

Project	IEEE 802.16 Broadband Wireless Access Working Group < http://ieee802.org/16 >	
Title	MAC and PHY MIB for WirelessMAN and WirelessHUMAN BS and SS	
Date Submitted	2004-05-15	
Source(s)	Joey Chou Intel Corporation	[mailto:joey.chou@intel.com]
	Russ Reynolds Proxim Corporation	[mailto:RReynolds@proxim.com]
	Vladimir Yanover Alvarion	[mailto: Vladimir.Yanover@alvarion.com]
	Shlomi Eini Airspan	[mailto:seini@Airspan.com]
	Radu Selea Bogdan Moldoveanu Redline Communications	[mailto:radu@redlinecommunications.com]
Re:		
Abstract	IEEE 802.16 working group defines WirelessMAN and WirelessHUMAN air interface specifications for the development of standard based Base Station (BS) and Subscriber Station (SS) to provide broadband wireless services to Metropolitan Area Networks (MANs). This contribution defines the 802.16 MIB for MAC and PHY layers to achieve management interoperability and provide the remote management capability that are urgently needed for massive WirelessMAN and WirelessHUMAN deployment by carriers.	
Purpose	Adoption	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	<p>The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/jpr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard."</p> <p>Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents</p>	

(granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <<http://ieee802.org/16/ipr/patents/notices>>.

Table of Content

1. Introduction	5
1.1 Scope.....	5
1.2 References.....	5
2. BWA Network Management Reference Model.....	6
3. Relationship with Interface MIB	7
3.1 MIB-2 Integration	7
3.2 Usage of MIB-II Tables	7
4. 802.16 MIB Structure	9
4.1 wmanIfBsObjects.....	10
4.1.1 wmanIfBsSystem.....	10
4.1.1.1 wmanIfBsRegisteredSsTable.....	10
4.1.2 wmanIfBsPacketCs.....	10
4.1.2.1 wmanIfBsProvisionedSfTable.....	10
4.1.2.2 wmanIfBsServiceClassTable	10
4.1.2.3 wmanIfBsClassifierRuleTable.....	11
4.1.3 wmanIfBsCps.....	11
4.1.3.1 wmanIfBsConfigurationTable	11
4.1.3.2 wmanIfBsChMeasurementTable.....	12
4.1.4 wmanIfBsPkm	12
4.1.4.1 wmanIfBsPkmBaselineTable	12
4.1.4.2 wmanIfBsPkmAuthTable	12
4.1.4.3 wmanIfBsPkmTekTable	12
4.1.5 wmanIfBsNotification.....	12
4.2 wmanIfSsObjