      INTEGER (0..255)
55      MAX-ACCESS  read-only
56      STATUS      current
57      DESCRIPTION
58          "Receive / Transmit Transition Gap."
59      REFERENCE
60          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
61      ::= { wmanIf2SsOfdmaDownlinkChannelEntry 4 }
62
63
64
65  wmanIf2SsOfdmaInitRngMaxRSS OBJECT-TYPE

```

```

1      SYNTAX      INTEGER (-32768..32767)
2      UNITS       "dBm"
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "Initial Ranging Max. equivalent isotropic received power
7          at BS Signed in units of 1 dBm."
8      REFERENCE
9          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
10     ::= { wmanIf2SsOfdmaDownlinkChannelEntry 5 }
11
12
13
14     wmanIf2SsOfdmaDownlinkCenterFreq OBJECT-TYPE
15         SYNTAX      Unsigned32
16         UNITS       "kHz"
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "Downlink center frequency (kHz)."

```

```

1      STATUS      current
2      DESCRIPTION
3          "The duration of the frame. The frame duration code values
4          are specified in Table 232 in IEEE Std 802.16-2004."
5      REFERENCE
6          "Subclause 11.4.1, Table 358, in IEEE Std 802.16-2004"
7      ::= { wmanIf2SsOfdmaDownlinkChannelEntry 9 }
8
9
10
11 wmanIf2SsOfdmaSizeCqichIdField OBJECT-TYPE
12     SYNTAX      INTEGER {threebits(1),
13                 fourbits(2),
14                 fivebits(3),
15                 sixbits(4),
16                 sevenbits(5),
17                 eightbits(6),
18                 ninebits(7)}
19
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "This object defines the size of CQICH ID field.
24         0 = Reserved
25         1 = 3 bits
26         2 = 4 bits
27         3 = 5 bits
28         4 = 6 bits
29         5 = 7 bits
30         6 = 8 bits
31         7 = 9 bits
32         8...255 = Reserved"
33     REFERENCE
34         "Subclause 11.3.1, Table 358, in IEEE Std 802.16-2004"
35     ::= { wmanIf2SsOfdmaDownlinkChannelEntry 10 }
36
37
38
39
40
41 wmanIf2SsOfdmaHARQAackDelayBurst OBJECT-TYPE
42     SYNTAX      INTEGER {oneframeoffset(1),
43                 twoframesoffset(2),
44                 threeframesoffset(3)}
45
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "This object defines the OFDMA H-ARQ ACK delay for DL burst.
50         1 = one frame offset
51         2 = two frames offset
52         3 = three frames offset"
53     REFERENCE
54         "Subclause 11.3.1, Table 358, in IEEE Std 802.16-2004"
55     ::= { wmanIf2SsOfdmaDownlinkChannelEntry 11 }
56
57
58
59 wmanIf2SsOfdmaUcdBurstProfileTable OBJECT-TYPE
60     SYNTAX      SEQUENCE OF WmanIf2SsOfdmaUcdBurstProfileEntry
61     MAX-ACCESS  not-accessible
62     STATUS      current
63     DESCRIPTION
64         "This table contains UCD burst profiles for each uplink
65

```

```

1         channel"
2     REFERENCE
3         "Subclause 11.3.1.1, Table 288 and Table 357, in IEEE
4         Std 802.16-2004"
5     ::= { wmanIf2SsOfdmaPhy 3 }
6
7
8     wmanIf2SsOfdmaUcdBurstProfileEntry OBJECT-TYPE
9         SYNTAX      WmanIf2SsOfdmaUcdBurstProfileEntry
10        MAX-ACCESS  not-accessible
11        STATUS      current
12        DESCRIPTION
13            "This table provides one row for each UCD burst profile.
14            This table is double indexed. The primary index is an
15            ifIndex with an ifType of ieee80216WMAN. The secondary
16            index is wmanIf2SsOfdmaUiucIndex."
17        INDEX       { ifIndex, wmanIf2SsOfdmaUiucIndex }
18        ::= { wmanIf2SsOfdmaUcdBurstProfileTable 1 }
19
20
21
22
23     WmanIf2SsOfdmaUcdBurstProfileEntry ::= SEQUENCE {
24         wmanIf2SsOfdmaUiucIndex          INTEGER,
25         wmanIf2SsOfdmaUcdFecCodeType    WmanIf2OfdmaFecCodeType,
26         wmanIf2SsOfdmaRangingDataRatio  INTEGER,
27         wmanIf2SsOfdmaNorCOverNOverride OCTET STRING}
28
29
30     wmanIf2SsOfdmaUiucIndex OBJECT-TYPE
31         SYNTAX      INTEGER (1 .. 10)
32         MAX-ACCESS  read-only
33         STATUS      current
34         DESCRIPTION
35             "The Uplink Interval Usage Code indicates the uplink burst
36             profile in the UCD message, and is used along with ifIndex
37             to identify an entry in the
38             wmanIf2SsOfdmaUcdBurstProfileTable."
39         REFERENCE
40             "Subclause 8.4.5.4.1, in IEEE Std 802.16-2004"
41         ::= { wmanIf2SsOfdmaUcdBurstProfileEntry 1 }
42
43
44
45     wmanIf2SsOfdmaUcdFecCodeType OBJECT-TYPE
46         SYNTAX      WmanIf2OfdmaFecCodeType
47         MAX-ACCESS  read-only
48         STATUS      current
49         DESCRIPTION
50             "Uplink FEC code type and modulation type"
51         REFERENCE
52             "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
53         ::= { wmanIf2SsOfdmaUcdBurstProfileEntry 2 }
54
55
56
57     wmanIf2SsOfdmaRangingDataRatio OBJECT-TYPE
58         SYNTAX      INTEGER (0 .. 255)
59         MAX-ACCESS  read-only
60         STATUS      current
61         DESCRIPTION
62             "Reducing factor in units of 1 dB, between the power used
63             for this burst and power should be used for CDMA Ranging."
64
65

```

```

1      REFERENCE
2      "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
3      ::= { wmanIf2SsOfdmaUcdBurstProfileEntry 3 }
4
5
6      wmanIf2SsOfdmaNorCOverNOVERRIDE OBJECT-TYPE
7      SYNTAX OCTET STRING (SIZE (5))
8      MAX-ACCESS read-only
9      STATUS current
10     DESCRIPTION
11         "This is a list of numbers, where each number is encoded by
12         one nibble, and interpreted as a signed integer. The nibbles
13         correspond in order to the list define by Table 334 in IEEE
14         Std 802.16-2004 starting from the second line, such that
15         the LS nibble of the first byte corresponds to the second
16         line in the table. The number encoded by each nibble
17         represents the difference in normalized C/N relative to the
18         previous line in the table"
19     REFERENCE
20         "Subclause 11.3.1.1, Table 357, in IEEE Std 802.16-2004"
21     ::= { wmanIf2SsOfdmaUcdBurstProfileEntry 4 }
22
23
24     wmanIf2SsOfdmaDcdBurstProfileTable OBJECT-TYPE
25     SYNTAX SEQUENCE OF WmanIf2SsOfdmaDcdBurstProfileEntry
26     MAX-ACCESS not-accessible
27     STATUS current
28     DESCRIPTION
29         "This table provides one row for each DCD burst profile.
30         This table is double indexed. The primary index is an
31         ifIndex with an ifType of ieee80216WMAN. The secondary
32         index is wmanIf2SsOfdmaDiucIndex."
33     ::= { wmanIf2SsOfdmaPhy 4 }
34
35
36     wmanIf2SsOfdmaDcdBurstProfileEntry OBJECT-TYPE
37     SYNTAX WmanIf2SsOfdmaDcdBurstProfileEntry
38     MAX-ACCESS not-accessible
39     STATUS current
40     DESCRIPTION
41         "This table provides one row for each DCD burst profile,
42         and is double indexed. The primary index is an ifIndex
43         with an ifType of ieee80216WMAN. The secondary index is
44         wmanIf2SsOfdmaDiucIndex."
45     INDEX { ifIndex, wmanIf2SsOfdmaDiucIndex }
46     ::= { wmanIf2SsOfdmaDcdBurstProfileTable 1 }
47
48
49     WmanIf2SsOfdmaDcdBurstProfileEntry ::= SEQUENCE {
50         wmanIf2SsOfdmaDiucIndex INTEGER,
51         wmanIf2SsOfdmaDownlinkFrequency Unsigned32,
52         wmanIf2SsOfdmaDcdFecCodeType WmanIf2OfdmaFecCodeType,
53         wmanIf2SsOfdmaDiucMandatoryExitThresh INTEGER,
54         wmanIf2SsOfdmaDiucMinEntryThresh INTEGER}
55
56
57     wmanIf2SsOfdmaDiucIndex OBJECT-TYPE
58     SYNTAX INTEGER (0 .. 12)
59     MAX-ACCESS read-only
60
61
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "The Downlink Interval Usage Code indicates the downlink
4          burst profile in the DCD message, and is used
5          along with ifIndex to identify an entry in the
6          wmanIf2SsOfdmaDcdBurstProfileTable."
7      REFERENCE
8          "Subclause 8.4.5.3.1, in IEEE Std 802.16-2004"
9      ::= { wmanIf2SsOfdmaDcdBurstProfileEntry 1 }
10
11
12
13      wmanIf2SsOfdmaDownlinkFrequency OBJECT-TYPE
14          SYNTAX      Unsigned32
15          UNITS        "kHz"
16          MAX-ACCESS  read-only
17          STATUS      current
18          DESCRIPTION
19              "Downlink Frequency (kHz)."

```

```

1
2
3  --
4  -- Common object group - containing common tables and objects to be
5  -- implemented in both Base Station and Subscriber Station
6  --
7  -- wmanIf2CmnPacketCs contain the Packet Convergence Sublayer objects
8  -- that are common to both Base Station and Subscriber Station
9  --
10
11 wmanIf2CmnPacketCs OBJECT IDENTIFIER ::= { wmanIf2CommonObjects 1 }
12
13 wmanIf2CmnClassifierRuleTable OBJECT-TYPE
14     SYNTAX      SEQUENCE OF WmanIf2CmnClassifierRuleEntry
15     MAX-ACCESS  not-accessible
16     STATUS      current
17     DESCRIPTION
18         "This table contains packet classifier rules associated
19         with service flows."
20     ::= { wmanIf2CmnPacketCs 1 }
21
22
23
24 wmanIf2CmnClassifierRuleEntry OBJECT-TYPE
25     SYNTAX      WmanIf2CmnClassifierRuleEntry
26     MAX-ACCESS  not-accessible
27     STATUS      current
28     DESCRIPTION
29         "This table provides one row for each packet classifier
30         rule, and is indexed by ifIndex, wmanIf2CmnCpsSfId, and
31         wmanIf2CmnClassifierRuleIndex. ifIndex is associated with
32         the BS sector. wmanIf2CmnCpsSfId identifies the service
33         flow, and wmanIf2CmnClassifierRuleIndex identifies the
34         packet classifier rule."
35     INDEX { ifIndex, wmanIf2CmnCpsSfId,
36             wmanIf2CmnClassifierRuleIndex }
37     ::= { wmanIf2CmnClassifierRuleTable 1 }
38
39
40
41
42 WmanIf2CmnClassifierRuleEntry ::= SEQUENCE {
43     wmanIf2CmnClassifierRuleIndex      Unsigned32,
44     wmanIf2CmnClassifierRulePriority    INTEGER,
45     wmanIf2CmnClassifierRuleIpTosLow   INTEGER,
46     wmanIf2CmnClassifierRuleIpTosHigh  INTEGER,
47     wmanIf2CmnClassifierRuleIpTosMask  INTEGER,
48     wmanIf2CmnClassifierRuleIpProtocol Integer32,
49     wmanIf2CmnClassifierRuleIpSourceAddr InetAddress,
50     wmanIf2CmnClassifierRuleIpSourceMask InetAddress,
51     wmanIf2CmnClassifierRuleIpDestAddr  InetAddress,
52     wmanIf2CmnClassifierRuleIpDestMask  InetAddress,
53     wmanIf2CmnClassifierRuleSourcePortStart Integer32,
54     wmanIf2CmnClassifierRuleSourcePortEnd Integer32,
55     wmanIf2CmnClassifierRuleDestPortStart Integer32,
56     wmanIf2CmnClassifierRuleDestPortEnd Integer32,
57     wmanIf2CmnClassifierRuleDestMacAddr MacAddress,
58     wmanIf2CmnClassifierRuleDestMacMask MacAddress,
59     wmanIf2CmnClassifierRuleSourceMacAddr MacAddress,
60     wmanIf2CmnClassifierRuleSourceMacMask MacAddress,
61     wmanIf2CmnClassifierRuleEnetProtocolType INTEGER,
62
63
64
65

```

```

1      wmanIf2CmnClassifierRuleEnetProtocol      Integer32,
2      wmanIf2CmnClassifierRuleUserPriLow      Integer32,
3      wmanIf2CmnClassifierRuleUserPriHigh    Integer32,
4      wmanIf2CmnClassifierRuleVlanId        Integer32,
5      wmanIf2CmnClassifierRuleState          INTEGER,
6      wmanIf2CmnClassifierRulePkts          Counter64,
7      wmanIf2CmnClassifierRuleIpv6FlowLabel  WmanIf2Ipv6FlowLabel,
8      wmanIf2CmnClassifierRuleBitMap        WmanIf2ClassifierBitMap
9
10     }
11
12
13     wmanIf2CmnClassifierRuleIndex OBJECT-TYPE
14         SYNTAX      Unsigned32 (1..4294967295)
15         MAX-ACCESS  not-accessible
16         STATUS      current
17         DESCRIPTION
18             "An index is assigned to each classifier in the classifiers
19             table"
20         ::= { wmanIf2CmnClassifierRuleEntry 1 }
21
22
23
24     wmanIf2CmnClassifierRulePriority OBJECT-TYPE
25         SYNTAX      INTEGER (0..255)
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "The value specifies the order of evaluation of the
30             classifiers. The higher the value the higher the
31             priority. The value of 0 is used as default in
32             provisioned service flows classifiers. The default
33             value of 64 is used for dynamic service flow classifiers.
34             If the referenced parameter is not present in a classifier
35             , this object reports the default value as defined above"
36         REFERENCE
37             "Subclause 11.13.19.3.4.1 in IEEE Std 802.16-2004"
38         DEFVAL      { 0 }
39         ::= { wmanIf2CmnClassifierRuleEntry 2 }
40
41
42
43
44     wmanIf2CmnClassifierRuleIpTosLow OBJECT-TYPE
45         SYNTAX      INTEGER (0 .. 255)
46         MAX-ACCESS  read-only
47         STATUS      current
48         DESCRIPTION
49             "The low value of a range of TOS byte values. If the
50             referenced parameter is not present in a classifier, this
51             object reports the value of 0."
52         REFERENCE
53             "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
54         ::= { wmanIf2CmnClassifierRuleEntry 3 }
55
56
57
58
59     wmanIf2CmnClassifierRuleIpTosHigh OBJECT-TYPE
60         SYNTAX      INTEGER (0 .. 255)
61         MAX-ACCESS  read-only
62         STATUS      current
63         DESCRIPTION
64             "The 8-bit high value of a range of TOS byte values.
65

```



```

1         If the referenced parameter is not present in a classifier
2         , this object reports the value of 0."
3
4     REFERENCE
5         "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
6     ::= { wmanIf2CmnClassifierRuleEntry 4 }
7
8 wmanIf2CmnClassifierRuleIpTosMask OBJECT-TYPE
9     SYNTAX      INTEGER (0 .. 255)
10    MAX-ACCESS  read-only
11    STATUS      current
12    DESCRIPTION
13        "The mask value is bitwise ANDed with TOS byte in an IP
14        packet and this value is used for the range checking of
15        TosLow and TosHigh. If the referenced parameter is not
16        present in a classifier, this object reports the value
17        of 0."
18    REFERENCE
19        "Subclause 11.13.19.3.4.2 in IEEE Std 802.16-2004"
20    ::= { wmanIf2CmnClassifierRuleEntry 5 }
21
22 wmanIf2CmnClassifierRuleIpProtocol OBJECT-TYPE
23     SYNTAX      Integer32 (0..255)
24     MAX-ACCESS  read-only
25     STATUS      current
26     DESCRIPTION
27         "This object indicates the value of the IP Protocol field
28         required for IP packets to match this rule. If the
29         referenced parameter is not present in a classifier, this
30         object reports the value of 0."
31     REFERENCE
32         "Subclause 11.13.19.3.4.3 in IEEE Std 802.16-2004"
33     ::= { wmanIf2CmnClassifierRuleEntry 6 }
34
35 wmanIf2CmnClassifierRuleIpSourceAddr OBJECT-TYPE
36     SYNTAX      InetAddress
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40         "This object specifies the value of the IP Source Address
41         required for packets to match this rule. An IP packet
42         matches the rule when the packet ip source address bitwise
43         ANDed with the wmanIf2CmnClassifierRuleIpSourceMask value
44         equals the wmanIf2CmnClassifierRuleIpSourceAddr value.
45         If the referenced parameter is not present in a classifier
46         , this object reports the value of 0.0.0.0."
47     REFERENCE
48         "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
49     ::= { wmanIf2CmnClassifierRuleEntry 7 }
50
51 wmanIf2CmnClassifierRuleIpSourceMask OBJECT-TYPE
52     SYNTAX      InetAddress
53     MAX-ACCESS  read-only
54     STATUS      current
55     DESCRIPTION

```

```

1      "This object specifies which bits of a packet's IP Source
2      Address that are compared to match this rule. An IP packet
3      matches the rule when the packet source address bitwise
4      ANDED with the
5      wmanIf2CmnClassifierRuleIpSourceMask value equals the
6      wmanIf2CmnClassifierRuleIpSourceAddr value.
7      If the referenced parameter is not present in a classifier
8      , this object reports the value of 0.0.0.0."
9
10     REFERENCE
11     "Subclause 11.13.19.3.4.4 in IEEE Std 802.16-2004"
12     ::= { wmanIf2CmnClassifierRuleEntry 8 }
13
14
15     wmanIf2CmnClassifierRuleIpDestAddr OBJECT-TYPE
16     SYNTAX      InetAddress
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20     "This object specifies the value of the IP Destination
21     Address required for packets to match this rule. An IP
22     packet matches the rule when the packet IP destination
23     address bitwise ANDED with the
24     wmanIf2CmnClassifierRuleIpDestMask value equals the
25     wmanIf2CmnClassifierRuleIpDestAddr value.
26     If the referenced parameter is not present in a
27     classifier, this object reports the value of 0.0.0.0."
28
29     REFERENCE
30     "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
31     ::= { wmanIf2CmnClassifierRuleEntry 9 }
32
33
34
35     wmanIf2CmnClassifierRuleIpDestMask OBJECT-TYPE
36     SYNTAX      InetAddress
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40     "This object specifies which bits of a packet's IP
41     Destination Address that are compared to match this rule.
42     An IP packet matches the rule when the packet destination
43     address bitwise ANDED with the
44     wmanIf2CmnClassifierRuleIpDestMask value equals the
45     wmanIf2CmnClassifierRuleIpDestAddr value.
46     If the referenced parameter is not present in a classifier
47     , this object reports the value of 0.0.0.0."
48
49     REFERENCE
50     "Subclause 11.13.19.3.4.5 in IEEE Std 802.16-2004"
51     ::= { wmanIf2CmnClassifierRuleEntry 10 }
52
53
54
55
56     wmanIf2CmnClassifierRuleSourcePortStart OBJECT-TYPE
57     SYNTAX      Integer32 (0..65535)
58     MAX-ACCESS  read-only
59     STATUS      current
60     DESCRIPTION
61     "This object specifies the low end inclusive range of
62     TCP/UDP source port numbers to which a packet is compared
63     . This object is irrelevant for non-TCP/UDP IP packets.
64
65

```

```

1         If the referenced parameter is not present in a
2         classifier, this object reports the value of 0."
3     REFERENCE
4         "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
5     ::= { wmanIf2CmnClassifierRuleEntry 11 }
6
7
8     wmanIf2CmnClassifierRuleSourcePortEnd OBJECT-TYPE
9     SYNTAX      Integer32 (0..65535)
10    MAX-ACCESS  read-only
11    STATUS      current
12    DESCRIPTION
13        "This object specifies the high end inclusive range of
14        TCP/UDP source port numbers to which a packet is compared.
15        This object is irrelevant for non-TCP/UDP IP packets.
16        If the referenced parameter is not present in a classifier,
17        this object reports the value of 65535."
18    REFERENCE
19        "Subclause 11.13.19.3.4.6 in IEEE Std 802.16-2004"
20    ::= { wmanIf2CmnClassifierRuleEntry 12 }
21
22
23    wmanIf2CmnClassifierRuleDestPortStart OBJECT-TYPE
24    SYNTAX      Integer32 (0..65535)
25    MAX-ACCESS  read-only
26    STATUS      current
27    DESCRIPTION
28        "This object specifies the low end inclusive range of
29        TCP/UDP destination port numbers to which a packet is
30        compared. If the referenced parameter is not present
31        in a classifier, this object reports the value of 0."
32    REFERENCE
33        "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
34    ::= { wmanIf2CmnClassifierRuleEntry 13 }
35
36
37    wmanIf2CmnClassifierRuleDestPortEnd OBJECT-TYPE
38    SYNTAX      Integer32 (0..65535)
39    MAX-ACCESS  read-only
40    STATUS      current
41    DESCRIPTION
42        "This object specifies the high end inclusive range of
43        TCP/UDP destination port numbers to which a packet is
44        compared. If the referenced parameter is not present
45        in a classifier, this object reports the value of
46        65535."
47    REFERENCE
48        "Subclause 11.13.19.3.4.7 in IEEE Std 802.16-2004"
49    ::= { wmanIf2CmnClassifierRuleEntry 14 }
50
51
52    wmanIf2CmnClassifierRuleDestMacAddr OBJECT-TYPE
53    SYNTAX      MacAddress
54    MAX-ACCESS  read-only
55    STATUS      current
56    DESCRIPTION
57        "An Ethernet packet matches an entry when its destination
58        MAC address bitwise ANDed with
59
60
61
62
63
64
65

```

```

1         wmanIf2CmnClassifierRuleDestMacMask equals the value of
2         wmanIf2CmnClassifierRuleDestMacAddr. If the referenced
3         parameter is not present in a classifier, this object
4         reports the value of '000000000000'H."
5
6     REFERENCE
7         "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
8     ::= { wmanIf2CmnClassifierRuleEntry 15 }
9
10
11 wmanIf2CmnClassifierRuleDestMacMask OBJECT-TYPE
12     SYNTAX      MacAddress
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "An Ethernet packet matches an entry when its destination
17         MAC address bitwise ANDed with
18         wmanIf2CmnClassifierRuleDestMacMask equals the value of
19         wmanIf2CmnClassifierRuleDestMacAddr. If the referenced
20         parameter is not present in a classifier, this object
21         reports the value of '000000000000'H."
22
23     REFERENCE
24         "Subclause 11.13.19.3.4.8 in IEEE Std 802.16-2004"
25     ::= { wmanIf2CmnClassifierRuleEntry 16 }
26
27
28
29 wmanIf2CmnClassifierRuleSourceMacAddr OBJECT-TYPE
30     SYNTAX      MacAddress
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "An Ethernet packet matches this entry when its source
35         MAC address bitwise ANDed with
36         wmanIf2CmnClassifierRuleSourceMacMask equals the value
37         of wmanIf2CmnClassifierRuleSourceMacAddr. If the
38         referenced parameter is not present in a classifier,
39         this object reports the value of '000000000000'H."
40
41     REFERENCE
42         "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
43     ::= { wmanIf2CmnClassifierRuleEntry 17 }
44
45
46
47 wmanIf2CmnClassifierRuleSourceMacMask OBJECT-TYPE
48     SYNTAX      MacAddress
49     MAX-ACCESS  read-only
50     STATUS      current
51     DESCRIPTION
52         "An Ethernet packet matches an entry when its destination
53         MAC address bitwise ANDed with
54         wmanIf2CmnClassifierRuleSourceMacMask equals the value of
55         wmanIf2CmnClassifierRuleSourceMacAddr. If the referenced
56         parameter is not present in a classifier, this object
57         reports the value of '000000000000'H."
58
59     REFERENCE
60         "Subclause 11.13.19.3.4.9 in IEEE Std 802.16-2004"
61     ::= { wmanIf2CmnClassifierRuleEntry 18 }
62
63
64
65 wmanIf2CmnClassifierRuleEnetProtocolType OBJECT-TYPE

```

```

1      SYNTAX      INTEGER {none(0),
2                      ethertype(1),
3                      dsap(2)}
4
5      MAX-ACCESS  read-only
6      STATUS      current
7      DESCRIPTION
8          "This object indicates the format of the layer 3 protocol
9          id in the Ethernet packet. A value of none(0) means that
10         the rule does not use the layer 3 protocol type as a
11         matching criteria. A value of ethertype(1) means that the
12         rule applies only to frames which contains an EtherType
13         value. Ethertype values are contained in packets using
14         the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
15         Sub-Network Access Protocol (SNAP) encapsulation formats.
16         A value of dsap(2) means that the rule applies only to
17         frames using the IEEE802.3 encapsulation format with a
18         Destination Service Access Point (DSAP) other than 0xAA
19         (which is reserved for SNAP). If the Ethernet frame
20         contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
21         this object applies to the embedded EtherType field within
22         the 802.1P/Q header. If the referenced parameter is not
23         present in a classifier, this object reports the value of
24         0."
25
26      REFERENCE
27          "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"
28          ::= { wmanIf2CmnClassifierRuleEntry 19 }
29
30      wmanIf2CmnClassifierRuleEnetProtocol OBJECT-TYPE
31      SYNTAX      Integer32 (0..65535)
32      MAX-ACCESS  read-only
33      STATUS      current
34      DESCRIPTION
35          "If wmanIf2CmnClassifierRuleEnetProtocolType is none(0),
36          this object is ignored when considering whether a packet
37          matches the current rule.
38          If wmanIf2CmnClassifierRuleEnetProtocolType is ethertype(1)
39          ,this object gives the 16-bit value of the EtherType that
40          the packet must match in order to match the rule.
41          If wmanIf2CmnClassifierRuleEnetProtocolType is dsap(2), the
42          lower 8 bits of this object's value must match the DSAP
43          byte of the packet in order to match the rule.
44          If the Ethernet frame contains an 802.1P/Q Tag header
45          (i.e. EtherType 0x8100), this object applies to the
46          embedded EtherType field within the 802.1P/Q header.
47          If the referenced parameter is not present in the
48          classifier, the value of this object is reported as 0."
49
50      REFERENCE
51          "Subclause 11.13.19.3.4.10 in IEEE Std 802.16-2004"
52          ::= { wmanIf2CmnClassifierRuleEntry 20 }
53
54      wmanIf2CmnClassifierRuleUserPriLow OBJECT-TYPE
55      SYNTAX      Integer32 (0..7)
56      MAX-ACCESS  read-only
57      STATUS      current
58
59
60
61
62
63
64
65

```

```

1      DESCRIPTION
2          "This object applies only to Ethernet frames using the
3          802.1P/Q tag header (indicated with EtherType 0x8100).
4          Such frames include a 16-bit Tag that contains a 3 bit
5          Priority field and a 12 bit VLAN number.
6          Tagged Ethernet packets must have a 3-bit Priority field
7          within the range of wmanIf2CmnClassifierRulePriLow and
8          wmanIf2CmnClassifierRulePriHigh in order to match this
9          rule.
10         If the referenced parameter is not present in the
11         classifier, the value of this object is reported as 0."
12
13     REFERENCE
14         "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
15         ::= { wmanIf2CmnClassifierRuleEntry 21 }
16
17 wmanIf2CmnClassifierRuleUserPriHigh OBJECT-TYPE
18     SYNTAX      Integer32 (0..7)
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "This object applies only to Ethernet frames using the
23         802.1P/Q tag header (indicated with EtherType 0x8100).
24         Such frames include a 16-bit Tag that contains a 3 bit
25         Priority field and a 12 bit VLAN number.
26         Tagged Ethernet packets must have a 3-bit Priority
27         field within the range of wmanIf2CmnClassifierRulePriLow
28         and wmanIf2CmnClassifierRulePriHigh in order to match
29         this rule.
30         If the referenced parameter is not present in the
31         classifier, the value of this object is reported as 7."
32
33     REFERENCE
34         "Subclause 11.13.19.3.4.11 in IEEE Std 802.16-2004"
35         ::= { wmanIf2CmnClassifierRuleEntry 22 }
36
37 wmanIf2CmnClassifierRuleVlanId OBJECT-TYPE
38     SYNTAX      Integer32 (0..4095)
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "This object applies only to Ethernet frames using the
43         802.1P/Q tag header.
44         If this object's value is nonzero, tagged packets must
45         have a VLAN Identifier that matches the value in order
46         to match the rule.
47         Only the least significant 12 bits of this object's
48         value are valid.
49         If the referenced parameter is not present in the
50         classifier, the value of this object is reported as 0."
51
52     REFERENCE
53         "Subclause 11.13.19.3.4.12 in IEEE Std 802.16-2004"
54         ::= { wmanIf2CmnClassifierRuleEntry 23 }
55
56 wmanIf2CmnClassifierRuleState OBJECT-TYPE
57     SYNTAX      INTEGER {active(1),
58
59
60
61
62
63
64
65

```

```

1           inactive(2) }
2
3   MAX-ACCESS  read-only
4   STATUS      deprecated
5   DESCRIPTION
6       "This object indicates whether or not the classifier is
7       enabled to classify packets to a Service Flow.
8       If the referenced parameter is not present in the
9       classifier, the value of this object is reported
10      as active(1)."
```

```

11      ::= { wmanIf2CmnClassifierRuleEntry 24 }
12
13
14 wmanIf2CmnClassifierRulePkts OBJECT-TYPE
15     SYNTAX      Counter64
16     MAX-ACCESS  read-only
17     STATUS      current
18     DESCRIPTION
19         "This object counts the number of packets that have
20         been classified using this entry."
21     ::= { wmanIf2CmnClassifierRuleEntry 25 }
22
23
24
25 wmanIf2CmnClassifierRuleIpv6FlowLabel OBJECT-TYPE
26     SYNTAX      WmanIf2Ipv6FlowLabel
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "The value of this field specifies the matching values for
31         the IPv6 Flow label field."
32     ::= { wmanIf2CmnClassifierRuleEntry 26 }
33
34
35
36 wmanIf2CmnClassifierRuleBitMap OBJECT-TYPE
37     SYNTAX      WmanIf2ClassifierBitMap
38     MAX-ACCESS  read-only
39     STATUS      current
40     DESCRIPTION
41         "This object indicates which parameter encodings were
42         actually present in the entry. A bit set to '1' indicates
43         the corresponding classifier encoding is present, and '0'
44         means otherwise"
45     ::= { wmanIf2CmnClassifierRuleEntry 27 }
46
47
48
49 wmanIf2CmnPhsRuleTable OBJECT-TYPE
50     SYNTAX      SEQUENCE OF WmanIf2CmnPhsRuleEntry
51     MAX-ACCESS  not-accessible
52     STATUS      current
53     DESCRIPTION
54         "This table contains PHS rule dictionary entries. Each
55         entry contains the data of the header to be suppressed
56         along with its identification - PHSI. The classifier
57         uniquely maps packets to its associated PHS Rule. The
58         receiving entity uses the CID and the PHSI to restore the
59         PHSF. Once a PHSF has been assigned to a PHSI, it shall
60         not be changed. To change the value of a PHSF on a
61         service flow, a new PHS rule shall be defined, the old
62         rule is removed from the service flow, and the new rule
63
64
65
```

```

1         is added. When a classifier is deleted, any associated
2         PHS rule shall also be deleted."
3
4     REFERENCE
5         "Subclause 5.2.3 in IEEE Std 802.16-2004"
6     ::= { wmanIf2CmnPacketCs 2 }
7
8
9     wmanIf2CmnPhsRuleEntry OBJECT-TYPE
10        SYNTAX      WmanIf2CmnPhsRuleEntry
11        MAX-ACCESS  not-accessible
12        STATUS      current
13        DESCRIPTION
14            "This table provides one row for each PHS rule created
15            dynamically by the BS and SS on a given service flow. The
16            PHS rule is defined by the pair (PHSS, PHSM) for each
17            distinct header data. It is indexed by IfIndex,
18            wmanIf2CmnCpsSfId, and wmanIf2CmnPhsIndex. The table is
19            read-only for NMS. "
20
21        INDEX      { ifIndex, wmanIf2CmnCpsSfId,
22                    wmanIf2CmnPhsRulePhsIndex }
23
24        ::= { wmanIf2CmnPhsRuleTable 1 }
25
26
27     WmanIf2CmnPhsRuleEntry ::= SEQUENCE {
28         wmanIf2CmnPhsRulePhsIndex      INTEGER,
29         wmanIf2CmnPhsRulePhsField      OCTET STRING,
30         wmanIf2CmnPhsRulePhsMask      OCTET STRING,
31         wmanIf2CmnPhsRulePhsSize      Integer32,
32         wmanIf2CmnPhsRulePhsVerify    WmanIf2PhsRuleVerify}
33
34
35     wmanIf2CmnPhsRulePhsIndex OBJECT-TYPE
36        SYNTAX      INTEGER (1..255)
37        MAX-ACCESS  not-accessible
38        STATUS      current
39        DESCRIPTION
40            "The PHSI (PHS Index) has a value between 1 and 255, which
41            uniquely references the suppressed byte string. The index
42            is unique per service flow. The uplink and downlink PHSI
43            values are independent of each other."
44
45        REFERENCE
46            "Subclause 11.13.19.3.7.1 in IEEE Std 802.16-2004"
47        ::= { wmanIf2CmnPhsRuleEntry 1 }
48
49
50     wmanIf2CmnPhsRulePhsField OBJECT-TYPE
51        SYNTAX      OCTET STRING (SIZE(0..65535))
52        MAX-ACCESS  read-only
53        STATUS      current
54        DESCRIPTION
55            "The PHSF (PHS Field) is a string of bytes containing the
56            header information to be suppressed by the sending CS and
57            reconstructed by the receiving CS. The most significant
58            byte of the string corresponds to the first byte of the
59            CS-SDU."
60
61        REFERENCE
62            "Subclause 11.13.19.3.7.2 in IEEE Std 802.16-2004"
63        ::= { wmanIf2CmnPhsRuleEntry 2 }
64
65

```



```

1
2
3 wmanIf2CmnPhsRulePhsMask OBJECT-TYPE
4     SYNTAX      OCTET STRING (SIZE(0..65535))
5     MAX-ACCESS  read-only
6     STATUS      current
7     DESCRIPTION
8         "The PHSM An 8-bit mask that indicates which bytes in the
9         PHS Field (PHSF) to suppress and which bytes to not
10        suppress. The PHSM allows fields, such as sequence numbers
11        or checksums (which vary in value), to be excluded from
12        suppression with the constant bytes around them suppressed.
13        It is encoded as follows:
14        bit 0:
15            0 = don't suppress the 1st byte of the suppression field
16            1 = suppress first byte of the suppression field
17        bit 1:
18            0 = don't suppress the 2nd byte of the suppression field
19            1 = suppress second byte of the suppression field
20        bit x:
21            0 = don't suppress the (x+1) byte of the suppression
22            field
23            1 = suppress (x+1) byte of the suppression field
24        where the length of the octet string is ceiling
25        (wmanIf2CmnPhsRulePhsSize/8)."
```

```

30     REFERENCE
31         "Subclause 11.13.19.3.7.3 in IEEE Std 802.16-2004"
32     ::= { wmanIf2CmnPhsRuleEntry 3 }
33
34 wmanIf2CmnPhsRulePhsSize OBJECT-TYPE
35     SYNTAX      Integer32 (0..255)
36     UNITS       "byte"
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40         "The value of this field - PHSS is the total number of bytes
41         in the header to be suppressed and then restored in a
42         service flow that uses PHS."
43     REFERENCE
44         "Subclause 11.13.19.3.7.4 in IEEE Std 802.16-2004"
45     DEFVAL      {0}
46     ::= { wmanIf2CmnPhsRuleEntry 4 }
47
48 wmanIf2CmnPhsRulePhsVerify OBJECT-TYPE
49     SYNTAX      WmanIf2PhsRuleVerify
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The value of this field indicates to the sending entity
54         whether or not the packet header contents are to be
55         verified prior to performing suppression."
56     DEFVAL      { phsVerifyEnable }
57     ::= { wmanIf2CmnPhsRuleEntry 5 }
58
59
60
61
62
63
64
65 --

```

```

1  -- wmanIf2CmnCps contain the Common Part Sublayer objects that are
2  -- common to both Base Station and Subscriber Station
3  --
4
5  wmanIf2CmnCps OBJECT IDENTIFIER ::= { wmanIf2CommonObjects 2 }
6
7  wmanIf2CmnCpsServiceFlowTable OBJECT-TYPE
8      SYNTAX      SEQUENCE OF WmanIf2CmnCpsServiceFlowEntry
9      MAX-ACCESS  not-accessible
10     STATUS      current
11     DESCRIPTION
12         "This table contains Service Flow managed objects that
13         are common in BS and SS."
14     ::= { wmanIf2CmnCps 1 }
15
16
17  wmanIf2CmnCpsServiceFlowEntry OBJECT-TYPE
18     SYNTAX      WmanIf2CmnCpsServiceFlowEntry
19     MAX-ACCESS  not-accessible
20     STATUS      current
21     DESCRIPTION
22         "This table provides one row for each created service
23         flow for a given MacAddress, and is indexed by ifIndex,
24         wmanIf2CmnCpsCpsSfMacAddress, and wmanIf2CmnCpsSfId.
25         IfIndex is associated with the BS sector."
26     INDEX      { ifIndex, wmanIf2CmnCpsSfMacAddress,
27                 wmanIf2CmnCpsSfId }
28     ::= { wmanIf2CmnCpsServiceFlowTable 1 }
29
30
31  WmanIf2CmnCpsServiceFlowEntry ::= SEQUENCE {
32
33     wmanIf2CmnCpsSfMacAddress      MacAddress,
34     wmanIf2CmnCpsSfId              Unsigned32,
35     wmanIf2CmnCpsSfCid             WmanIf2CidType,
36     wmanIf2CmnCpsSfDirection       INTEGER,
37     wmanIf2CmnCpsSfState           WmanIf2SfState,
38     wmanIf2CmnCpsTrafficPriority    INTEGER,
39     wmanIf2CmnCpsMaxSustainedRate  Unsigned32,
40     wmanIf2CmnCpsMaxTrafficBurst   Unsigned32,
41     wmanIf2CmnCpsMinReservedRate   Unsigned32,
42     wmanIf2CmnCpsToleratedJitter   Unsigned32,
43     wmanIf2CmnCpsMaxLatency        Unsigned32,
44     wmanIf2CmnCpsFixedVsVariableSduInd  INTEGER,
45     wmanIf2CmnCpsSduSize           Unsigned32,
46     wmanIf2CmnCpsSfSchedulingType  WmanIf2SfSchedulingType,
47     wmanIf2CmnCpsArqEnable         TruthValue,
48     wmanIf2CmnCpsArqWindowSize     INTEGER,
49     wmanIf2CmnCpsArqBlockLifetime  INTEGER,
50     wmanIf2CmnCpsArqSyncLossTimeout  INTEGER,
51     wmanIf2CmnCpsArqDeliverInOrder TruthValue,
52     wmanIf2CmnCpsArqRxPurgeTimeout  INTEGER,
53     wmanIf2CmnCpsArqBlockSize     INTEGER,
54     wmanIf2CmnCpsMinRsvdTolerableRate  Unsigned32,
55     wmanIf2CmnCpsReqTxPolicy       BITS,
56     wmanIf2CmnSfCsSpecification    WmanIf2CsSpecification,
57     wmanIf2CmnCpsTargetSaid        INTEGER}
58
59
60
61
62
63
64
65

```

```

1  wmanIf2CmnCpsSfMacAddress OBJECT-TYPE
2      SYNTAX      MacAddress
3      MAX-ACCESS  not-accessible
4      STATUS      current
5      DESCRIPTION
6          "When this table is implemented on the basestation, this
7          object contains the SS Mac address, the reported service
8          flow was created for. On the SS, the value returned is
9          the SS's own Mac address."
10         ::= { wmanIf2CmnCpsServiceFlowEntry 1 }
11
12
13
14  wmanIf2CmnCpsSfId OBJECT-TYPE
15      SYNTAX      Unsigned32 ( 1 .. 4294967295)
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19          "A 32 bit quantity that uniquely identifies a service flow
20          to both the subscriber station and base station (BS)."

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this parameter specifies the priority
4          assigned to a service flow. For uplink service flows,
5          the BS should use this parameter when determining
6          precedence in request service and grant generation,
7          and the SS shall preferentially select contention
8          Request opportunities for Priority Request CIDs
9          based on this priority"
10     REFERENCE
11         "Subclause 11.13.5 in IEEE Std 802.16-2004"
12     ::= { wmanIf2CmnCpsServiceFlowEntry 6 }
13
14 wmanIf2CmnCpsMaxSustainedRate OBJECT-TYPE
15     SYNTAX      Unsigned32
16     UNITS       "b/s"
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20         "This parameter defines the peak information rate
21         of the service. The rate is expressed in bits per
22         second and pertains to the SDUs at the input to
23         the system."
24     REFERENCE
25         "Subclause 11.13.6 in IEEE Std 802.16-2004"
26     ::= { wmanIf2CmnCpsServiceFlowEntry 7 }
27
28 wmanIf2CmnCpsMaxTrafficBurst OBJECT-TYPE
29     SYNTAX      Unsigned32
30     UNITS       "byte"
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "This parameter defines the maximum burst size that
35         must be accommodated for the service."
36     REFERENCE
37         "Subclause 11.13.7 in IEEE Std 802.16-2004"
38     ::= { wmanIf2CmnCpsServiceFlowEntry 8 }
39
40 wmanIf2CmnCpsMinReservedRate OBJECT-TYPE
41     SYNTAX      Unsigned32
42     UNITS       "byte"
43     MAX-ACCESS  read-only
44     STATUS      current
45     DESCRIPTION
46         "This parameter specifies the minimum rate reserved
47         for this service flow."
48     REFERENCE
49         "Subclause 11.13.8 in IEEE Std 802.16-2004"
50     ::= { wmanIf2CmnCpsServiceFlowEntry 9 }
51
52 wmanIf2CmnCpsToleratedJitter OBJECT-TYPE
53     SYNTAX      Unsigned32
54     UNITS       "millisecond"

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This parameter defines the Maximum delay
5          variation (jitter) for the connection."
6      REFERENCE
7          "Subclause 11.13.13 in IEEE Std 802.16-2004"
8      ::= { wmanIf2CmnCpsServiceFlowEntry 10 }
9
10
11
12  wmanIf2CmnCpsMaxLatency OBJECT-TYPE
13      SYNTAX      Unsigned32
14      UNITS       "millisecond"
15      MAX-ACCESS  read-only
16      STATUS      current
17      DESCRIPTION
18          "The value of this parameter specifies the maximum
19          latency between the reception of a packet by the BS
20          or SS on its network interface and the forwarding
21          of the packet to its RF Interface."
22      REFERENCE
23          "Subclause 11.13.14 in IEEE Std 802.16-2004"
24      ::= { wmanIf2CmnCpsServiceFlowEntry 11 }
25
26
27
28
29  wmanIf2CmnCpsFixedVsVariableSduInd OBJECT-TYPE
30      SYNTAX      INTEGER {variableLength(0),
31                  fixedLength(1)}
32      MAX-ACCESS  read-only
33      STATUS      current
34      DESCRIPTION
35          "The value of this parameter specifies whether the SDUs
36          on the service flow are variable-length (0) or
37          fixed-length (1). The parameter is used only if
38          packing is on for the service flow. The default value
39          is 0, i.e., variable-length SDUs."
40      REFERENCE
41          "Subclause 11.13.15 in IEEE Std 802.16-2004"
42      DEFVAL      { variableLength }
43      ::= { wmanIf2CmnCpsServiceFlowEntry 12 }
44
45
46
47
48  wmanIf2CmnCpsSduSize OBJECT-TYPE
49      SYNTAX      Unsigned32
50      UNITS       "byte"
51      MAX-ACCESS  read-only
52      STATUS      current
53      DESCRIPTION
54          "The value of this parameter specifies the length of the
55          SDU for a fixed-length SDU service flow. This parameter
56          is used only if packing is on and the service flow is
57          indicated as carrying fixed-length SDUs. The default
58          value is 49 bytes, i.e., VC-switched ATM cells with PHS.
59          The parameter is relevant for both ATM and Packet
60          Convergence Sublayers."
61      REFERENCE
62          "Subclause 11.13.16 in IEEE Std 802.16-2004"
63
64
65

```

```

1      DEFVAL      { 49 }
2      ::= { wmanIf2CmnCpsServiceFlowEntry 13 }
3
4
5      wmanIf2CmnCpsSfsSchedulingType OBJECT-TYPE
6      SYNTAX      WmanIf2SfsSchedulingType
7      MAX-ACCESS  read-only
8      STATUS      current
9      DESCRIPTION
10     "Specifies the upstream scheduling service used for
11     upstream service flow. If the referenced parameter
12     is not present in the corresponding 802.16 QOS
13     Parameter Set of an upstream service flow, the
14     default value of this object is bestEffort(2)."
```

```

15     REFERENCE
16     "Subclause 11.13.11 in IEEE Std 802.16-2004"
17     DEFVAL      { bestEffort }
18     ::= { wmanIf2CmnCpsServiceFlowEntry 14 }
19
20
21
22
23     wmanIf2CmnCpsArqEnable OBJECT-TYPE
24     SYNTAX      TruthValue
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28     "True(1) ARQ enabling is requested for the connection."
29     ::= { wmanIf2CmnCpsServiceFlowEntry 15 }
30
31
32
33     wmanIf2CmnCpsArqWindowSize OBJECT-TYPE
34     SYNTAX      INTEGER (1..1024)
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38     "Indicates the maximum number of unacknowledged
39     fragments at any time."
40     ::= { wmanIf2CmnCpsServiceFlowEntry 16 }
41
42
43
44     wmanIf2CmnCpsArqBlockLifetime OBJECT-TYPE
45     SYNTAX      INTEGER (0 .. 65535)
46     UNITS       "10 us"
47     MAX-ACCESS  read-only
48     STATUS      current
49     DESCRIPTION
50     "The maximum time interval an ARQ fragment will be
51     managed by the transmitter ARQ machine, once
52     initial transmission of the fragment has occurred.
53     If transmission or retransmission of the fragment
54     is not acknowledged by the receiver before the
55     time limit is reached, the fragment is discarded.
56     A value of 0 means Infinite."
57     ::= { wmanIf2CmnCpsServiceFlowEntry 17 }
58
59
60
61
62     wmanIf2CmnCpsArqSyncLossTimeout OBJECT-TYPE
63     SYNTAX      INTEGER (0 .. 65535 )
64     UNITS       "10 us"
65     MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "The maximum interval before declaring a loss
4          of synchronization of the sender and receiver
5          state machines. A value of 0 means Infinite."
6      ::= { wmanIf2CmnCpsServiceFlowEntry 18 }
7
8
9
10     wmanIf2CmnCpsArqDeliverInOrder  OBJECT-TYPE
11         SYNTAX      TruthValue
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "Indicates whether or not data is to be delivered
16             by the receiving MAC to its client application
17             in the order in which data was handed off to the
18             originating MAC."
19         ::= { wmanIf2CmnCpsServiceFlowEntry 19 }
20
21
22
23     wmanIf2CmnCpsArqRxPurgeTimeout  OBJECT-TYPE
24         SYNTAX      INTEGER (0 .. 65535)
25         UNITS       "10 us"
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "Indicates the time interval the ARQ window is advanced
30             after a fragment is received. A value of 0 means
31             Infinite."
32         ::= { wmanIf2CmnCpsServiceFlowEntry 20 }
33
34
35
36     wmanIf2CmnCpsArqBlockSize  OBJECT-TYPE
37         SYNTAX      INTEGER (1..2040)
38         UNITS       "byte"
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "This value of this parameter specifies the size of an
43             ARQ block. This parameter shall be established by
44             negotiation during the connection creation dialog."
45         REFERENCE
46             "Subclause 11.13.18.8 in IEEE Std 802.16-2004"
47         ::= { wmanIf2CmnCpsServiceFlowEntry 21 }
48
49
50
51     wmanIf2CmnCpsMinRsvdTolerableRate  OBJECT-TYPE
52         SYNTAX      Unsigned32
53         UNITS       "b/s"
54         MAX-ACCESS  read-only
55         STATUS      current
56         DESCRIPTION
57             "Minimum Tolerable Traffic Rate = R (bits/sec) with
58             time base T(sec) means the following. Let S denote
59             additional demand accumulated at the MAC SAP of the
60             transmitter during an arbitrary time interval of the
61             length T. Then the amount of data forwarded at the
62             receiver to CS (in bits) during this interval should
63
64
65

```

```

1         be not less than min {S, R * T}."
2     REFERENCE
3         "Subclause 11.13.9 in IEEE Std 802.16-2004"
4     ::= { wmanIf2CmnCpsServiceFlowEntry 22 }
5
6
7     wmanIf2CmnCpsReqTxPolicy OBJECT-TYPE
8         SYNTAX      BITS {noBroadcastBwReq(0),
9                     reserved1(1),
10                    noPiggybackReq(2),
11                    noFragmentData(3),
12                    noPHS(4),
13                    noSduPacking(5),
14                    noCrc(6),
15                    reserved2(7)}
16
17     MAX-ACCESS    read-only
18     STATUS        current
19     DESCRIPTION
20         "The value of this parameter provides the capability to
21         specify certain attributes for the associated service
22         flow. An attribute is enabled by setting the
23         corresponding bit position to 1."
24     REFERENCE
25         "Subclause 11.13.12 in IEEE Std 802.16-2004"
26     ::= { wmanIf2CmnCpsServiceFlowEntry 23 }
27
28
29     wmanIf2CmnSfCsSpecification OBJECT-TYPE
30         SYNTAX      WmanIf2CsSpecification
31         MAX-ACCESS  read-only
32         STATUS      current
33         DESCRIPTION
34             "This parameter specifies the convergence sublayer
35             encapsulation mode."
36         REFERENCE
37             "Subclause 11.13.19.1 in IEEE Std 802.16-2004"
38         ::= { wmanIf2CmnCpsServiceFlowEntry 24 }
39
40
41     wmanIf2CmnCpsTargetSaid OBJECT-TYPE
42         SYNTAX      INTEGER (0 .. 65535)
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "The target SAID parameter indicates the SAID onto
47             which the service flow being set up shall be mapped."
48         REFERENCE
49             "Subclause 11.13.17 in IEEE Std 802.16-2004"
50         ::= { wmanIf2CmnCpsServiceFlowEntry 25 }
51
52
53     --
54     -- wmanIf2CmnBsSsConfigurationTable contains global parameters
55     -- common in BS and SS
56     --
57     wmanIf2CmnBsSsConfigurationTable OBJECT-TYPE
58         SYNTAX      SEQUENCE OF WmanIf2CmnBsSsConfigurationEntry
59         MAX-ACCESS  not-accessible
60
61
62
63
64
65

```



```

1      STATUS      current
2      DESCRIPTION
3          "This table provides one row for each BS sector that
4          contains the system parameters common in both SS and
5          BS. All SSs shall have the same parameters as the BS
6          to which the SSs are associated."
7
8      REFERENCE
9          "Subclause 10.1 in IEEE Std 802.16-2004"
10     ::= { wmanIf2CmnCps 2 }
11
12
13     wmanIf2CmnBsSsConfigurationEntry OBJECT-TYPE
14         SYNTAX      WmanIf2CmnBsSsConfigurationEntry
15         MAX-ACCESS  not-accessible
16         STATUS      current
17         DESCRIPTION
18             "This table is indexed by ifIndex, indicating BS
19             sector."
20         INDEX       { ifIndex }
21         ::= { wmanIf2CmnBsSsConfigurationTable 1 }
22
23
24
25     WmanIf2CmnBsSsConfigurationEntry ::= SEQUENCE {
26         wmanIf2CmnInvitedRangRetries      INTEGER,
27         wmanIf2CmnDSxReqRetries           Unsigned32,
28         wmanIf2CmnDSxRespRetries         Unsigned32,
29         wmanIf2CmnT7Timeout               INTEGER,
30         wmanIf2CmnT8Timeout               INTEGER,
31         wmanIf2CmnT10Timeout              INTEGER,
32         wmanIf2CmnT22Timeout              INTEGER}
33
34
35
36     wmanIf2CmnInvitedRangRetries OBJECT-TYPE
37         SYNTAX      INTEGER (16..65535)
38         MAX-ACCESS  read-write
39         STATUS      current
40         DESCRIPTION
41             "Number of retries on inviting Ranging Requests."
42         ::= { wmanIf2CmnBsSsConfigurationEntry 1 }
43
44
45
46     wmanIf2CmnDSxReqRetries OBJECT-TYPE
47         SYNTAX      Unsigned32
48         MAX-ACCESS  read-write
49         STATUS      current
50         DESCRIPTION
51             "Number of Timeout Retries on DSA/DSC/DSD Requests."
52         DEFVAL     { 3 }
53         ::= { wmanIf2CmnBsSsConfigurationEntry 2 }
54
55
56
57     wmanIf2CmnDSxRespRetries OBJECT-TYPE
58         SYNTAX      Unsigned32
59         MAX-ACCESS  read-write
60         STATUS      current
61         DESCRIPTION
62             "Number of Timeout Retries on DSA/DSC/DSD Responses."
63         DEFVAL     { 3 }
64         ::= { wmanIf2CmnBsSsConfigurationEntry 3 }
65

```

```

1
2
3 wmanIf2CmnT7Timeout OBJECT-TYPE
4     SYNTAX      INTEGER (0 .. 1000)
5     UNITS       "milliseconds"
6     MAX-ACCESS  read-write
7     STATUS      current
8     DESCRIPTION
9         "Wait for DSA/DSC/DSD Response Timeout in ms."
10    ::= { wmanIf2CmnBsSsConfigurationEntry 4 }
11
12
13 wmanIf2CmnT8Timeout OBJECT-TYPE
14     SYNTAX      INTEGER (0 .. 300)
15     UNITS       "milliseconds"
16     MAX-ACCESS  read-write
17     STATUS      current
18     DESCRIPTION
19         "Wait for DSA/DSC/DSD Acknowledge Timeout in ms."
20    ::= { wmanIf2CmnBsSsConfigurationEntry 5 }
21
22
23
24 wmanIf2CmnT10Timeout OBJECT-TYPE
25     SYNTAX      INTEGER (0 .. 3000)
26     UNITS       "milliseconds"
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "Wait for Transaction End timeout in ms."
31    ::= { wmanIf2CmnBsSsConfigurationEntry 6 }
32
33
34
35 wmanIf2CmnT22Timeout OBJECT-TYPE
36     SYNTAX      INTEGER (0 .. 500)
37     UNITS       "milliseconds"
38     MAX-ACCESS  read-write
39     STATUS      current
40     DESCRIPTION
41         "Wait for ARQ Reset in ms."
42    ::= { wmanIf2CmnBsSsConfigurationEntry 7 }
43
44
45 -- Common PKM group
46 -- wmanIf2CmnPkmObjects contain the Privacy Sublayer objects that are
47 -- common to both Base Station and Subscriber Station
48 --
49 --
50 wmanIf2CmnPkmObjects OBJECT IDENTIFIER ::= { wmanIf2CommonObjects 3 }
51
52 --
53 -- Table wmanIf2CmnCryptoSuiteTable
54 --
55 --
56 wmanIf2CmnCryptoSuiteTable OBJECT-TYPE
57     SYNTAX      SEQUENCE OF WmanIf2CmnCryptoSuiteEntry
58     MAX-ACCESS  not-accessible
59     STATUS      current
60     DESCRIPTION
61         "This table describes the PKM cryptographic suite
62         capabilities for each SS or BS wireless interface."
63    ::= { wmanIf2CmnPkmObjects 1 }
64
65

```

```

1
2
3 wmanIf2CmnCryptoSuiteEntry OBJECT-TYPE
4     SYNTAX      WmanIf2CmnCryptoSuiteEntry
5     MAX-ACCESS  not-accessible
6     STATUS      current
7     DESCRIPTION
8         "Each entry contains the cryptographic suite pair that SS
9         or BS supports."
10
11     INDEX       { ifIndex, wmanIf2CmnCryptoSuiteIndex }
12     ::= { wmanIf2CmnCryptoSuiteTable 1 }
13
14 WmanIf2CmnCryptoSuiteEntry ::= SEQUENCE {
15     wmanIf2CmnCryptoSuiteIndex      Integer32,
16     wmanIf2CmnCryptoSuiteDataEncryptAlg  WmanIf2DataEncryptAlgId,
17     wmanIf2CmnCryptoSuiteDataAuthAlg    WmanIf2DataAuthAlgId,
18     wmanIf2CmnCryptoSuiteTekEncryptAlg  WmanIf2TekEncryptAlgId}
19
20
21 wmanIf2CmnCryptoSuiteIndex OBJECT-TYPE
22     SYNTAX      Integer32 (1 .. 1000)
23     MAX-ACCESS  not-accessible
24     STATUS      current
25     DESCRIPTION
26         "The index for a cryptographic suite row."
27     ::= { wmanIf2CmnCryptoSuiteEntry 1 }
28
29
30
31 wmanIf2CmnCryptoSuiteDataEncryptAlg OBJECT-TYPE
32     SYNTAX      WmanIf2DataEncryptAlgId
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36         "The value of this object is the data encryption algorithm
37         for this cryptographic suite capability."
38     REFERENCE
39         "Table 375, IEEE Std 802.16-2004"
40     ::= { wmanIf2CmnCryptoSuiteEntry 2 }
41
42
43
44 wmanIf2CmnCryptoSuiteDataAuthAlg OBJECT-TYPE
45     SYNTAX      WmanIf2DataAuthAlgId
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "The value of this object is the data authentication
50         algorithm for this cryptographic suite capability."
51     REFERENCE
52         "Table 376, IEEE Std 802.16-2004"
53     ::= { wmanIf2CmnCryptoSuiteEntry 3 }
54
55
56
57 wmanIf2CmnCryptoSuiteTekEncryptAlg OBJECT-TYPE
58     SYNTAX      WmanIf2TekEncryptAlgId
59     MAX-ACCESS  read-only
60     STATUS      current
61     DESCRIPTION
62         "The value of this object is the TEK key encryption
63         algorithm for this cryptographic suite capability."
64
65

```

```

1      REFERENCE
2          "Table 377, IEEE Std 802.16-2004"
3      ::= { wmanIf2CmnCryptoSuiteEntry 4 }
4
5
6      --
7      -- Conformance Information
8      --
9      wmanIf2MibConformance OBJECT IDENTIFIER ::= {wmanIf2Mib 2}
10     wmanIf2MibGroups      OBJECT IDENTIFIER ::= {wmanIf2MibConformance 1}
11     wmanIf2MibCompliances OBJECT IDENTIFIER ::= {wmanIf2MibConformance 2}
12
13
14     -- compliance statements
15     wmanIf2MibCompliance MODULE-COMPLIANCE
16         STATUS          current
17         DESCRIPTION
18             "The compliance statement for devices that implement
19             Wireless MAN interfaces as defined in IEEE Std 802.16-2004."
20
21
22     MODULE -- wmanIf2Mib
23
24
25     MANDATORY-GROUPS          -- unconditionally mandatory groups
26         { wmanIf2MibCommonGroup }
27
28
29     GROUP wmanIf2MibQoSGroup  -- unconditionally mandatory group
30     DESCRIPTION
31         "This group is mandatory for Base Station and subscriber
32         station."
33
34
35     GROUP wmanIf2MibBsGroup   -- conditionally mandatory group
36     DESCRIPTION
37         "This group is mandatory for Base Station."
38
39
40     GROUP wmanIf2MibBsAasGroup -- optional group
41     DESCRIPTION
42         "This group is mandatory for Base Station."
43
44
45     GROUP wmanIf2MibSsGroup   -- conditionally mandatory group
46     DESCRIPTION
47         "This group is mandatory for Subscriber Station."
48
49
50     GROUP wmanIf2MibBsOfdmGroup -- conditionally mandatory group
51     DESCRIPTION
52         "This group is mandatory for Base Station
53         implementaing the OFDM PHY."
54
55
56     GROUP wmanIf2MibSsOfdmGroup -- conditionally mandatory group
57     DESCRIPTION
58         "This group is mandatory for Subscriber Station
59         implementing the OFDM PHY."
60
61
62     GROUP wmanIf2MibBsOfdmaGroup -- conditionally mandatory group
63     DESCRIPTION
64         "This group is mandatory for Base Station
65         implementaing the OFDMA PHY."

```

```

1
2     GROUP wmanIf2MibSsOfdmaGroup -- conditionally mandatory group
3     DESCRIPTION
4         "This group is mandatory for Subscriber Station
5         implementing the OFDMA PHY."
6
7
8     GROUP wmanIf2MibBsNotificationGroup -- unconditionally
9         -- mandatory groups
10
11     DESCRIPTION
12         "This group is mandatory for Base Station."
13
14     GROUP wmanIf2MibSsNotificationGroup -- optional group
15     DESCRIPTION
16         "This group is optional for Subscriber Station."
17
18
19     GROUP wmanIf2MibCmnPhsGroup -- optional group
20     DESCRIPTION
21         "This group is optional for Base Station and
22         Subscriber Station."
23
24
25     GROUP wmanIf2MibBsPhsGroup -- optional group
26     DESCRIPTION
27         "This group is optional for Base Station."
28     ::= { wmanIf2MibCompliances 1 }
29
30
31 wmanIf2MibCommonGroup      OBJECT-GROUP
32     OBJECTS {-- Classification
33         wmanIf2CmnClassifierRulePriority,
34         wmanIf2CmnClassifierRuleIpTosLow,
35         wmanIf2CmnClassifierRuleIpTosHigh,
36         wmanIf2CmnClassifierRuleIpTosMask,
37         wmanIf2CmnClassifierRuleIpProtocol,
38         wmanIf2CmnClassifierRuleIpSourceAddr,
39         wmanIf2CmnClassifierRuleIpSourceMask,
40         wmanIf2CmnClassifierRuleIpDestAddr,
41         wmanIf2CmnClassifierRuleIpDestMask,
42         wmanIf2CmnClassifierRuleSourcePortStart,
43         wmanIf2CmnClassifierRuleSourcePortEnd,
44         wmanIf2CmnClassifierRuleDestPortStart,
45         wmanIf2CmnClassifierRuleDestPortEnd,
46         wmanIf2CmnClassifierRuleDestMacAddr,
47         wmanIf2CmnClassifierRuleDestMacMask,
48         wmanIf2CmnClassifierRuleSourceMacAddr,
49         wmanIf2CmnClassifierRuleSourceMacMask,
50         wmanIf2CmnClassifierRuleEnetProtocolType,
51         wmanIf2CmnClassifierRuleEnetProtocol,
52         wmanIf2CmnClassifierRuleUserPriLow,
53         wmanIf2CmnClassifierRuleUserPriHigh,
54         wmanIf2CmnClassifierRuleVlanId,
55         wmanIf2CmnClassifierRuleState,
56         wmanIf2CmnClassifierRulePkts,
57         wmanIf2CmnClassifierRuleIpv6FlowLabel,
58         wmanIf2CmnClassifierRuleBitMap,
59
60
61
62
63
64
65

```

```

1      -- Configuration parameters
2      wmanIf2CmnCpsTargetSaid,
3      wmanIf2CmnInvitedRangRetries,
4      wmanIf2CmnDSxReqRetries,
5      wmanIf2CmnDSxRespRetries,
6      wmanIf2CmnT7Timeout,
7      wmanIf2CmnT8Timeout,
8      wmanIf2CmnT10Timeout,
9      wmanIf2CmnT22Timeout,
10     wmanIf2CmnCryptoSuiteDataEncryptAlg,
11     wmanIf2CmnCryptoSuiteDataAuthentAlg,
12     wmanIf2CmnCryptoSuiteTekEncryptAlg}
13
14     STATUS          current
15
16     DESCRIPTION
17         "This group contains objects for both BS and SS,
18         and are independent of PHY."
19     ::= { wmanIf2MibGroups 1 }
20
21
22
23 wmanIf2MibQoSGroup      OBJECT-GROUP
24     OBJECTS {wmanIf2CmnCpsSfId,
25             wmanIf2CmnCpsSfCid,
26             wmanIf2CmnCpsSfDirection,
27             wmanIf2CmnCpsSfState,
28             wmanIf2CmnCpsTrafficPriority,
29             wmanIf2CmnCpsMaxSustainedRate,
30             wmanIf2CmnCpsMaxTrafficBurst,
31             wmanIf2CmnCpsMinReservedRate,
32             wmanIf2CmnCpsToleratedJitter,
33             wmanIf2CmnCpsMaxLatency,
34             wmanIf2CmnCpsFixedVsVariableSduInd,
35             wmanIf2CmnCpsSduSize,
36             wmanIf2CmnCpsSfSchedulingType,
37             wmanIf2CmnCpsArqEnable,
38             wmanIf2CmnCpsArqWindowSize,
39             wmanIf2CmnCpsArqBlockLifetime,
40             wmanIf2CmnCpsArqSyncLossTimeout,
41             wmanIf2CmnCpsArqDeliverInOrder,
42             wmanIf2CmnCpsArqRxPurgeTimeout,
43             wmanIf2CmnCpsArqBlockSize,
44             wmanIf2CmnCpsMinRsvdTolerableRate,
45             wmanIf2CmnCpsReqTxPolicy,
46             wmanIf2CmnSfCsSpecification}
47
48     STATUS          current
49
50     DESCRIPTION
51         "This group contains QoS objects for both BS and SS."
52     ::= { wmanIf2MibGroups 2 }
53
54
55
56
57 wmanIf2MibBsGroup      OBJECT-GROUP
58     OBJECTS {-- Service classes
59             wmanIf2BsSfDirection,
60             wmanIf2BsServiceClassIndex,
61             wmanIf2BsSfState,
62             wmanIf2BsSfProvisionedTime,
63             wmanIf2BsProvisionedSfRowStatus,
64
65

```

```
1      wmanIf2BsSsProvisionedForSfRowStatus,
2      wmanIf2BsSfCsSpecification,
3      wmanIf2BsQoSServiceClassName,
4      wmanIf2BsQoSSTrafficPriority,
5      wmanIf2BsQoSMaxSustainedRate,
6      wmanIf2BsQoSMaxTrafficBurst,
7      wmanIf2BsQoSMinReservedRate,
8      wmanIf2BsQOSToleratedJitter,
9      wmanIf2BsQoSMaxLatency,
10     wmanIf2BsQoSFixedVsVariableSduInd,
11     wmanIf2BsQoSsduSize,
12     wmanIf2BsQoSschedulingType,
13     wmanIf2BsQoSsArqEnable,
14     wmanIf2BsQoSsArqWindowSize,
15     wmanIf2BsQoSsArqBlockLifetime,
16     wmanIf2BsQoSsArqSyncLossTimeout,
17     wmanIf2BsQoSsArqDeliverInOrder,
18     wmanIf2BsQoSsArqRxPurgeTimeout,
19     wmanIf2BsQoSsArqBlockSize,
20     wmanIf2BsQoSsCMinRsvdTolerableRate,
21     wmanIf2BsQoSsReqTxPolicy,
22     wmanIf2BsQoSServiceClassRowStatus,
23
24     -- Classification
25     wmanIf2BsClassifierRulePriority,
26     wmanIf2BsClassifierRuleIpTosLow,
27     wmanIf2BsClassifierRuleIpTosHigh,
28     wmanIf2BsClassifierRuleIpTosMask,
29     wmanIf2BsClassifierRuleIpProtocol,
30     wmanIf2BsClassifierRuleIpSourceAddr,
31     wmanIf2BsClassifierRuleIpSourceMask,
32     wmanIf2BsClassifierRuleIpDestAddr,
33     wmanIf2BsClassifierRuleIpDestMask,
34     wmanIf2BsClassifierRuleSourcePortStart,
35     wmanIf2BsClassifierRuleSourcePortEnd,
36     wmanIf2BsClassifierRuleDestPortStart,
37     wmanIf2BsClassifierRuleDestPortEnd,
38     wmanIf2BsClassifierRuleDestMacAddr,
39     wmanIf2BsClassifierRuleDestMacMask,
40     wmanIf2BsClassifierRuleSourceMacAddr,
41     wmanIf2BsClassifierRuleSourceMacMask,
42     wmanIf2BsClassifierRuleEnetProtocolType,
43     wmanIf2BsClassifierRuleEnetProtocol,
44     wmanIf2BsClassifierRuleUserPriLow,
45     wmanIf2BsClassifierRuleUserPriHigh,
46     wmanIf2BsClassifierRuleVlanId,
47     wmanIf2BsClassifierRuleState,
48     wmanIf2BsClassifierRulePhsSize,
49     wmanIf2BsClassifierRulePhsMask,
50     wmanIf2BsClassifierRulePhsVerify,
51     wmanIf2BsClassifierRuleIpv6FlowLabel,
52     wmanIf2BsClassifierRuleBitMap,
53     wmanIf2BsClassifierRuleRowStatus,
```

```
1      -- Packet counters
2      wmanIf2BsSsMacSduCount,
3      wmanIf2BsSsOctetCount,
4      wmanIf2BsSsResetCounter,
5      wmanIf2BsSsResetCounterTime,
6
7
8      -- Capability negotiation
9      wmanIf2BsSsBasicCid,
10     wmanIf2BsSsPrimaryCid,
11     wmanIf2BsSsSecondaryCid,
12     wmanIf2BsSsManagementSupport,
13     wmanIf2BsSsIpManagementMode,
14     wmanIf2Bs2ndMgmtDlQoSProfileIndex,
15     wmanIf2Bs2ndMgmtUlQoSProfileIndex,
16     wmanIf2BsAutoSfidEnabled,
17     wmanIf2BsAutoSfidRangeMin,
18     wmanIf2BsAutoSfidRangeMax,
19     wmanIf2BsResetSector,
20     wmanIf2BsSs2ndMgmtArqEnable,
21     wmanIf2BsSs2ndMgmtArqWindowSize,
22     wmanIf2BsSs2ndMgmtArqDnLinkTxDelay,
23     wmanIf2BsSs2ndMgmtArqUpLinkTxDelay,
24     wmanIf2BsSs2ndMgmtArqDnLinkRxDelay,
25     wmanIf2BsSs2ndMgmtArqUpLinkRxDelay,
26     wmanIf2BsSs2ndMgmtArqBlockLifetime,
27     wmanIf2BsSs2ndMgmtArqSyncLossTimeout,
28     wmanIf2BsSs2ndMgmtArqDeliverInOrder,
29     wmanIf2BsSs2ndMgmtArqRxPurgeTimeout,
30     wmanIf2BsSs2ndMgmtArqBlockSize,
31     wmanIf2BsSsVendorIdEncoding,
32     wmanIf2BsSsAasBroadcastPermission,
33     wmanIf2BsSsMaxTxPowerBpsk,
34     wmanIf2BsSsMaxTxPowerQpsk,
35     wmanIf2BsSsMaxTxPower16Qam,
36     wmanIf2BsSsMaxTxPower64Qam,
37
38     -- Configuration parameters
39     wmanIf2BsSsMacVersion,
40     wmanIf2BsDcdInterval,
41     wmanIf2BsUcdInterval,
42     wmanIf2BsUcdTransition,
43     wmanIf2BsDcdTransition,
44     wmanIf2BsInitialRangingInterval,
45     wmanIf2BsSsULMapProcTime,
46     wmanIf2BsSsRangRespProcTime,
47     wmanIf2BsT5Timeout,
48     wmanIf2BsT9Timeout,
49     wmanIf2BsT13Timeout,
50     wmanIf2BsT15Timeout,
51     wmanIf2BsT17Timeout,
52     wmanIf2BsT27IdleTimer,
53     wmanIf2BsT27ActiveTimer,
54
55     -- Performance monitoring
```



```

1      wmanIf2BsHistogramIndex,
2      wmanIf2BsChannelNumber,
3      wmanIf2BsStartFrame,
4      wmanIf2BsDuration,
5      wmanIf2BsBasicReport,
6      wmanIf2BsMeanCinrReport,
7      wmanIf2BsMeanRssiReport,
8      wmanIf2BsStdDeviationCinrReport,
9      wmanIf2BsStdDeviationRssiReport,
10
11
12
13      -- Capability negotiation
14      wmanIf2BsSsReqCapUplinkCidSupport,
15      wmanIf2BsSsReqCapArqSupport,
16      wmanIf2BsSsReqCapDsxFlowControl,
17      wmanIf2BsSsReqCapMacCrcSupport,
18      wmanIf2BsSsReqCapMcaFlowControl,
19      wmanIf2BsSsReqCapMcpGroupCidSupport,
20      wmanIf2BsSsReqCapPkmFlowControl,
21      wmanIf2BsSsReqCapAuthPolicyControl,
22      wmanIf2BsSsReqCapMaxNumOfSupportedSA,
23      wmanIf2BsSsReqCapIpVersion,
24      wmanIf2BsSsReqCapMacCsSupportBitMap,
25      wmanIf2BsSsReqCapMaxNumOfClassifier,
26      wmanIf2BsSsReqCapPhsSupport,
27      wmanIf2BsSsReqCapBandwidthAllocSupport,
28      wmanIf2BsSsReqCapPduConstruction,
29      wmanIf2BsSsReqCapTtgTransitionGap,
30      wmanIf2BsSsReqCapRtgTransitionGap,
31      wmanIf2BsSsRspCapUplinkCidSupport,
32      wmanIf2BsSsRspCapArqSupport,
33      wmanIf2BsSsRspCapDsxFlowControl,
34      wmanIf2BsSsRspCapMacCrcSupport,
35      wmanIf2BsSsRspCapMcaFlowControl,
36      wmanIf2BsSsRspCapMcpGroupCidSupport,
37      wmanIf2BsSsRspCapPkmFlowControl,
38      wmanIf2BsSsRspCapAuthPolicyControl,
39      wmanIf2BsSsRspCapMaxNumOfSupportedSA,
40      wmanIf2BsSsRspCapIpVersion,
41      wmanIf2BsSsRspCapMacCsSupportBitMap,
42      wmanIf2BsSsRspCapMaxNumOfClassifier,
43      wmanIf2BsSsRspCapPhsSupport,
44      wmanIf2BsSsRspCapBandwidthAllocSupport,
45      wmanIf2BsSsRspCapPduConstruction,
46      wmanIf2BsSsRspCapTtgTransitionGap,
47      wmanIf2BsSsRspCapRtgTransitionGap,
48      wmanIf2BsCapUplinkCidSupport,
49      wmanIf2BsCapArqSupport,
50      wmanIf2BsCapDsxFlowControl,
51      wmanIf2BsCapMacCrcSupport,
52      wmanIf2BsCapMcaFlowControl,
53      wmanIf2BsCapMcpGroupCidSupport,
54      wmanIf2BsCapPkmFlowControl,
55      wmanIf2BsCapAuthPolicyControl,
56      wmanIf2BsCapMaxNumOfSupportedSA,
57
58
59
60
61
62
63
64
65

```

```
1 wmanIf2BsCapIpVersion,
2 wmanIf2BsCapMacCsSupportBitMap,
3 wmanIf2BsCapMaxNumOfClassifier,
4 wmanIf2BsCapPhsSupport,
5 wmanIf2BsCapBandwidthAllocSupport,
6 wmanIf2BsCapPduConstruction,
7 wmanIf2BsCapTtgTransitionGap,
8 wmanIf2BsCapRtgTransitionGap,
9 wmanIf2BsCapCfgUplinkCidSupport,
10 wmanIf2BsCapCfgArqSupport,
11 wmanIf2BsCapCfgDsxFowControl,
12 wmanIf2BsCapCfgMacCrcSupport,
13 wmanIf2BsCapCfgMcaFlowControl,
14 wmanIf2BsCapCfgMcpGroupCidSupport,
15 wmanIf2BsCapCfgPkmFlowControl,
16 wmanIf2BsCapCfgAuthPolicyControl,
17 wmanIf2BsCapCfgMaxNumOfSupportedSA,
18 wmanIf2BsCapCfgIpVersion,
19 wmanIf2BsCapCfgMacCsSupportBitMap,
20 wmanIf2BsCapCfgMaxNumOfClassifier,
21 wmanIf2BsCapCfgPhsSupport,
22 wmanIf2BsCapCfgBandwidthAllocSupport,
23 wmanIf2BsCapCfgPduConstruction,
24 wmanIf2BsCapCfgTtgTransitionGap,
25 wmanIf2BsCapCfgRtgTransitionGap,
26 wmanIf2BsSsActionsResetSs,
27 wmanIf2BsSsActionsAbortSs,
28 wmanIf2BsSsActionsOverrideDnFreq,
29 wmanIf2BsSsActionsOverrideChannelId,
30 wmanIf2BsSsActionsDeReRegSs,
31 wmanIf2BsSsActionsDeReRegSsCode,
32 wmanIf2BsSsActionsRowStatus,
33
34 -- Privacy sublayer
35 wmanIf2BsPkmDefaultAuthLifetime,
36 wmanIf2BsPkmDefaultTekLifetime,
37 wmanIf2BsPkmDefaultSelfSigManufCertTrust,
38 wmanIf2BsPkmCheckCertValidityPeriods,
39 wmanIf2BsPkmAuthentInfos,
40 wmanIf2BsPkmAuthRequests,
41 wmanIf2BsPkmAuthReplies,
42 wmanIf2BsPkmAuthRejects,
43 wmanIf2BsPkmAuthInvalids,
44 wmanIf2BsSsPkmAuthKeySequenceNumber,
45 wmanIf2BsSsPkmAuthExpiresOld,
46 wmanIf2BsSsPkmAuthExpiresNew,
47 wmanIf2BsSsPkmAuthLifetime,
48 wmanIf2BsSsPkmAuthReset,
49 wmanIf2BsSsPkmAuthInfos,
50 wmanIf2BsSsPkmAuthRequests,
51 wmanIf2BsSsPkmAuthReplies,
52 wmanIf2BsSsPkmAuthRejects,
53 wmanIf2BsSsPkmAuthInvalids,
54 wmanIf2BsSsPkmAuthRejectErrorCode,
```

```

1      wmanIf2BsSsPkmAuthRejectErrorString,
2      wmanIf2BsSsPkmAuthInvalidErrorCode,
3      wmanIf2BsSsPkmAuthInvalidErrorString,
4      wmanIf2BsSsPkmAuthPrimarySAId,
5      wmanIf2BsSsPkmAuthValidStatus,
6      wmanIf2BsPkmTekSAType,
7      wmanIf2BsPkmTekDataEncryptAlg,
8      wmanIf2BsPkmTekDataAuthentAlg,
9      wmanIf2BsPkmTekEncryptAlg,
10     wmanIf2BsPkmTekLifetime,
11     wmanIf2BsPkmTekKeySequenceNumber,
12     wmanIf2BsPkmTekExpiresOld,
13     wmanIf2BsPkmTekExpiresNew,
14     wmanIf2BsPkmTekReset,
15     wmanIf2BsPkmKeyRequests,
16     wmanIf2BsPkmKeyReplies,
17     wmanIf2BsPkmKeyRejects,
18     wmanIf2BsPkmTekInvalids,
19     wmanIf2BsPkmKeyRejectErrorCode,
20     wmanIf2BsPkmKeyRejectErrorString,
21     wmanIf2BsPkmTekInvalidErrorCode,
22     wmanIf2BsPkmTekInvalidErrorString,
23
24     -- Notification
25     wmanIf2BsTrapControlRegister,
26     wmanIf2BsStatusTrapControlRegister,
27     wmanIf2BsRssiLowThreshold,
28     wmanIf2BsRssiHighThreshold,
29     wmanIf2BsSsNotificationMacAddr,
30     wmanIf2BsSsStatusValue,
31     wmanIf2BsSsStatusInfo,
32     wmanIf2BsDynamicServiceType,
33     wmanIf2BsDynamicServiceFailReason,
34     wmanIf2BsSsRssiStatus,
35     wmanIf2BsSsRssiStatusInfo,
36     wmanIf2BsSsRegisterStatus}
37
38 STATUS          current
39 DESCRIPTION
40     "This group contains objects for BS, and are
41     independent of PHY."
42 ::= { wmanIf2MibGroups 3 }
43
44 wmanIf2MibBsAasGroup    OBJECT-GROUP
45 OBJECTS {-- AAS Configuration parameters
46     wmanIf2BsAasChanFbckReqFreq,
47     wmanIf2BsAasBeamSelectFreq,
48     wmanIf2BsAasChanFbckReqResolution,
49     wmanIf2BsAasBeamReqResolution,
50     wmanIf2BsAasNumOptDiversityZones}
51
52 STATUS          current
53 DESCRIPTION
54     "This group contains objects for AAS in BS."
55 ::= { wmanIf2MibGroups 4 }
56
57
58
59
60
61
62
63
64
65

```

```

1  wmanIf2MibSsGroup      OBJECT-GROUP
2      OBJECTS {-- Configuration parameters
3          wmanIf2SsLostDLMapInterval,
4          wmanIf2SsLostULMapInterval,
5          wmanIf2SsContentionRangRetries,
6          wmanIf2SsRequestRetries,
7          wmanIf2SsRegRequestRetries,
8          wmanIf2SsTftpBackoffStart,
9          wmanIf2SsTftpBackoffEnd,
10         wmanIf2SsTftpRequestRetries,
11         wmanIf2SsTftpDownloadRetries,
12         wmanIf2SsTftpWait,
13         wmanIf2SsToDRetries,
14         wmanIf2SsToDRetryPeriod,
15         wmanIf2SsT1Timeout,
16         wmanIf2SsT2Timeout,
17         wmanIf2SsT3Timeout,
18         wmanIf2SsT4Timeout,
19         wmanIf2SsT6Timeout,
20         wmanIf2SsT12Timeout,
21         wmanIf2SsT14Timeout,
22         wmanIf2SsT16Timeout,
23         wmanIf2SsT18Timeout,
24         wmanIf2SsT19Timeout,
25         wmanIf2SsT20Timeout,
26         wmanIf2SsT21Timeout,
27         wmanIf2SsSBCRequestRetries,
28         wmanIf2SsTftpCpltRetries,
29         wmanIf2SsT26Timeout,
30         wmanIf2SsDLManagProcTime,
31
32         -- Performance monitoring
33         wmanIf2SsChannelNumber,
34         wmanIf2SsStartFrame ,
35         wmanIf2SsDuration,
36         wmanIf2SsBasicReport,
37         wmanIf2SsMeanCinrReport,
38         wmanIf2SsStdDeviationCinrReport,
39         wmanIf2SsMeanRssiReport,
40         wmanIf2SsStdDeviationRssiReport,
41
42         -- Privacy sublayer
43         wmanIf2SsPkmAuthState,
44         wmanIf2SsPkmAuthKeySequenceNumber,
45         wmanIf2SsPkmAuthExpiresOld,
46         wmanIf2SsPkmAuthExpiresNew ,
47         wmanIf2SsPkmAuthReset,
48         wmanIf2SsPkmAuthentInfos,
49         wmanIf2SsPkmAuthRequests,
50         wmanIf2SsPkmAuthReplies,
51         wmanIf2SsPkmAuthRejects,
52         wmanIf2SsPkmAuthInvalids,
53         wmanIf2SsPkmAuthRejectErrorCode,
54         wmanIf2SsPkmAuthRejectErrorString,
55
56
57
58
59
60
61
62
63
64
65

```

```

1      wmanIf2SsPkmAuthInvalidErrorCode,
2      wmanIf2SsPkmAuthInvalidErrorString ,
3      wmanIf2SsPkmAuthGraceTime,
4      wmanIf2SsPkmTekGraceTime,
5      wmanIf2SsPkmAuthWaitTimeout,
6      wmanIf2SsPkmReauthWaitTimeout,
7      wmanIf2SsPkmOpWaitTimeout,
8      wmanIf2SsPkmRekeyWaitTimeout,
9      wmanIf2SsPkmAuthRejectWaitTimeout,
10     wmanIf2SsPkmTekSAType,
11     wmanIf2SsPkmTekDataEncryptAlg,
12     wmanIf2SsPkmTekDataAuthentAlg,
13     wmanIf2SsPkmTekEncryptAlg,
14     wmanIf2SsPkmTekState,
15     wmanIf2SsPkmTekKeySequenceNumber,
16     wmanIf2SsPkmTekExpiresOld,
17     wmanIf2SsPkmTekExpiresNew,
18     wmanIf2SsPkmTekKeyRequests,
19     wmanIf2SsPkmTekKeyReplies,
20     wmanIf2SsPkmTekKeyRejects,
21     wmanIf2SsPkmTekInvalids,
22     wmanIf2SsPkmTekAuthPends,
23     wmanIf2SsPkmTekKeyRejectErrorCode,
24     wmanIf2SsPkmTekKeyRejectErrorString,
25     wmanIf2SsPkmTekInvalidErrorCode,
26     wmanIf2SsPkmTekInvalidErrorString,
27     wmanIf2SsDeviceCert,
28     wmanIf2SsDeviceManufCert,
29
30     -- Notofocation
31     wmanIf2SsTrapControlRegister,
32     wmanIf2SsRssiLowThreshold,
33     wmanIf2SsRssiHighThreshold,
34     wmanIf2SsMacAddress,
35     wmanIf2SsUnknownTlv,
36     wmanIf2SsDynamicServiceType,
37     wmanIf2SsDynamicServiceFailReason,
38     wmanIf2SsRssiStatus,
39     wmanIf2SsRssiStatusInfo}
40
41     STATUS          current
42     DESCRIPTION
43     "This group contains objects for SS, and are
44     independent of PHY."
45     ::= { wmanIf2MibGroups 5 }
46
47 wmanIf2MibBsOfdmGroup      OBJECT-GROUP
48     OBJECTS {wmanIf2BsOfdmCtBasedResvTimeout,
49     wmanIf2BsOfdmBwReqOppSize,
50     wmanIf2BsOfdmRangReqOppSize,
51     wmanIf2BsOfdmUplinkCenterFreq,
52     wmanIf2BsOfdmNumSubChReqRegionFull,
53     wmanIf2BsOfdmNumSymbolsReqRegionFull,
54     wmanIf2BsOfdmSubChFocusCtCode,
55     wmanIf2BsOfdmUpLinkChannelId,

```

```

1      wmanIf2BsOfdmBsEIRP,
2      wmanIf2BsOfdmChannelNumber,
3      wmanIf2BsOfdmTTG,
4      wmanIf2BsOfdmRTG,
5      wmanIf2BsOfdmInitRngMaxRSS,
6      wmanIf2BsOfdmDownlinkCenterFreq,
7      wmanIf2BsOfdmBsId,
8      wmanIf2BsOfdmMacVersion,
9      wmanIf2BsOfdmFrameDurationCode,
10     wmanIf2BsOfdmDownLinkChannelId,
11     wmanIf2BsOfdmUcdFecCodeType,
12     wmanIf2BsOfdmFocusCtPowerBoost,
13     wmanIf2BsOfdmUcdTcsEnable,
14     wmanIf2BsOfdmUcdBurstProfileRowStatus,
15     wmanIf2BsOfdmDownlinkFrequency,
16     wmanIf2BsOfdmDcdFecCodeType,
17     wmanIf2BsOfdmDiucMandatoryExitThresh,
18     wmanIf2BsOfdmDiucMinEntryThresh,
19     wmanIf2BsOfdmTcsEnable,
20     wmanIf2BsOfdmDcdBurstProfileRowStatus,
21     wmanIf2BsOfdmMinReqRegionFullTxOpp,
22     wmanIf2BsOfdmMinFocusedCtTxOpp,
23     wmanIf2BsOfdmMaxRoundTripDelay,
24     wmanIf2BsOfdmRangeAbortTimingThold,
25     wmanIf2BsOfdmRangeAbortPowerThold ,
26     wmanIf2BsOfdmRangeAbortFreqThold,
27     wmanIf2BsOfdmDnlkRateId,
28     wmanIf2BsOfdmRatioG,
29     wmanIf2BsSsOfdmReqCapFftSizes,
30     wmanIf2BsSsOfdmReqCapSsDemodulator,
31     wmanIf2BsSsOfdmReqCapSsModulator,
32     wmanIf2BsSsOfdmReqCapFocusedCtSupport,
33     wmanIf2BsSsOfdmReqCapTcSublayerSupport,
34     wmanIf2BsSsOfdmRspCapFftSizes,
35     wmanIf2BsSsOfdmRspCapSsDemodulator,
36     wmanIf2BsSsOfdmRspCapSsModulator,
37     wmanIf2BsSsOfdmRspCapFocusedCtSupport,
38     wmanIf2BsSsOfdmRspCapTcSublayerSupport,
39     wmanIf2BsOfdmCapFftSizes,
40     wmanIf2BsOfdmCapSsDemodulator,
41     wmanIf2BsOfdmCapSsModulator,
42     wmanIf2BsOfdmCapFocusedCtSupport,
43     wmanIf2BsOfdmCapTcSublayerSupport,
44     wmanIf2BsOfdmCapCfgFftSizes,
45     wmanIf2BsOfdmCapCfgSsDemodulator,
46     wmanIf2BsOfdmCapCfgSsModulator,
47     wmanIf2BsOfdmCapCfgFocusedCtSupport,
48     wmanIf2BsOfdmCapCfgTcSublayerSupport }
49
50     STATUS          current
51     DESCRIPTION
52     "This group contains objects for BS and OFDM PHY."
53     ::= { wmanIf2MibGroups 6 }
54
55 wmanIf2MibSsOfdmGroup      OBJECT-GROUP
56
57
58
59
60
61
62
63
64
65

```

```

1      OBJECTS {wmanIf2SsOfdmCtBasedResvTimeout,
2              wmanIf2SsOfdmBwReqOppSize,
3              wmanIf2SsOfdmRangReqOppSize,
4              wmanIf2SsOfdmUplinkCenterFreq,
5              wmanIf2SsOfdmNumSubChReqRegionFull,
6              wmanIf2SsOfdmNumSymbolsReqRegionFull,
7              wmanIf2SsOfdmSubChFocusCtCode,
8              wmanIf2SsOfdmUpLinkChannelId,
9              wmanIf2SsOfdmBsEIRP,
10             wmanIf2SsOfdmChannelNumber,
11             wmanIf2SsOfdmTTG,
12             wmanIf2SsOfdmRTG,
13             wmanIf2SsOfdmInitRngMaxRSS,
14             wmanIf2SsOfdmDownlinkCenterFreq,
15             wmanIf2SsOfdmBsId,
16             wmanIf2SsOfdmMacVersion,
17             wmanIf2SsOfdmFrameDurationCode,
18             wmanIf2SsOfdmDownLinkChannelId,
19             wmanIf2SsOfdmUcdFecCodeType,
20             wmanIf2SsOfdmFocusCtPowerBoost,
21             wmanIf2SsOfdmUcdTcsEnable,
22             wmanIf2SsOfdmDownlinkFrequency,
23             wmanIf2SsOfdmDcdFecCodeType,
24             wmanIf2SsOfdmDiucMandatoryExitThresh,
25             wmanIf2SsOfdmDiucMinEntryThresh,
26             wmanIf2SsOfdmTcsEnable}
27
28     STATUS          current
29
30     DESCRIPTION
31         "This group contains objects for SS and OFDM PHY."
32     ::= { wmanIf2MibGroups 7 }
33
34
35
36
37
38 wmanIf2MibBsOfdmaGroup      OBJECT-GROUP
39     OBJECTS {wmanIf2BsOfdmaCtBasedResvTimeout,
40             wmanIf2BsOfdmaBwReqOppSize,
41             wmanIf2BsOfdmaRangReqOppSize,
42             wmanIf2BsOfdmaUplinkCenterFreq,
43             wmanIf2BsOfdmaInitRngCodes,
44             wmanIf2BsOfdmaPeriodicRngCodes,
45             wmanIf2BsOfdmaBWReqCodes,
46             wmanIf2BsOfdmaPerRngBackoffStart,
47             wmanIf2BsOfdmaPerRngBackoffEnd,
48             wmanIf2BsOfdmaStartOfRngCodes,
49             wmanIf2BsOfdmaPermutationBase,
50             wmanIf2BsOfdmaULAllocSubchBitmap,
51             wmanIf2BsOfdmaOptPermULAllocSubchBitmap,
52             wmanIf2BsOfdmaBandAMCAllocThreshold,
53             wmanIf2BsOfdmaBandAMCReleaseThreshold,
54             wmanIf2BsOfdmaBandAMCAllocTimer,
55             wmanIf2BsOfdmaBandAMCReleaseTimer,
56             wmanIf2BsOfdmaBandStatRepMAXPeriod,
57             wmanIf2BsOfdmaBandAMCRetryTimer,
58             wmanIf2BsOfdmaSafetyChAllocThreshold,
59             wmanIf2BsOfdmaSafetyChReleaseThreshold,
60             wmanIf2BsOfdmaSafetyChAllocTimer,
61
62
63
64
65

```

```

1      wmanIf2BsOfdmaSafetyChReleaseTimer,
2      wmanIf2BsOfdmaBinStatRepMAXPeriod,
3      wmanIf2BsOfdmaSafetyChARetryTimer,
4      wmanIf2BsOfdmaHARQAackDelayULBurst,
5      wmanIf2BsOfdmaCQICHBandAMCTranaDelay,
6      wmanIf2BsOfdmaBsEIRP,
7      wmanIf2BsOfdmaChannelNumber,
8      wmanIf2BsOfdmaTTG,
9      wmanIf2BsOfdmaRTG,
10     wmanIf2BsOfdmaInitRngMaxRSS,
11     wmanIf2BsOfdmaDownlinkCenterFreq,
12     wmanIf2BsOfdmaBsId,
13     wmanIf2BsOfdmaMacVersion,
14     wmanIf2BsOfdmaFrameDurationCode,
15     wmanIf2BsOfdmaSizeCqichIdField,
16     wmanIf2BsOfdmaHARQAackDelayBurst,
17     wmanIf2BsOfdmaUcdFecCodeType,
18     wmanIf2BsOfdmaRangingDataRatio,
19     wmanIf2BsOfdmaNorCOVerNOVERRIDE,
20     wmanIf2BsOfdmaUcdBurstProfileRowStatus,
21     wmanIf2BsOfdmaDownlinkFrequency,
22     wmanIf2BsOfdmaDcdFecCodeType,
23     wmanIf2BsOfdmaDiucMandatoryExitThresh,
24     wmanIf2BsOfdmaDiucMinEntryThresh,
25     wmanIf2BsOfdmaDcdBurstProfileRowStatus}
26
27 STATUS      current
28
29 DESCRIPTION
30     "This group contains objects for BS and OFDMA PHY."
31     ::= { wmanIf2MibGroups 8 }
32
33 wmanIf2MibSsOfdmaGroup      OBJECT-GROUP
34     OBJECTS {wmanIf2SsOfdmaCtBasedResvTimeout,
35             wmanIf2SsOfdmaBwReqOppSize,
36             wmanIf2SsOfdmaRangReqOppSize,
37             wmanIf2SsOfdmaUplinkCenterFreq,
38             wmanIf2SsOfdmaInitRngCodes,
39             wmanIf2SsOfdmaPeriodicRngCodes,
40             wmanIf2SsOfdmaBWReqCodes,
41             wmanIf2SsOfdmaPerRngBackoffStart,
42             wmanIf2SsOfdmaPerRngBackoffEnd,
43             wmanIf2SsOfdmaStartOfRngCodes,
44             wmanIf2SsOfdmaPermutationBase,
45             wmanIf2SsOfdmaULAllocSubchBitmap,
46             wmanIf2SsOfdmaOptPermULAllocSubchBitmap,
47             wmanIf2SsOfdmaBandAMCAAllocThreshold,
48             wmanIf2SsOfdmaBandAMCReleaseThreshold,
49             wmanIf2SsOfdmaBandAMCAAllocTimer,
50             wmanIf2SsOfdmaBandAMCReleaseTimer,
51             wmanIf2SsOfdmaBandStatRepMAXPeriod,
52             wmanIf2SsOfdmaBandAMCRetryTimer,
53             wmanIf2SsOfdmaSafetyChAllocThreshold,
54             wmanIf2SsOfdmaSafetyChReleaseThreshold,
55             wmanIf2SsOfdmaSafetyChAllocTimer,
56             wmanIf2SsOfdmaSafetyChReleaseTimer,
57
58
59
60
61
62
63
64
65

```



```

1      wmanIf2SsOfdmaBinStatRepMAXPeriod,
2      wmanIf2SsOfdmaSafetyChaRetryTimer,
3      wmanIf2SsOfdmaHARQAackDelayULBurst,
4      wmanIf2SsOfdmaCQICHBandAMCTranaDelay,
5      wmanIf2SsOfdmaBsEIRP,
6      wmanIf2SsOfdmaChannelNumber,
7      wmanIf2SsOfdmaTTG,
8      wmanIf2SsOfdmaRTG,
9      wmanIf2SsOfdmaInitRngMaxRSS,
10     wmanIf2SsOfdmaDownlinkCenterFreq,
11     wmanIf2SsOfdmaBsId,
12     wmanIf2SsOfdmaMacVersion,
13     wmanIf2SsOfdmaFrameDurationCode,
14     wmanIf2SsOfdmaSizeCqichIdField,
15     wmanIf2SsOfdmaHARQAackDelayBurst,
16     wmanIf2SsOfdmaUiucIndex,
17     wmanIf2SsOfdmaUcdFecCodeType,
18     wmanIf2SsOfdmaRangingDataRatio,
19     wmanIf2SsOfdmaNorCOVerNOVERRIDE,
20     wmanIf2SsOfdmaDiucIndex,
21     wmanIf2SsOfdmaDownlinkFrequency,
22     wmanIf2SsOfdmaDcdFecCodeType,
23     wmanIf2SsOfdmaDiucMandatoryExitThresh,
24     wmanIf2SsOfdmaDiucMinEntryThresh}
25
26     STATUS          current
27
28     DESCRIPTION
29         "This group contains objects for SS and OFDMA PHY."
30     ::= { wmanIf2MibGroups 9 }
31
32 wmanIf2MibBsNotificationGroup      NOTIFICATION-GROUP
33     NOTIFICATIONS {wmanIf2BsSsStatusNotificationTrap,
34                   wmanIf2BsSsDynamicServiceFailTrap,
35                   wmanIf2BsSsRssiStatusChangeTrap,
36                   wmanIf2BsSsPkmFailTrap,
37                   wmanIf2BsSsRegistrerTrap}
38
39     STATUS          current
40
41     DESCRIPTION
42         "This group contains event notifications for BS."
43     ::= { wmanIf2MibGroups 10 }
44
45 wmanIf2MibSsNotificationGroup      NOTIFICATION-GROUP
46     NOTIFICATIONS {wmanIf2SsTlvUnknownTrap,
47                   wmanIf2SsDynamicServiceFailTrap,
48                   wmanIf2SsDhcpSuccessTrap,
49                   wmanIf2SsRssiStatusChangeTrap}
50
51     STATUS          current
52
53     DESCRIPTION
54         "This group contains event notifications for SS."
55     ::= { wmanIf2MibGroups 11 }
56
57 wmanIf2MibCmnPhsGroup              OBJECT-GROUP
58     OBJECTS {-- Payload header supression
59             wmanIf2CmnPhsRulePhsField,
60             wmanIf2CmnPhsRulePhsMask,
61
62
63
64
65

```

```

1          wmanIf2CmnPhsRulePhsSize,
2          wmanIf2CmnPhsRulePhsVerify}
3
4      STATUS          current
5
6      DESCRIPTION
7          "This group contains common objects for PHS."
8
9      ::= { wmanIf2MibGroups 12 }
10
11 wmanIf2MibBsPhsGroup      OBJECT-GROUP
12     OBJECTS {-- Payload header supression
13         wmanIf2BsClassifierRulePhsSize,
14         wmanIf2BsClassifierRulePhsMask,
15         wmanIf2BsClassifierRulePhsVerify,
16         wmanIf2BsClassifierRuleBitMap}
17
18     STATUS          current
19
20     DESCRIPTION
21         "This group contains BS objects for PHS."
22
23     ::= { wmanIf2MibGroups 13 }
24
25 END
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

### 13.2.4 wmanIf2mMib

```

1
2
3
4 WMAN-IF2M-MIB DEFINITIONS ::= BEGIN
5
6     IMPORTS
7         MODULE-IDENTITY,
8         OBJECT-TYPE,
9         NOTIFICATION-TYPE,
10        Unsigned32, Integer32, Counter32,
11        Counter64, transmission
12            FROM SNMPv2-SMI
13        SnmpAdminString
14            FROM SNMP-FRAMEWORK-MIB
15        TEXTUAL-CONVENTION,
16        MacAddress, RowStatus, TruthValue,
17        TimeStamp, DateAndTime
18            FROM SNMPv2-TC
19        InetAddressType, InetAddress
20            FROM INET-ADDRESS-MIB
21        OBJECT-GROUP,
22        MODULE-COMPLIANCE,
23        NOTIFICATION-GROUP
24            FROM SNMPv2-CONF
25        ifIndex
26            FROM IF-MIB;
27
28
29
30
31
32 wmanIf2mMib MODULE-IDENTITY
33     LAST-UPDATED      "200611280000Z" -- November 28, 2006
34     ORGANIZATION      "IEEE 802.16"
35     CONTACT-INFO
36         "WG E-mail: stds-802-16@ieee.org
37         WG Chair: Roger B. Marks
38         Postal: NextWave Broadband, Inc.
39         E-mail: r.b.marks@ieee.org
40
41         TGF Chair: Phillip Barber
42         Postal: Huawei Technologies Co., Ltd
43         E-mail: pbarber@futurewei.com
44
45         Editor: Joey Chou
46         Postal: Intel Corporation
47         5000 W. Chandler Blvd,
48         Chandler, AZ 85227, USA
49         E-mail: joey.chou@intel.com"
50
51
52
53     DESCRIPTION
54         "This material is from IEEE Std 802.16i
55         Copyright (c) 2006 IEEE.
56         This MIB Module defines managed objects for
57         Subscriber Station and Base Station based on IEEE Std
58         802.16-2004 and its amendment IEEE Std 802.16e-2005.
59         The MIB contains managed objects that are specific
60         to mobile Broadband Wireless Networks."
61
62
63     REVISION          "20061120000Z"
64
65     DESCRIPTION

```

```

1         "WMAN-IF2M-MIB module that is included in
2         IEEE 802.16i-06/001r5."
3     REVISION         "200610160000Z"
4     DESCRIPTION
5         "The 1st revision of WMAN-IF2M-MIB module."
6     ::= { iso std(0) iso8802(8802) wman(16) 3 }
7
8
9
10    wmanIf2mMibObjects    OBJECT IDENTIFIER ::= { wmanIf2mMib 1 }
11    wmanIf2mBsObjects     OBJECT IDENTIFIER ::= { wmanIf2mMibObjects 1 }
12    wmanIf2mSsObjects     OBJECT IDENTIFIER ::= { wmanIf2mMibObjects 2 }
13    wmanIf2mCommonObjects OBJECT IDENTIFIER ::= { wmanIf2mMibObjects 3 }
14
15
16    --
17    -- Textual Conventions
18    --
19    WmanIf2mOfdmaMobility ::= TEXTUAL-CONVENTION
20        STATUS         current
21        DESCRIPTION
22            "This field indicates whether or not the MS supports
23            mobility hand-over, Sleepmode, and Idle-mode. A bit
24            value of 0 indicates 'not supported' while 1 indicates
25            it is supported."
26        REFERENCE
27            "Subclause 11.7.14.1 in IEEE Std 802.16e-2005"
28        SYNTAX         BITS { handoverSupport(0),
29                        sleepModeSupport(1),
30                        idleModeSupport(2) }
31
32
33
34    WmanIf2mHandoverType ::= TEXTUAL-CONVENTION
35        STATUS         current
36        DESCRIPTION
37            "Indicates what type(s) of Handover the BS and the MS
38            supports.
39            bit#0: when it is set to 1, MDHO/FBSS HO not supported.
40            the BS shall ignore all other bits.
41            bit#1: when it is set to 1, FBSS/MDHO DL RF Combining
42            is supported with monitoring MAPs from active BSs
43            bit#2: when it is set to 1, MDHO DL soft Combining is
44            supported with monitoring single MAP from
45            anchor BS
46            bit#3: when it is set to 1, MDHO DL soft combining is
47            supported with monitoring MAPs from active BSs
48            bit#3: when it is set to 1, MDHO UL Multiple
49            transmission is supported"
50        REFERENCE
51            "Subclause 11.7.8.12 in IEEE Std 802.16e-2005"
52        SYNTAX         BITS { mdhcFbssHoNotSupported(0),
53                        mdhcFbssDlMapsFromActiveBss(1),
54                        mdhcDlMapFromAnchorBs(2),
55                        mdhcDlMapsFromActiveBss(3),
56                        mdhcUlMultipleTx(4) }
57
58
59
60
61
62
63    WmanIf2mCidType ::= TEXTUAL-CONVENTION
64        STATUS         current
65

```

```

1      DESCRIPTION
2          "Type of CID."
3      SYNTAX      INTEGER (0 .. 65535)
4
5
6      WmanIf2mPsClassId ::= TEXTUAL-CONVENTION
7          STATUS      current
8      DESCRIPTION
9          "Indicates the index to Power Saving Classes. The ID shall
10         be unique within the group of Power Saving Classes
11         associated with the MS. This ID may be used in further
12         MOB_SLP-REQ/RSP messages for activation / deactivation
13         of Power Saving Class."
14      REFERENCE
15          "Subclause 6.3.2.3.44 in IEEE Std 802.16e-2005"
16      SYNTAX      INTEGER (0..63)
17
18
19
20      WmanIf2mPsClassType ::= TEXTUAL-CONVENTION
21          STATUS      current
22      DESCRIPTION
23          "The types of power saving classes."
24      REFERENCE
25          "Table 374a in IEEE Std 802.16e-2005"
26      SYNTAX      INTEGER {powerSavingClassTypeI(1),
27                    powerSavingClassTypeII(2),
28                    powerSavingClassTypeIII(3)}
29
30
31
32      WmanIf2mPsClassCidDir ::= TEXTUAL-CONVENTION
33          STATUS      current
34      DESCRIPTION
35          "The direction of power saving class's CIDs.
36          0b00 = Unspecified. Each CID has its own direction
37          assign in its connection creation. Can be
38          DL, UL, or both (in the case of management
39          connections).
40          0b01 = Downlink direction only.
41          0b10 = Uplink direction only."
42      REFERENCE
43          "Subclause 6.3.2.3.44 in IEEE Std 802.16e-2005"
44      SYNTAX      INTEGER {unspecified(0),
45                    downlink(1),
46                    uplink(2)}
47
48
49
50
51      WmanIf2mPowerSavingMode ::= TEXTUAL-CONVENTION
52          STATUS      current
53      DESCRIPTION
54          "Power saving class mode active or not active."
55      REFERENCE
56          "Subclause 6.3.2.3.44 in IEEE Std 802.16e-2005"
57      SYNTAX      INTEGER {psNotActive(0),
58                    psActive(1)}
59
60
61
62      --
63      -- wmanIf2mBsObjects - containing tables and objects to be implemented
64      -- in the Base station
65

```

```

1  --
2  -- wmanIf2mBsCm contain the Base Station Configuration Management
3  -- objects
4  --
5
6  wmanIf2mBsCm OBJECT IDENTIFIER ::= { wmanIf2mBsObjects 1 }
7
8  --
9  -- Base Station capabilities
10 --
11 --
12 wmanIf2mBsCapabilities OBJECT IDENTIFIER ::= { wmanIf2mBsCm 1 }
13
14 wmanIf2mBsSsReqCapabilitiesTable OBJECT-TYPE
15     SYNTAX      SEQUENCE OF WmanIf2mBsSsReqCapabilitiesEntry
16     MAX-ACCESS  not-accessible
17     STATUS      current
18     DESCRIPTION
19         "This table contains the SS's capabilities that are necessary
20         for supporting mobility. SS reports these capabilities in
21         the REG-REQ messages."
22     REFERENCE
23         "Subclause 6.3.2.3.7 in IEEE Std 802.16e-2005"
24     ::= { wmanIf2mBsCapabilities 1 }
25
26
27
28
29 wmanIf2mBsSsReqCapabilitiesEntry OBJECT-TYPE
30     SYNTAX      WmanIf2mBsSsReqCapabilitiesEntry
31     MAX-ACCESS  not-accessible
32     STATUS      current
33     DESCRIPTION
34         "This table provides one row for each SS that has entered and
35         registered into the BS. The primary index is the ifIndex
36         with an ifType of ieee80216WMAN, indicating the BS sector
37         with which the SS is associated. wmanIf2mBsSsMacAddress
38         identifies the SS being registered."
39     INDEX { ifIndex, wmanIf2mBsSsMacAddress }
40     ::= { wmanIf2mBsSsReqCapabilitiesTable 1 }
41
42
43
44 WmanIf2mBsSsReqCapabilitiesEntry ::= SEQUENCE {
45     wmanIf2mBsSsMacAddress          MacAddress,
46     wmanIf2mBsSsReqCapHandoverSupported  WmanIf2mHandoverType,
47     wmanIf2mBsSsReqCapHoProcessTimer    Unsigned32,
48     wmanIf2mBsSsReqCapMobilityFeature    WmanIf2mOfdmaMobility,
49     wmanIf2mBsSsReqCapSleepRecoveryTime Unsigned32,
50     wmanIf2mBsSsReqCapPreviousIpAddr    OCTET STRING,
51     wmanIf2mBsSsReqCapIdleModeTimeout   Unsigned32,
52     wmanIf2mBsSsReqCapHoConnProcessTime Unsigned32,
53     wmanIf2mBsSsReqCapHoTekProcessTime   Unsigned32 }
54
55
56
57 wmanIf2mBsSsMacAddress OBJECT-TYPE
58     SYNTAX      MacAddress
59     MAX-ACCESS  not-accessible
60     STATUS      current
61     DESCRIPTION
62         "The MAC address of SS is received from the RNG-REQ
63         message, and used as the identifier to the SS."
64
65

```

```

1      REFERENCE
2          "Subclause 6.3.2.3.5 in IEEE Std 802.16e-2005"
3      ::= { wmanIf2mBsSsReqCapabilitiesEntry 1 }
4
5
6      wmanIf2mBsSsReqCapHandoverSupported OBJECT-TYPE
7          SYNTAX      WmanIf2mHandoverType
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "Indicates what type(s) of Handover the BS or MS supports."
12         REFERENCE
13             "Subclause 11.7.12 in IEEE Std 802.16e-2005"
14         ::= { wmanIf2mBsSsReqCapabilitiesEntry 2 }
15
16
17
18         wmanIf2mBsSsReqCapHoProcessTimer OBJECT-TYPE
19             SYNTAX      Unsigned32
20             UNITS      "frames"
21             MAX-ACCESS  read-only
22             STATUS      current
23             DESCRIPTION
24                 "The duration in frames the MS shall wait until receipt of
25                 the next unsolicited network re-entry MAC management
26                 message as indicated in the HO Process Optimization
27                 element of the RNG-RSP message."
28             REFERENCE
29                 "Subclause 11.7.13.2 in IEEE Std 802.16e-2005"
30             ::= { wmanIf2mBsSsReqCapabilitiesEntry 3 }
31
32
33
34
35         wmanIf2mBsSsReqCapMobilityFeature OBJECT-TYPE
36             SYNTAX      WmanIf2mOfdmaMobility
37             MAX-ACCESS  read-only
38             STATUS      current
39             DESCRIPTION
40                 "The field indicates whether or not the MS supports
41                 mobility hand-over, Sleepmode, and Idle-mode."
42             REFERENCE
43                 "Subclause 11.7.14.1 in IEEE Std 802.16e-2005"
44             ::= { wmanIf2mBsSsReqCapabilitiesEntry 4 }
45
46
47
48         wmanIf2mBsSsReqCapSleepRecoveryTime OBJECT-TYPE
49             SYNTAX      Unsigned32
50             UNITS      "frames"
51             MAX-ACCESS  read-only
52             STATUS      current
53             DESCRIPTION
54                 "The object indicates the time required for an MS that is
55                 in a sleep mode to return to awake-mode. This may be used
56                 by the BS to determine sleep interval window sizes when
57                 initiating sleep mode with an MS."
58             REFERENCE
59                 "Subclause 11.7.15 in IEEE Std 802.16e-2005"
60             ::= { wmanIf2mBsSsReqCapabilitiesEntry 5 }
61
62
63
64
65         wmanIf2mBsSsReqCapPreviousIpAddr OBJECT-TYPE

```

```

1      SYNTAX      OCTET STRING
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "The object indicates the IP address that the MS was assigned
6          on the secondary management connection based on an
7          association with its last serving BS. An IPv4 address shall
8          be specified in conventional dotted format; e.g.,
9          '134.234.2.3'. An IPv6 address may be expressed in abridged
10         or unbridged form; however, the form chosen shall be
11         consistent with RFC 2373."
12
13     REFERENCE
14         "Subclause 11.7.16 in IEEE Std 802.16e-2005"
15     ::= { wmanIf2mBsSsReqCapabilitiesEntry 6 }
16
17 wmanIf2mBsSsReqCapIdleModeTimeout OBJECT-TYPE
18     SYNTAX      Unsigned32
19     UNITS        "seconds"
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "Max time interval between MS Idle Mode Location Updates."
24     REFERENCE
25         "Subclause 11.7.20.1 in IEEE Std 802.16e-2005"
26     ::= { wmanIf2mBsSsReqCapabilitiesEntry 7 }
27
28 wmanIf2mBsSsReqCapHoConnProcessTime OBJECT-TYPE
29     SYNTAX      Unsigned32
30     UNITS        "milliseconds"
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "The duration that the MS needs to process information
35         on connections provided in RNG-RSP or REG-RSP message
36         during Handoff."
37     REFERENCE
38         "Subclause 11.7.24 in IEEE Std 802.16e-2005"
39     ::= { wmanIf2mBsSsReqCapabilitiesEntry 8 }
40
41 wmanIf2mBsSsReqCapHoTekProcessTime OBJECT-TYPE
42     SYNTAX      Unsigned32
43     UNITS        "milliseconds"
44     MAX-ACCESS  read-only
45     STATUS      current
46     DESCRIPTION
47         "The duration that the MS needs to completely process
48         TEK information during Handoff."
49     REFERENCE
50         "Subclause 11.7.24 in IEEE Std 802.16e-2005"
51     ::= { wmanIf2mBsSsReqCapabilitiesEntry 9 }
52
53 wmanIf2mBsSsRspCapabilitiesTable OBJECT-TYPE
54     SYNTAX      SEQUENCE OF WmanIf2mBsSsRspCapabilitiesEntry
55     MAX-ACCESS  not-accessible
56
57
58
59
60
61
62
63
64
65

```



```

1      STATUS      current
2      DESCRIPTION
3          "This table contains the SS's capabilities that are necessary
4          for supporting mobility. BS acknowledges the capabilities in
5          the REG-RSP message in response to REG-REQ messages."
6      REFERENCE
7          "Subclause 6.3.2.3.7 in IEEE Std 802.16e-2005"
8      ::= { wmanIf2mBsCapabilities 2 }
9
10
11
12 wmanIf2mBsSsRspCapabilitiesEntry OBJECT-TYPE
13     SYNTAX      WmanIf2mBsSsRspCapabilitiesEntry
14     MAX-ACCESS  not-accessible
15     STATUS      current
16     DESCRIPTION
17         "This table provides one row for each SS that has entered and
18         registered into the BS. The primary index is the ifIndex
19         with an ifType of ieee80216WMAN, indicating the BS sector
20         with which the SS is associated. wmanIf2mBsSsMacAddress
21         identifies the SS being registered."
22     INDEX { ifIndex, wmanIf2mBsSsMacAddress }
23     ::= { wmanIf2mBsSsRspCapabilitiesTable 1 }
24
25
26
27 WmanIf2mBsSsRspCapabilitiesEntry ::= SEQUENCE {
28     wmanIf2mBsSsRspCapHandoverSupported      WmanIf2mHandoverType,
29     wmanIf2mBsSsRspCapRetrainTime            Unsigned32,
30     wmanIf2mBsSsRspCapHoProcessTimer        Unsigned32,
31     wmanIf2mBsSsRspCapRetransmissionTimer    Unsigned32,
32     wmanIf2mBsSsRspCapMobilityFeature        WmanIf2mOfdmaMobility,
33     wmanIf2mBsSsRspCapNewSaid                Integer32,
34     wmanIf2mBsSsRspCapOldSaid                Integer32,
35     wmanIf2mBsSsRspCapIdleModeTimeout        Unsigned32,
36     wmanIf2mBsSsRspCapHoConnProcessTime      Unsigned32,
37     wmanIf2mBsSsRspCapHoTekProcessTime       Unsigned32 }
38
39
40
41
42 wmanIf2mBsSsRspCapHandoverSupported OBJECT-TYPE
43     SYNTAX      WmanIf2mHandoverType
44     MAX-ACCESS  read-only
45     STATUS      current
46     DESCRIPTION
47         "Indicates what type(s) of Handover the BS or MS supports."
48     REFERENCE
49         "Subclause 11.7.12 in IEEE Std 802.16e-2005"
50     ::= { wmanIf2mBsSsRspCapabilitiesEntry 1 }
51
52
53
54 wmanIf2mBsSsRspCapRetrainTime OBJECT-TYPE
55     SYNTAX      Unsigned32
56     UNITS       "100 milliseconds"
57     MAX-ACCESS  read-only
58     STATUS      current
59     DESCRIPTION
60         "Indicates the duration for MS's connection information that
61         will be retained in serving BS. BS shall start
62         Resource_Retain_Time timer at MS notification of pending HO
63         attempt through MOB_HO-IND or by detecting an MS drop."
64
65

```

```

1      REFERENCE
2          "Subclause 11.7.13.1 in IEEE Std 802.16e-2005"
3      ::= { wmanIf2mBsSsRspCapabilitiesEntry 2 }
4
5
6      wmanIf2mBsSsRspCapHoProcessTimer OBJECT-TYPE
7          SYNTAX      Unsigned32
8          UNITS       "frames"
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "The duration in frames the MS shall wait until receipt of
13             the next unsolicited network re-entry MAC management
14             message as indicated in the HO Process Optimization
15             element of the RNG-RSP message. On HO Process Optimization
16             MS Timer timeout and while HO Process Optimization MS
17             Timer Retries is valid, MS shall send the network re-entry
18             MAC management request message corresponding to the
19             expected and pending network re-entry MAC management
20             response message as indicated in HO Process Optimization
21             and recycle HO Process Optimization MS Timer."
22
23         REFERENCE
24             "Subclause 11.7.13.2 in IEEE Std 802.16e-2005"
25         ::= { wmanIf2mBsSsRspCapabilitiesEntry 3 }
26
27
28         wmanIf2mBsSsRspCapRetransmissionTimer OBJECT-TYPE
29             SYNTAX      Unsigned32
30             UNITS       "frames"
31             MAX-ACCESS  read-only
32             STATUS      current
33             DESCRIPTION
34                 "When an MS transmits MOB_MSHO-REQ to initiate a handover
35                 process, it shall start MS Handover Retransmission Timer
36                 and shall not transmit another MOB_MSHO-REQ until the
37                 expiration of the MS Handover Retransmission Timer."
38
39             REFERENCE
40                 "Subclause 11.7.13.3 in IEEE Std 802.16e-2005"
41             ::= { wmanIf2mBsSsRspCapabilitiesEntry 4 }
42
43
44         wmanIf2mBsSsRspCapMobilityFeature OBJECT-TYPE
45             SYNTAX      WmanIf2mOfdmaMobility
46             MAX-ACCESS  read-only
47             STATUS      current
48             DESCRIPTION
49                 "The field indicates the mobility hand-over, Sleepmode,
50                 and Idle-mode negotiated for MS."
51
52             REFERENCE
53                 "Subclause 11.7.14.1 in IEEE Std 802.16e-2005"
54             ::= { wmanIf2mBsSsRspCapabilitiesEntry 5 }
55
56
57         wmanIf2mBsSsRspCapNewSaid OBJECT-TYPE
58             SYNTAX      Integer32 (0 .. 65535)
59             MAX-ACCESS  read-only
60             STATUS      current
61             DESCRIPTION
62
63
64
65

```

```

1         "The field indicates New SAID after handover to new BS. It
2         provides a translation table that allows an MS to update
3         its security associations so that it may continue security
4         service after a handover to a new serving BS."
5
6     REFERENCE
7         "Subclause 11.7.18 in IEEE Std 802.16e-2005"
8     ::= { wmanIf2mBsSsRspCapabilitiesEntry 6 }
9
10
11 wmanIf2mBsSsRspCapOldSaid OBJECT-TYPE
12     SYNTAX      Integer32 (0 .. 65535)
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "The field indicates Old SAID after handover to new BS. It
17         provides a translation table that allows an MS to update
18         its security associations so that it may continue security
19         service after a handover to a new serving BS."
20
21     REFERENCE
22         "Subclause 11.7.18 in IEEE Std 802.16e-2005"
23     ::= { wmanIf2mBsSsRspCapabilitiesEntry 7 }
24
25
26 wmanIf2mBsSsRspCapIdleModeTimeout OBJECT-TYPE
27     SYNTAX      Unsigned32
28     UNITS       "seconds"
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32         "Max time interval between MS Idle Mode Location Updates."
33
34     REFERENCE
35         "Subclause 11.7.20.1 in IEEE Std 802.16e-2005"
36     ::= { wmanIf2mBsSsRspCapabilitiesEntry 8 }
37
38
39 wmanIf2mBsSsRspCapHoConnProcessTime OBJECT-TYPE
40     SYNTAX      Unsigned32
41     UNITS       "milliseconds"
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45         "The duration that the MS needs to process information
46         on connections provided in RNG-RSP or REG-RSP message
47         during Handoff."
48
49     REFERENCE
50         "Subclause 11.7.24 in IEEE Std 802.16e-2005"
51     ::= { wmanIf2mBsSsRspCapabilitiesEntry 9 }
52
53
54
55 wmanIf2mBsSsRspCapHoTekProcessTime OBJECT-TYPE
56     SYNTAX      Unsigned32
57     UNITS       "milliseconds"
58     MAX-ACCESS  read-only
59     STATUS      current
60     DESCRIPTION
61         "The duration that the MS needs to completely process
62         TEK information during Handoff."
63
64     REFERENCE
65

```

```

1      "Subclause 11.7.24 in IEEE Std 802.16e-2005"
2      ::= { wmanIf2mBsSsRspCapabilitiesEntry 10 }
3
4
5  wmanIf2mBsBasicCapabilitiesTable OBJECT-TYPE
6      SYNTAX      SEQUENCE OF WmanIf2mBsBasicCapabilitiesEntry
7      MAX-ACCESS  not-accessible
8      STATUS      current
9      DESCRIPTION
10         "This table contains the basic capabilities of the BS as
11         implemented in BS hardware and software. These capabilities
12         along with the configuration for them
13         (wmanIf2mBsCapabilitiesConfigTable) are used for negotiation
14         of basic capabilities with SS using RNG-RSP, SBC-RSP and
15         REG-RSP messages. The negotiated capabilities are obtained
16         by interSubclause of SS raw reported capabilities, BS raw
17         capabilities and BS configured capabilities. The objects in
18         the table have read-only access. The table is maintained
19         by BS."
20
21     ::= { wmanIf2mBsCapabilities 3 }
22
23
24
25  wmanIf2mBsBasicCapabilitiesEntry OBJECT-TYPE
26      SYNTAX      WmanIf2mBsBasicCapabilitiesEntry
27      MAX-ACCESS  not-accessible
28      STATUS      current
29      DESCRIPTION
30         "This table provides one row for each BS sector and is
31         indexed by ifIndex."
32
33     INDEX { ifIndex }
34
35     ::= { wmanIf2mBsBasicCapabilitiesTable 1 }
36
37  WmanIf2mBsBasicCapabilitiesEntry ::= SEQUENCE {
38      wmanIf2mBsCapHandoverSupported      WmanIf2mHandoverType,
39      wmanIf2mBsCapRetrainTime            Unsigned32,
40      wmanIf2mBsCapHoProcessTimer        Unsigned32,
41      wmanIf2mBsCapRetransmissionTimer    Unsigned32,
42      wmanIf2mBsCapMobilityFeature        WmanIf2mOfdmaMobility,
43      wmanIf2mBsCapIdleModeTimeout        Unsigned32,
44      wmanIf2mBsCapHoConnProcessTime     Unsigned32,
45      wmanIf2mBsCapHoTekProcessTime      Unsigned32}
46
47
48
49  wmanIf2mBsCapHandoverSupported OBJECT-TYPE
50      SYNTAX      WmanIf2mHandoverType
51      MAX-ACCESS  read-only
52      STATUS      current
53      DESCRIPTION
54         "Indicates what type(s) of Handover the BS or MS supports."
55
56     REFERENCE
57         "Subclause 11.7.12 in IEEE Std 802.16e-2005"
58
59     ::= { wmanIf2mBsBasicCapabilitiesEntry 1 }
60
61  wmanIf2mBsCapRetrainTime OBJECT-TYPE
62      SYNTAX      Unsigned32
63      UNITS       "100 milliseconds"
64      MAX-ACCESS  read-only
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "Indicates the duration for MS's connection information that
4          will be retained in serving BS. BS shall start
5          Resource_Retain_Time timer at MS notification of pending HO
6          attempt through MOB_HO-IND or by detecting an MS drop."
7      REFERENCE
8          "Subclause 11.7.13.1 in IEEE Std 802.16e-2005"
9      ::= { wmanIf2mBsBasicCapabilitiesEntry 2 }
10
11
12
13      wmanIf2mBsCapHoProcessTimer OBJECT-TYPE
14          SYNTAX      Unsigned32
15          UNITS       "frames"
16          MAX-ACCESS  read-only
17          STATUS      current
18          DESCRIPTION
19              "The duration in frames the MS shall wait until receipt of
20              the next unsolicited network re-entry MAC management
21              message as indicated in the HO Process Optimization
22              element of the RNG-RSP message."
23          REFERENCE
24              "Subclause 11.7.13.2 in IEEE Std 802.16e-2005"
25          ::= { wmanIf2mBsBasicCapabilitiesEntry 3 }
26
27
28
29
30      wmanIf2mBsCapRetransmissionTimer OBJECT-TYPE
31          SYNTAX      Unsigned32
32          UNITS       "frames"
33          MAX-ACCESS  read-only
34          STATUS      current
35          DESCRIPTION
36              "When an MS transmits MOB_MSHO-REQ to initiate a handover
37              process, it shall start MS Handover Retransmission Timer
38              and shall not transmit another MOB_MSHO-REQ until the
39              expiration of the MS Handover Retransmission Timer."
40          REFERENCE
41              "Subclause 11.7.13.3 in IEEE Std 802.16e-2005"
42          ::= { wmanIf2mBsBasicCapabilitiesEntry 4 }
43
44
45
46
47      wmanIf2mBsCapMobilityFeature OBJECT-TYPE
48          SYNTAX      WmanIf2mOfdmaMobility
49          MAX-ACCESS  read-only
50          STATUS      current
51          DESCRIPTION
52              "The field indicates the mobility hand-over, Sleepmode,
53              and Idle-mode supported by BS."
54          REFERENCE
55              "Subclause 11.7.14.1 in IEEE Std 802.16e-2005"
56          ::= { wmanIf2mBsBasicCapabilitiesEntry 5 }
57
58
59
60      wmanIf2mBsCapIdleModeTimeout OBJECT-TYPE
61          SYNTAX      Unsigned32
62          UNITS       "seconds"
63          MAX-ACCESS  read-only
64          STATUS      current
65

```

```

1      DESCRIPTION
2          "Max time interval between MS Idle Mode Location Updates."
3      REFERENCE
4          "Subclause 11.7.20.1 in IEEE Std 802.16e-2005"
5          ::= { wmanIf2mBsBasicCapabilitiesEntry 6 }
6
7
8      wmanIf2mBsCapHoConnProcessTime OBJECT-TYPE
9          SYNTAX      Unsigned32
10         UNITS       "milliseconds"
11         MAX-ACCESS  read-only
12         STATUS      current
13         DESCRIPTION
14             "The duration that the MS needs to process information
15              on connections provided in RNG-RSP or REG-RSP message
16              during Handoff."
17         REFERENCE
18             "Subclause 11.7.24 in IEEE Std 802.16e-2005"
19             ::= { wmanIf2mBsBasicCapabilitiesEntry 7 }
20
21
22      wmanIf2mBsCapHoTekProcessTime OBJECT-TYPE
23         SYNTAX      Unsigned32
24         UNITS       "milliseconds"
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "The duration that the MS needs to completely process
29              TEK information during Handoff."
30         REFERENCE
31             "Subclause 11.7.24 in IEEE Std 802.16e-2005"
32             ::= { wmanIf2mBsBasicCapabilitiesEntry 8 }
33
34
35      wmanIf2mBsCapabilitiesConfigTable OBJECT-TYPE
36         SYNTAX      SEQUENCE OF WmanIf2mBsCapabilitiesConfigEntry
37         MAX-ACCESS  not-accessible
38         STATUS      current
39         DESCRIPTION
40             "This table contains the configuration for basic
41              capabilities of BS. The table is intended to be used to
42              restrict the Capabilities implemented by BS, for example in
43              order to comply with local regulatory requirements. The BS
44              should use the configuration along with the implemented
45              Capabilities (wmanIf2mBsBasicCapabilitiesTable) for
46              negotiation of basic capabilities with SS using RNG-RSP,
47              SBC-RSP and REG-RSP messages. The negotiated capabilities
48              are obtained by interSubclause of SS reported capabilities,
49              BS raw capabilities and BS configured capabilities. The
50              objects in the table have read-write access. The rows are
51              created by BS as a copy of wmanIf2mBsBasicCapabilitiesTable
52              and can be modified by NMS."
53             ::= { wmanIf2mBsCapabilities 4 }
54
55
56      wmanIf2mBsCapabilitiesConfigEntry OBJECT-TYPE
57         SYNTAX      WmanIf2mBsCapabilitiesConfigEntry
58         MAX-ACCESS  not-accessible
59
60
61
62
63
64
65

```

```

1      STATUS      current
2      DESCRIPTION
3          "This table provides one row for each BS sector and is
4          indexed by ifIndex."
5      INDEX { ifIndex }
6      ::= { wmanIf2mBsCapabilitiesConfigTable 1 }
7
8
9
10     WmanIf2mBsCapabilitiesConfigEntry ::= SEQUENCE {
11         wmanIf2mBsCapCfgHandoverSupported      WmanIf2mHandoverType,
12         wmanIf2mBsCapCfgRetrainTime            Unsigned32,
13         wmanIf2mBsCapCfgHoProcessTimer         Unsigned32,
14         wmanIf2mBsCapCfgRetransmissionTimer    Unsigned32,
15         wmanIf2mBsCapCfgMobilityFeature        WmanIf2mOfdmaMobility,
16         wmanIf2mBsCapCfgIdleModeTimeout       Unsigned32,
17         wmanIf2mBsCapCfgHoConnProcessTime      Unsigned32,
18         wmanIf2mBsCapCfgHoTekProcessTime       Unsigned32}
19
20
21
22     wmanIf2mBsCapCfgHandoverSupported OBJECT-TYPE
23         SYNTAX      WmanIf2mHandoverType
24         MAX-ACCESS  read-write
25         STATUS      current
26         DESCRIPTION
27             "Indicates what type(s) of Handover the BS or MS supports."
28         REFERENCE
29             "Subclause 11.7.12 in IEEE Std 802.16e-2005"
30         ::= { wmanIf2mBsCapabilitiesConfigEntry 1 }
31
32
33
34     wmanIf2mBsCapCfgRetrainTime OBJECT-TYPE
35         SYNTAX      Unsigned32
36         UNITS       "100 milliseconds"
37         MAX-ACCESS  read-write
38         STATUS      current
39         DESCRIPTION
40             "Indicates the duration for MS's connection information that
41             will be retained in serving BS. BS shall start
42             Resource_Retain_Time timer at MS notification of pending HO
43             attempt through MOB_HO-IND or by detecting an MS drop."
44         REFERENCE
45             "Subclause 11.7.13.1 in IEEE Std 802.16e-2005"
46         DEFVAL     { 1 }
47         ::= { wmanIf2mBsCapabilitiesConfigEntry 2 }
48
49
50
51
52     wmanIf2mBsCapCfgHoProcessTimer OBJECT-TYPE
53         SYNTAX      Unsigned32
54         UNITS       "frames"
55         MAX-ACCESS  read-write
56         STATUS      current
57         DESCRIPTION
58             "The duration in frames the MS shall wait until receipt of
59             the next unsolicited network re-entry MAC management
60             message as indicated in the HO Process Optimization
61             element of the RNG-RSP message."
62         REFERENCE
63             "Subclause 11.7.13.2 in IEEE Std 802.16e-2005"
64
65

```

```

1      ::= { wmanIf2mBsCapabilitiesConfigEntry 3 }
2
3
4  wmanIf2mBsCapCfgRetransmissionTimer OBJECT-TYPE
5      SYNTAX      Unsigned32
6      UNITS       "frames"
7      MAX-ACCESS  read-write
8      STATUS      current
9      DESCRIPTION
10         "When an MS transmits MOB_MSHO-REQ to initiate a handover
11         process, it shall start MS Handover Retransmission Timer
12         and shall not transmit another MOB_MSHO-REQ until the
13         expiration of the MS Handover Retransmission Timer."
14     REFERENCE
15         "Subclause 11.7.13.3 in IEEE Std 802.16e-2005"
16     ::= { wmanIf2mBsCapabilitiesConfigEntry 4 }
17
18
19
20  wmanIf2mBsCapCfgMobilityFeature OBJECT-TYPE
21      SYNTAX      WmanIf2mOfdmaMobility
22      MAX-ACCESS  read-write
23      STATUS      current
24      DESCRIPTION
25         "The field indicates the mobility hand-over, Sleepmode,
26         and Idle-mode configured for the BS."
27     REFERENCE
28         "Subclause 11.7.14.1 in IEEE Std 802.16e-2005"
29     ::= { wmanIf2mBsCapabilitiesConfigEntry 5 }
30
31
32
33  wmanIf2mBsCapCfgIdleModeTimeout OBJECT-TYPE
34      SYNTAX      Unsigned32
35      UNITS       "seconds"
36      MAX-ACCESS  read-write
37      STATUS      current
38      DESCRIPTION
39         "Max time interval between MS Idle Mode Location Updates."
40     REFERENCE
41         "Subclause 11.7.20.1 in IEEE Std 802.16e-2005"
42     DEFVAL      { 4096 }
43     ::= { wmanIf2mBsCapabilitiesConfigEntry 6 }
44
45
46
47  wmanIf2mBsCapCfgHoConnProcessTime OBJECT-TYPE
48      SYNTAX      Unsigned32
49      UNITS       "milliseconds"
50      MAX-ACCESS  read-write
51      STATUS      current
52      DESCRIPTION
53         "The duration that the MS needs to process information
54         on connections provided in RNG-RSP or REG-RSP message
55         during Handoff."
56     REFERENCE
57         "Subclause 11.7.24 in IEEE Std 802.16e-2005"
58     ::= { wmanIf2mBsCapabilitiesConfigEntry 7 }
59
60
61
62  wmanIf2mBsCapCfgHoTekProcessTime OBJECT-TYPE
63      SYNTAX      Unsigned32
64
65

```



```

1      UNITS          "milliseconds"
2      MAX-ACCESS    read-write
3      STATUS        current
4      DESCRIPTION
5          "The duration that the MS needs to completely process
6          TEK information during Handoff."
7      REFERENCE
8          "Subclause 11.7.24 in IEEE Std 802.16e-2005"
9      ::= { wmanIf2mBsCapabilitiesConfigEntry 8 }
10
11
12
13  --
14  -- Base Station Power Saving Mode
15  --
16  wmanIf2mBsPowerSavingMode OBJECT IDENTIFIER ::= { wmanIf2mBsCm 2 }
17
18
19  --
20  -- wmanIf2mBsSsPowerSavingStatusTable contains the power saving status
21  --
22
23  wmanIf2mBsSsPowerSavingStatusTable OBJECT-TYPE
24      SYNTAX          SEQUENCE OF WmanIf2mBsSsPowerSavingStatusEntry
25      MAX-ACCESS      not-accessible
26      STATUS          current
27      DESCRIPTION
28          "This table contains the power saving status for each CID
29          in an MS. When the MS roams to a different BS, all entries
30          associated with such MS will be deleted."
31      ::= { wmanIf2mBsPowerSavingMode 1 }
32
33
34
35  wmanIf2mBsSsPowerSavingStatusEntry OBJECT-TYPE
36      SYNTAX          WmanIf2mBsSsPowerSavingStatusEntry
37      MAX-ACCESS      not-accessible
38      STATUS          current
39      DESCRIPTION
40          "This table provides one row for each CID in an MS, and
41          is indexed by ifIndex, wmanIf2mBsSsMacAddress, and
42          wmanIf2mBsSsCid."
43      INDEX          { ifIndex,
44                      wmanIf2mBsSsMacAddress,
45                      wmanIf2mBsSsCid }
46      ::= { wmanIf2mBsSsPowerSavingStatusTable 1 }
47
48
49
50  WmanIf2mBsSsPowerSavingStatusEntry ::= SEQUENCE {
51      wmanIf2mBsSsCid          WmanIf2mCidType,
52      wmanIf2mBsSsPowerSavingClassId  WmanIf2mPsClassId }
53
54
55  wmanIf2mBsSsCid OBJECT-TYPE
56      SYNTAX          WmanIf2mCidType
57      MAX-ACCESS      read-only
58      STATUS          current
59      DESCRIPTION
60          "A 16 bit channel identifier to identify a connection."
61      ::= { wmanIf2mBsSsPowerSavingStatusEntry 1 }
62
63
64
65  wmanIf2mBsSsPowerSavingClassId OBJECT-TYPE

```

```

1      SYNTAX      WmanIf2mPsClassId
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "wmanIf2mBsSsPowerSavingClassId identifies the power
6          saving class associated with this CID. It maps to an
7          entry in wmanIf2mBsSsPowerSavingClassesTable."
8      ::= { wmanIf2mBsSsPowerSavingStatusEntry 2 }
9
10
11
12      --
13      -- wmanIf2mBsSsPowerSavingClassesTable contains the power saving classes
14      -- information
15      --
16      wmanIf2mBsSsPowerSavingClassesTable OBJECT-TYPE
17          SYNTAX      SEQUENCE OF WmanIf2mBsSsPowerSavingClassesEntry
18          MAX-ACCESS  not-accessible
19          STATUS      current
20          DESCRIPTION
21              "This table contains the power saving classes definitions,
22              and activation / deactivation information that are provided
23              by MOB_SLP-REQ and MOB_SLP-RSP messages. When the BS roams
24              to a different BS, all entries associated with such MS will
25              be deleted."
26          ::= { wmanIf2mBsPowerSavingMode 2 }
27
28
29      wmanIf2mBsSsPowerSavingClassesEntry OBJECT-TYPE
30          SYNTAX      WmanIf2mBsSsPowerSavingClassesEntry
31          MAX-ACCESS  not-accessible
32          STATUS      current
33          DESCRIPTION
34              "This table is indexed by ifIndex, wmanIf2mBsSsMacAddress,
35              and wmanIf2mBsSsPsClassesId. It is intended to support both
36              unicast and multicast service flows.
37              wmanIf2mBsSsMacAddress contains the MAC address of the MS
38              to which the power saving classes are associated."
39          INDEX { ifIndex,
40                wmanIf2mBsSsMacAddress,
41                wmanIf2mBsSsPsClassId }
42          ::= { wmanIf2mBsSsPowerSavingClassesTable 1 }
43
44
45      WmanIf2mBsSsPowerSavingClassesEntry ::= SEQUENCE {
46          wmanIf2mBsSsPsClassId          WmanIf2mPsClassId,
47          wmanIf2mBsSsStartFrameNumber  INTEGER,
48          wmanIf2mBsSsPowerSavingClassType WmanIf2mPsClassType,
49          wmanIf2mBsSsPsClassCidDirection WmanIf2mPsClassCidDir,
50          wmanIf2mBsSsTrafficTriggeredWakening INTEGER,
51          wmanIf2mBsSsInitialSleepWindow INTEGER,
52          wmanIf2mBsSsFinalSleepWindowBase INTEGER,
53          wmanIf2mBsSsFinalSleepWindowExponent INTEGER,
54          wmanIf2mBsSsListeningWindow INTEGER,
55          wmanIf2mBsSsPowerSavingMode WmanIf2mPowerSavingMode,
56          wmanIf2mBsSsSlpId            INTEGER}
57
58      wmanIf2mBsSsPsClassId OBJECT-TYPE
59
60
61
62
63
64
65

```

```

1      SYNTAX      WmanIf2mPsClassId
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This object uniquely identifies the power saving classes
6          in a MS."
7      ::= { wmanIf2mBsSsPowerSavingClassesEntry 1 }
8
9
10     wmanIf2mBsSsStartFrameNumber OBJECT-TYPE
11     SYNTAX      INTEGER
12     MAX-ACCESS  read-write
13     STATUS      current
14     DESCRIPTION
15         "Start frame number for first sleep window."
16     REFERENCE
17         "Subclause 6.3.2.3.44 in IEEE Std 802.16e-2005"
18     ::= { wmanIf2mBsSsPowerSavingClassesEntry 2 }
19
20
21
22     wmanIf2mBsSsPowerSavingClassType OBJECT-TYPE
23     SYNTAX      WmanIf2mPsClassType
24     MAX-ACCESS  read-write
25     STATUS      current
26     DESCRIPTION
27         "Power saving classes type I - BE & NRT-VR,
28         Power saving classes type II - UGS & RT-VR,
29         Power saving classes type III - multicast, management CID"
30     REFERENCE
31         "Subclause 6.3.21.2-4, in IEEE Std 802.16e-2005"
32     ::= { wmanIf2mBsSsPowerSavingClassesEntry 3 }
33
34
35
36
37     wmanIf2mBsSsPsClassCidDirection OBJECT-TYPE
38     SYNTAX      WmanIf2mPsClassCidDir
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "The direction of power saving class's CIDs."
43     REFERENCE
44         "Subclause 6.3.2.3.44, in IEEE Std 802.16e-2005"
45     ::= { wmanIf2mBsSsPowerSavingClassesEntry 4 }
46
47
48
49     wmanIf2mBsSsTrafficTriggeredWakening OBJECT-TYPE
50     SYNTAX      INTEGER (0..1)
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "0 = Power Saving Class shall not be deactivated if
55         traffic appears at the connection as per 6.3.19.2.
56         1 = Power Saving Class shall be deactivated if
57         traffic appears at the connection as 6.3.19.2."
58     REFERENCE
59         "Subclause 6.3.19.2, in IEEE Std 802.16e-2005"
60     ::= { wmanIf2mBsSsPowerSavingClassesEntry 5 }
61
62
63
64     wmanIf2mBsSsInitialSleepWindow OBJECT-TYPE
65

```

```

1      SYNTAX      INTEGER (0..255)
2      UNITS       "frame"
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "The initial duration for the sleep window. It is not
7          relevant for Power Saving Class type III, and shall
8          return '0'."
9
10     REFERENCE
11         "Subclause 6.3.2.3.44, in IEEE Std 802.16e-2005"
12     ::= { wmanIf2mBsSsPowerSavingClassesEntry 6 }
13
14
15     wmanIf2mBsSsFinalSleepWindowBase OBJECT-TYPE
16     SYNTAX      INTEGER (0..1023)
17     UNITS       "frame"
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21         "The final value for the sleep interval. It is not
22         relevant for Power Saving Class type II, and shall
23         return '0'. For Power Saving Class type III, it is the
24         base for duration of single sleep window request."
25     REFERENCE
26         "Subclause 6.3.2.3.44, in IEEE Std 802.16e-2005"
27     ::= { wmanIf2mBsSsPowerSavingClassesEntry 7 }
28
29
30     wmanIf2mBsSsFinalSleepWindowExponent OBJECT-TYPE
31     SYNTAX      INTEGER (0..7)
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "The factor by which the final-sleep window base is
36         multiplied in order to calculate the final-sleep window.
37         The following formula is used:
38         final-sleep window = final-sleep window base x
39                             2^(final-sleep window exponent)
40         For Power Saving Class type III, it is the exponent for
41         the duration of single sleep window request."
42     REFERENCE
43         "Subclause 6.3.2.3.44, in IEEE Std 802.16e-2005"
44     ::= { wmanIf2mBsSsPowerSavingClassesEntry 8 }
45
46
47     wmanIf2mBsSsListeningWindow OBJECT-TYPE
48     SYNTAX      INTEGER (0..255)
49     UNITS       "frame"
50     MAX-ACCESS  read-write
51     STATUS      current
52     DESCRIPTION
53         "The Duration of MS listening window. It is not
54         relevant for Power Saving Class type III, and shall
55         return '0'."
56     REFERENCE
57         "Subclause 6.3.2.3.44, in IEEE Std 802.16e-2005"
58     ::= { wmanIf2mBsSsPowerSavingClassesEntry 9 }
59
60
61
62
63
64
65

```

```

1
2
3 wmanIf2mBsSsPowerSavingMode OBJECT-TYPE
4     SYNTAX      WmanIf2mPowerSavingMode
5     MAX-ACCESS  read-write
6     STATUS      current
7     DESCRIPTION
8         "Indicate whether the power saving class mode of such
9         CID is active or not.
10        wmanIf2mBsSsPowerSavingMode = Sleep_Approved && Operation."
11
12    REFERENCE
13        "Subclause 6.3.2.3.45, in IEEE Std 802.16e-2005"
14    ::= { wmanIf2mBsSsPowerSavingClassesEntry 10 }
15
16
17 wmanIf2mBsSsSlpId OBJECT-TYPE
18     SYNTAX      INTEGER (0..1023)
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "wmanIf2mBsSsSlpId is assigned by the BS whenever an MS is
23         instructed to enter sleep mode. This number shall be unique
24         among all MSs that are in sleep mode."
25
26    REFERENCE
27        "Subclause 6.3.2.3.45, in IEEE Std 802.16e-2005"
28    ::= { wmanIf2mBsSsPowerSavingClassesEntry 11 }
29
30
31 wmanIf2mBsPm OBJECT IDENTIFIER ::= { wmanIf2mBsObjects 2 }
32
33 --
34 -- Mobile Station Sleep Mode Statistics Table
35 --
36
37 wmanIf2mBsSsSleepModeStatisticsTable OBJECT-TYPE
38     SYNTAX      SEQUENCE OF WmanIf2mBsSsSleepModeStatisticsEntry
39     MAX-ACCESS  not-accessible
40     STATUS      current
41     DESCRIPTION
42         "This table contains the sleep mode statistic for MS. This
43         table shall be maintained as FIFO to store the sleep mode
44         statistics over a period of time that is subject to
45         implementation. This statistics information can be to
46         monitor, finetuning, or debugging the power saving
47         performance of each MS. When the statistics entry for an
48         MS reaches the limit, it wraps around to the beginning, and
49         overwrites the oldest entry with the new entry. When the BS
50         roams to a different BS, all entries associated with such
51         MS will be deleted."
52
53    REFERENCE
54        "6.3.21 in IEEE Std 802.16e-2005"
55    ::= { wmanIf2mBsPm 1 }
56
57
58
59
60 wmanIf2mBsSsSleepModeStatisticsEntry OBJECT-TYPE
61     SYNTAX      WmanIf2mBsSsSleepModeStatisticsEntry
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65

```

```

1      "Each entry in the table contains the event of an MS
2      entering the sleep mode. It is indexed by ifIndex,
3      wmanIf2mBsSsMacAddress, and wmanIf2mBsSsStatisticsIndex.
4      wmanIf2mBsSsStatisticsIndex is the index to sleep mode event
5      entry in the table, and should be increased monotonically,
6      and wraps around when it reaches the implementation
7      specific limit. A time stamp is provided in each entry to
8      indicate when the sleep mode event took place."
9
10     INDEX      { ifIndex,
11                 wmanIf2mBsSsMacAddress,
12                 wmanIf2mBsSsCid,
13                 wmanIf2mBsSsStatisticsIndex }
14
15     ::= { wmanIf2mBsSsSleepModeStatisticsTable 1 }
16
17
18     WmanIf2mBsSsSleepModeStatisticsEntry ::= SEQUENCE {
19         wmanIf2mBsSsStatisticsIndex      Unsigned32,
20         wmanIf2mBsSsSleepWindowStarted   Unsigned32,
21         wmanIf2mBsSsListeningWindowStarted Unsigned32,
22         wmanIf2mBsSsPendingMsdu         INTEGER,
23         wmanIf2mBsSsSleepWindowTimeStamp DateAndTime}
24
25
26     wmanIf2mBsSsStatisticsIndex OBJECT-TYPE
27         SYNTAX      Unsigned32 (1 .. 4294967295)
28         MAX-ACCESS  read-only
29         STATUS      current
30         DESCRIPTION
31             "wmanIf2mBsSsStatisticsIndex identifies the entry in the
32             table where the latest sleep mode event took place."
33
34     ::= { wmanIf2mBsSsSleepModeStatisticsEntry 1 }
35
36
37     wmanIf2mBsSsSleepWindowStarted OBJECT-TYPE
38         SYNTAX      Unsigned32 (1 .. 166777215)
39         UNITS       "frame"
40         MAX-ACCESS  read-only
41         STATUS      current
42         DESCRIPTION
43             "wmanIf2mBsSsSleepWindowStarted identifies when the sleep
44             mode is activated.
45             wmanIf2mBsSsSleepWindowStarted = current frame number +
46             Start_frame_number.
47             The frame number is provided in the DL-MAP, and is
48             incremented by 1 MOD 2^24 each frame."
49
50     ::= { wmanIf2mBsSsSleepModeStatisticsEntry 2 }
51
52
53
54     wmanIf2mBsSsListeningWindowStarted OBJECT-TYPE
55         SYNTAX      Unsigned32 (1 .. 166777215)
56         UNITS       "frame"
57         MAX-ACCESS  read-only
58         STATUS      current
59         DESCRIPTION
60             "wmanIf2mBsSsListeningWindowStarted identifies when the sleep
61             mode is deactivated.
62             wmanIf2mBsSsListeningWindowStarted =
63             wmanIf2mBsSsListeningWindowStarted + sleep window
64
65

```

```

1           The frame number is provided in the DL-MAP, and is
2           incremented by 1 MOD 2^24 each frame."
3           ::= { wmanIf2mBsSsSleepModeStatisticsEntry 3 }
4
5
6 wmanIf2mBsSsPendingMsdu OBJECT-TYPE
7     SYNTAX      INTEGER
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11      "Indicate the number of MAC SDU that are received from the
12       network during the sleep window."
13    ::= { wmanIf2mBsSsSleepModeStatisticsEntry 4 }
14
15
16 wmanIf2mBsSsSleepWindowTimeStamp OBJECT-TYPE
17     SYNTAX      DateAndTime
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21      "This is the time when sleep window is started in seconds.
22       The definition of time is as in IETF RFC 868."
23    ::= { wmanIf2mBsSsSleepModeStatisticsEntry 5 }
24
25
26
27 wmanIf2mBsFm OBJECT IDENTIFIER ::= { wmanIf2mBsObjects 3 }
28
29
30 wmanIf2mBsSm OBJECT IDENTIFIER ::= { wmanIf2mBsObjects 4 }
31
32
33 wmanIf2mBsAm OBJECT IDENTIFIER ::= { wmanIf2mBsObjects 5 }
34
35 --
36 -- wmanIf2mSsObjects - containing tables and objects to be implemented in
37 -- the Mobile station
38 --
39 -- wmanIf2mSsCm contain the Mobile Station Configuration Management
40 -- objects
41 --
42 --
43 wmanIf2mSsCm OBJECT IDENTIFIER ::= { wmanIf2mSsObjects 1 }
44
45
46 wmanIf2mSsPm OBJECT IDENTIFIER ::= { wmanIf2mSsObjects 2 }
47
48
49 wmanIf2mSsFm OBJECT IDENTIFIER ::= { wmanIf2mSsObjects 3 }
50
51
52 wmanIf2mSsSm OBJECT IDENTIFIER ::= { wmanIf2mSsObjects 4 }
53
54
55 wmanIf2mSsAm OBJECT IDENTIFIER ::= { wmanIf2mSsObjects 5 }
56
57 --
58 -- wmanIf2mCommonObjects - containing tables and objects to be
59 -- implemented in the Mobile station
60 --
61 -- wmanIf2mCmnCm contain the Mobile Station Configuration Management
62 -- objects
63 --
64 wmanIf2mCmnCm OBJECT IDENTIFIER ::= { wmanIf2mCommonObjects 1 }
65

```

```
1  wmanIf2mCmnPm OBJECT IDENTIFIER ::= { wmanIf2mCommonObjects 2 }
2
3  wmanIf2mCmnFm OBJECT IDENTIFIER ::= { wmanIf2mCommonObjects 3 }
4
5  wmanIf2mCmnSm OBJECT IDENTIFIER ::= { wmanIf2mCommonObjects 4 }
6
7  wmanIf2mCmnAm OBJECT IDENTIFIER ::= { wmanIf2mCommonObjects 5 }
8
9
10
11  END
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

**Annex E.**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## **Annex F. Proposal for Adding Mobility Handover and Paging group MIBs**

### **1. Introduction**

With the mobility feature introduced, handover between BS and its neighbouring BS is inevitable. This contribution proposes to add BS handover related parameters which will help to execute smoothly handover.

Paging group configuration is also very important in the mobility scenario, proper paging group settings will make the paging procedure simple and effective. The configuration of paging group is also included in this contribution.

### **2. Proposed Text Introduction**

#### **2.1 wmanIfBsObjects**

##### **2.1.1 wmanIfBsMobility**

###### **2.1.1.1 wmanIfBsHandoverConfiguration**

wmanIfBsHandoverConfiguration contains handover related parameters. Handover related parameters include BS configuration parameters and its neighbouring BSes configuration parameters.

###### **2.1.1.2 wmanIfBsPagingGroupTable**

wmanIfBsPagingGroupTable contains paging group related parameters

### 3. ASN.1 Definitions of 802.16 MIB for SNMP

```

1
2
3
4
5     wman2IfMibObjects OBJECT IDENTIFIER ::= { wman2IfMib 1 }
6     wman2IfBsObjects OBJECT IDENTIFIER ::= { wman2IfMibObjects 1 }
7     wman2IfMsObjects OBJECT IDENTIFIER ::= { wman2IfMibObjects 2 }
8     wman2IfCommonObjects OBJECT IDENTIFIER ::= { wman2IfMibObjects 3 }
9
10    wmanIfBsFm OBJECT IDENTIFIER ::= { wman2IfBsObjects x } -- e.g. x=0
11    wmanIfBsCm OBJECT IDENTIFIER ::= { wman2IfBsObjects x } -- e.g. x=7
12    wmanIfBsAm OBJECT IDENTIFIER ::= { wman2IfBsObjects x } -- e.g. x=0
13    wmanIfBsPm OBJECT IDENTIFIER ::= { wman2IfBsObjects x } -- e.g. x=0
14    wmanIfBsSm OBJECT IDENTIFIER ::= { wman2IfBsObjects x } -- e.g. x=0
15    wmanIfMsFm OBJECT IDENTIFIER ::= { wman2IfMsObjects x } -- e.g. x=0
16    wmanIfMsCm OBJECT IDENTIFIER ::= { wman2IfMsObjects x } -- e.g. x=0
17    wmanIfMsAm OBJECT IDENTIFIER ::= { wman2IfMsObjects x } -- e.g. x=0
18    wmanIfMsPm OBJECT IDENTIFIER ::= { wman2IfMsObjects x } -- e.g. x=0
19    wmanIfMsSm OBJECT IDENTIFIER ::= { wman2IfMsObjects x } -- e.g. x=0
20    wmanIfBsCmHOConfiguration OBJECT IDENTIFIER ::= { wmanIfBsCm 2 }
21
22
23
24    wmanIfBsOperatorId OBJECT-TYPE
25        SYNTAX Integer32
26        MAX-ACCESS read-write
27        STATUS current
28        DESCRIPTION
29            "An unique operator identifier."
30        ::= { wmanIfBsHandoverConfiguration 1 }
31
32
33
34    wmanIfBsId OBJECT-TYPE
35        SYNTAX WmanIfBsIdType
36        MAX-ACCESS read-write
37        STATUS current
38        DESCRIPTION
39            "An unique BS identifier."
40        ::= { wmanIfBsHandoverConfiguration 2 }
41
42
43
44    wmanIfBsHandoverSupport OBJECT-TYPE
45        SYNTAX BITS
46            {
47                MDHO/FBSS HO not supported(0),
48                FBSS/MDHO DLRf combining supported(1),
49                MDHO DL soft combining supported monitoring single MAP from anchor
50    BS(2),
51                MDHO DL soft combining supported monitoring MAPS from active BSs(3),
52                reserved1(5),
53                reserved2(6),
54                reserved3(7)
55            }
56
57        MAX-ACCESS read-write
58        STATUS current
59        DESCRIPTION
60            "The Handover supported field indicates what type(s) of HO the BS and the MS
61    supports."
62        ::= { wmanIfBsHandoverConfiguration 3 }
63
64
65

```

1 wmanIfBsHandoverSupport OBJECT-TYPE  
2 SYNTAX BITS  
3 {  
4 mdho/fbss HO not supported(0),  
5 fbss/mdho DLRf combining supported(1),  
6 mdho DL soft combining supported monitoring single MAP from anchor BS(2),  
7 mdho DL soft combining supported monitoring MAPS from active BSs(3)  
8 }  
9  
10 MAX-ACCESS read-write  
11 STATUS current  
12 DESCRIPTION  
13 "The Handover supported field indicates what type(s) of HO the BS and the MS  
14 supports."  
15 ::= { wmanIfBsHandoverConfiguration 3 }  
16  
17  
18  
19 wmanIfBsResourceRetainTime OBJECT-TYPE  
20 SYNTAX Integer32  
21 MAX-ACCESS read-write  
22 STATUS current  
23 DESCRIPTION  
24 "The Resource\_Retain\_Time is the duration for MS s connection information  
25 that will be retained in serving BS. BS shall start Resource\_Retain\_Time timer at MS notification  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1 of pending HO attempt through MOB\_HO-IND or by detecting an MS drop. The unit of this value  
 2 is 100 milliseconds."  
 3 ::= { wmanIfBsHandoverConfiguration 4 }  
 4  
 5  
 6 wmanIfBsHOProcessOptimizationMSTimer OBJECT-TYPE  
 7 SYNTAX INTEGER  
 8 MAX-ACCESS read-write  
 9 STATUS current  
 10 DESCRIPTION  
 11 "the duration in frames MS shall wait until receipt  
 12 of the next unsolicited network re-entry MAC  
 13 management message as indicated in the HO Process  
 14 Optimization element of the RNG-RSP message."  
 15 ::= { wmanIfBsHandoverConfiguration 5 }  
 16  
 17  
 18  
 19 wmanIfBsMsHOREtransmissionTimer OBJECT-TYPE  
 20 SYNTAX INTEGER  
 21 MAX-ACCESS read-write  
 22 STATUS current  
 23 DESCRIPTION  
 24 "After a MS transmits MOB\_MSHO-REQ to initiate a handover process, it shall  
 25 start MS Handover Retransmission Timer and shall not transmit another MOB\_MSHO-REQ until  
 26 the expiration of the MS Handover Retransmission Timer."  
 27 ::= { wmanIfBsHandoverConfiguration 6 }  
 28  
 29  
 30  
 31 wmanIfBsMobilityModeSupport OBJECT-TYPE  
 32 SYNTAX BITS  
 33 {  
 34 handover support(0),  
 35 sleep-mode support(1),  
 36 idle-mode support(2)  
 37 }  
 38 MAX-ACCESS read-write  
 39 STATUS current  
 40 DESCRIPTION  
 41 "This parameter is to represent the supported mobility mode."  
 42 ::= { wmanIfBsHandoverConfiguration 7 }  
 43  
 44  
 45  
 46  
 47 wmanIfBsMsHOCconnectProcessingTime OBJECT-TYPE  
 48 SYNTAX INTEGER  
 49 MAX-ACCESS read-write  
 50 STATUS current  
 51 DESCRIPTION  
 52 "Time in ms the MS needs to process information  
 53 on connections provided in  
 54 RNGRSP or REG-RSP message during  
 55 HO."  
 56 ::= { wmanIfBsHandoverConfiguration 8 }  
 57  
 58  
 59  
 60 wmanIfBsMsHoTekProcessingTime OBJECT-TYPE  
 61 SYNTAX INTEGER  
 62 MAX-ACCESS read-write  
 63 STATUS current  
 64 DESCRIPTION  
 65

```

1           "Time in ms the MS needs to completely
2           process TEK information during HO."
3           ::= { wmanIfBsHandoverConfiguration 9 }
4
5
6   wmanIfBsULPermutationBase OBJECT-TYPE
7       SYNTAX OCTET STRING
8       MAX-ACCESS read-write
9       STATUS current
10      DESCRIPTION
11          "This parameter is used for uplink subcarrier allocation."
12      ::= { wmanIfBsHandoverConfiguration 10 }
13
14
15  wmanIfBsDLPermutationBase OBJECT-TYPE
16      SYNTAX OCTET STRING
17      MAX-ACCESS read-write
18      STATUS current
19      DESCRIPTION
20          "This parameter is used for downlink subcarrier allocation."
21      ::= { wmanIfBsHandoverConfiguration 11 }
22
23
24
25  wmanIfBsPreambleIndex OBJECT-TYPE
26      SYNTAX OCTET STRING
27      MAX-ACCESS read-write
28      STATUS current
29      DESCRIPTION
30          "This parameter is used for downlink synchronization by MS."
31      ::= { wmanIfBsHandoverConfiguration 12 }
32
33
34
35  wmanIfBsSegmentNumber OBJECT-TYPE
36      SYNTAX INTEGER
37      MAX-ACCESS read-write
38      STATUS current
39      DESCRIPTION
40          "This parameter is an unique segment identifier ."
41      ::= { wmanIfBsHandoverConfiguration 13 }
42
43
44
45  wmanIfNeighbourBsTable OBJECT-TYPE
46      SYNTAX SEQUENCE OF WmanIfNeighbourBsEntry
47      MAX-ACCESS not-accessible
48      STATUS current
49      DESCRIPTION
50          "This table contains neighbouring BS related parameters."
51      ::= { wmanIfBsHandoverConfiguration 14 }
52
53
54  wmanIfNeighbourBsEntry OBJECT-TYPE
55      SYNTAX WmanIfNeighbourBsEntry
56      MAX-ACCESS not-accessible
57      STATUS current
58      DESCRIPTION
59          "This table is indexed by wmanIfNeighbourBsId."
60      INDEX { ifIndex, wmanIfNeighbourBsId }
61      ::= { wmanIfNeighbourBsTable 1 }
62
63
64  wmanIfNeighbourBsEntry ::= SEQUENCE {
65

```

```

1      wmanIfNeighbourBsId          WmanIfBsIdType,
2      wmanIfNeighbourBsFAIndex INTEGER,
3      wmanIfNeighbourBsEIRP        INTEGER (-128..127),
4      wmanIfNeighbourBsHOPProcessOptimizationInteger32,
5      wmanIfNeighbourBsSchedulingServiceSupportBITS,
6      wmanIfNeighbourBsBandwidthInteger32,
7      wmanIfNeighbourBsFFTSize     Integer32,
8      wmanIfNeighbourBsCyclePrefixInteger32,
9      wmanIfNeighbourBsFrameDurationCodeInteger32,
10     wmanIfNeighbourBsULPermutationBaseInteger32,
11     wmanIfNeighbourBsDLPermutationBaseInteger32,
12     wmanIfNeighbourBsSegmentNumberInteger32,
13     wmanIfNeighbourBsPreambleIndexInteger32
14     }
15
16
17
18
19     wmanIfNeighbourBsId OBJECT-TYPE
20         SYNTAX WmanIfBsIdType
21         MAX-ACCESS read-write
22         STATUS current
23         DESCRIPTION
24             "The neighbouring BS identifier."
25             ::= { wmanIfNeighbourBsEntry 1 }
26
27
28
29     wmanIfNeighbourBsFAIndex OBJECT-TYPE
30         SYNTAX INTEGER
31         MAX-ACCESS read-write
32         STATUS current
33         DESCRIPTION
34             "Frequency Assignment Index."
35             ::= { wmanIfNeighbourBsEntry 2 }
36
37
38
39     wmanIfNeighbourBsEIRP OBJECT-TYPE
40         SYNTAX INTEGER (-128..127)
41         MAX-ACCESS read-write
42         STATUS current
43         DESCRIPTION
44             "Neighbour BS EIRP."
45             ::= { wmanIfNeighbourBsEntry 3 }
46
47
48
49     wmanIfNeighbourBsHOPProcessOptimization OBJECT-TYPE
50         SYNTAX Integer32
51         MAX-ACCESS read-write
52         STATUS current
53         DESCRIPTION
54             "Identifies re-entry process management messages that may be omitted during
55             the current HO attempt due to the availability of MS service and operational context information,
56             and the MS service and operational status post-HO completion."
57             ::= { wmanIfNeighbourBsEntry 4 }
58
59
60
61     wmanIfNeighbourBsSchedulingServiceSupport OBJECT-TYPE
62         SYNTAX BITS
63         {
64             real-time polling service(0),
65             extended real-time polling service(1),

```



```

1           non-real-time polling service(2),
2           unsolicited grant service(3),
3           best effort(4)
4       }
5
6       MAX-ACCESS read-write
7       STATUS current
8       DESCRIPTION
9           "This parameter is used to indicate neighbouring BS scheduling service type."
10      ::= { wmanIfNeighbourBsEntry 5 }
11
12
13      wmanIfNeighbourBsBandwidth OBJECT-TYPE
14          SYNTAX Integer32
15          MAX-ACCESS read-write
16          STATUS current
17          DESCRIPTION
18              "This parameter is used to indicate neighbouring BS bandwidth."
19          ::= { wmanIfNeighbourBsEntry 6 }
20
21
22
23      wmanIfNeighbourBsFFTSIZE OBJECT-TYPE
24          SYNTAX Integer32
25          MAX-ACCESS read-write
26          STATUS current
27          DESCRIPTION
28              "This parameter is used to indicate neighbouring BS FFT size."
29          ::= { wmanIfNeighbourBsEntry 7 }
30
31
32
33      wmanIfNeighbourBsCyclePrefix OBJECT-TYPE
34          SYNTAX Integer32
35          MAX-ACCESS read-write
36          STATUS current
37          DESCRIPTION
38              "This parameter is used to indicate neighbouring BS Cycle prefix."
39          ::= { wmanIfNeighbourBsEntry 8 }
40
41
42
43      wmanIfNeighbourBsFrameDurationCode OBJECT-TYPE
44          SYNTAX Integer32
45          MAX-ACCESS read-write
46          STATUS current
47          DESCRIPTION
48              "This parameter is used to indicate neighbouring BS Frame duration code."
49          ::= { wmanIfNeighbourBsEntry 9 }
50
51
52
53      wmanIfNeighbourBsULPermutationBase OBJECT-TYPE
54          SYNTAX Integer32
55          MAX-ACCESS read-write
56          STATUS current
57          DESCRIPTION
58              "This parameter is used to indicate neighbouring BS uplink permutation base."
59          ::= { wmanIfNeighbourBsEntry 10 }
60
61
62
63      wmanIfNeighbourBsDLPermutationBase OBJECT-TYPE
64          SYNTAX Integer32
65          MAX-ACCESS read-write
66          STATUS current

```

```

1         DESCRIPTION
2             "This parameter is used to indicate neighbouring BS downlink permutation
3 base."
4         ::= { wmanIfNeighbourBsEntry 11 }
5
6
7 wmanIfNeighbourBsSegmentNumber OBJECT-TYPE
8     SYNTAX Integer32
9     MAX-ACCESS read-write
10    STATUS current
11    DESCRIPTION
12        "This parameter is used to indicate neighbouring BS segment number."
13    ::= { wmanIfNeighbourBsEntry 12 }
14
15
16 wmanIfNeighbourBsPreambleIndex OBJECT-TYPE
17     SYNTAX Integer32
18     MAX-ACCESS read-write
19     STATUS current
20     DESCRIPTION
21        "This parameter is used to indicate neighbouring BS preamble index."
22    ::= { wmanIfNeighbourBsEntry 13 }
23
24
25
26 wmanIfBsPagingGroupTable OBJECT-TYPE
27     SYNTAX SEQUENCE OF WmanIfBsPagingGroupEntry
28     MAX-ACCESS not-accessible
29     STATUS current
30     DESCRIPTION
31        "This table contains paging group related parameters."
32    ::= { wmanIfBsMobility 3 }
33
34
35
36 wmanIfBsPagingGroupEntry OBJECT-TYPE
37     SYNTAX WmanIfBsPagingGroupEntry
38     MAX-ACCESS not-accessible
39     STATUS current
40     DESCRIPTION
41        "This table is indexed by wmanIfBsPagingGroupId."
42     INDEX { wmanIfBsPagingGroupId }
43     ::= { wmanIfBsPagingGroupTable 1 }
44
45
46
47 wmanIfBsPagingGroupEntry ::= SEQUENCE {
48     wmanIfBsPagingControlId          IpAddress,
49     wmanIfBsPagingGroupId            INTEGER,
50     wmanIfBsMgmtResourceHoldingTimerInteger32,
51     wmanIfBsT46Timer                 Integer32,
52     wmanIfBsPagingRetryCount         INTEGER,
53     wmanIfBsREQDuration              INTEGER,
54     wmanIfBsMACHashSkipThresholdInteger32,
55     wmanIfBsCDMATransmissionOpportunityAssignmentINTEGER,
56     wmanIfBsPagingResponseWindow    INTEGER,
57     wmanIfBsIdleModeTimer            INTEGER,
58     wmanIfBsIdleModeSystemTimer     INTEGER,
59     wmanIfBsPagingIntervalLength    INTEGER,
60     wmanIfBsPagingCycle              INTEGER
61 }
62
63
64
65

```

```

1      wmanIfBsPagingControlId OBJECT-TYPE
2          SYNTAX IpAddress
3          MAX-ACCESS read-write
4          STATUS current
5          DESCRIPTION
6              "This parameter is used to indicate paging controller identifier connected by BS."
7          ::= { wmanIfBsPagingGroupEntry 1 }
8
9
10     wmanIfBsPagingGroupId OBJECT-TYPE
11         SYNTAX INTEGER
12         MAX-ACCESS read-write
13         STATUS current
14         DESCRIPTION
15             "This parameter is used to indicate the paging group identifier assigned to BS by
16 network."
17         ::= { wmanIfBsPagingGroupEntry 2 }
18
19
20
21     wmanIfBsMgmtResourceHoldingTimer OBJECT-TYPE
22         SYNTAX Integer32
23         MAX-ACCESS read-write
24         STATUS current
25         DESCRIPTION
26             "Time the BS maintain connection
27 information with the MS after the
28 BS send DREG-CMD to the MS"
29         ::= { wmanIfBsPagingGroupEntry 3 }
30
31
32
33     wmanIfBsT46Timer OBJECT-TYPE
34         SYNTAX Integer32
35         MAX-ACCESS read-write
36         STATUS current
37         DESCRIPTION
38             "Time the BS waits for DREGREQ
39 in case of unsolicited Idle
40 Mode initiation from BS."
41         ::= { wmanIfBsPagingGroupEntry 4 }
42
43
44
45     wmanIfBsPagingRetryCount OBJECT-TYPE
46         SYNTAX INTEGER
47         MAX-ACCESS read-write
48         STATUS current
49         DESCRIPTION
50             "Number of retries on paging
51 transmission. If the BS does not
52 receive RNG-REQ from the MS
53 until this value decreases to zero,
54 it determines that the MS is
55 unavailable."
56         ::= { wmanIfBsPagingGroupEntry 5 }
57
58
59
60
61     wmanIfBsREQDuration OBJECT-TYPE
62         SYNTAX INTEGER
63         MAX-ACCESS read-write
64         STATUS current
65

```

1 DESCRIPTION  
2 "Waiting value for the DREG-REQ message re-transmission  
3 (measured in frames)."  
4 ::= { wmanIfBsPagingGroupEntry 6 }  
5  
6  
7 wmanIfBsMACHashSkipThreshold OBJECT-TYPE  
8 SYNTAX Integer32  
9 MAX-ACCESS read-write  
10 STATUS current  
11 DESCRIPTION  
12 "Maximum number of successive MOB\_PAG-ADV messages  
13 that may be sent from a BS without individual notification for  
14 an MS for which BS is allowed to skip MS MAC Address Hash  
15 when the Action Code for the MS is 0b00,'No Action Required'.  
16  
17 ::= { wmanIfBsPagingGroupEntry 7 }  
18  
19  
20 wmanIfBsCDMATransmissionOpportunityAssignment OBJECT-TYPE  
21 SYNTAX INTEGER  
22 MAX-ACCESS read-write  
23 STATUS current  
24 DESCRIPTION  
25 "The CDMA code and transmission opportunity  
26 assignment field indicates the assigned code  
27 and transmission opportunity for a MS who is  
28 paged to use over dedicated CDMA ranging region."  
29  
30 ::= { wmanIfBsPagingGroupEntry 8 }  
31  
32  
33 wmanIfBsPagingResponseWindow OBJECT-TYPE  
34 SYNTAX INTEGER  
35 MAX-ACCESS read-write  
36 STATUS current  
37 DESCRIPTION  
38 "The Page-Response Window indicates the Page-Response window for a MS  
39 who is paged to transmit  
40 the assigned code for CDMA ranging channel."  
41  
42 ::= { wmanIfBsPagingGroupEntry 9 }  
43  
44  
45 wmanIfBsIdleModeTimer OBJECT-TYPE  
46 SYNTAX INTEGER (128..65536)  
47 MAX-ACCESS read-write  
48 STATUS current  
49 DESCRIPTION  
50 "MS timed interval to conduct  
51 Location Update. Set timer to MS  
52 Idle Mode Timeout capabilities  
53 setting. Timer recycles on successful  
54 Idle Mode Location Update."  
55  
56 ::= { wmanIfBsPagingGroupEntry 10 }  
57  
58  
59 wmanIfBsIdleModeSystemTimer OBJECT-TYPE  
60 SYNTAX INTEGER (128..65536)  
61 MAX-ACCESS read-write  
62 STATUS current  
63 DESCRIPTION  
64  
65

1                   "For BS acting as Paging Controller,  
2                   timed interval to receive notification  
3                   of MS Idle Mode Location Update. Set  
4                   timer to MS Idle Mode Timeout. Timer  
5                   recycles on successful Idle Mode  
6                   Location Update."  
7  
8                   ::= { wmanIfBsPagingGroupEntry 11 }  
9  
10           wmanIfBsPagingIntervalLength OBJECT-TYPE  
11           SYNTAX INTEGER (2..5)  
12           MAX-ACCESS read-write  
13           STATUS current  
14           DESCRIPTION  
15                 "time duration of Paging Interval  
16                 of the BS."  
17                 ::= { wmanIfBsPagingGroupEntry 12 }  
18  
19           wmanIfBsPagingCycle OBJECT-TYPE  
20           SYNTAX INTEGER  
21           MAX-ACCESS read-write  
22           STATUS current  
23           DESCRIPTION  
24                 "Cycle in which the paging message is transmitted  
25                 within the paging group."  
26                 ::= { wmanIfBsPagingGroupEntry 13 }  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65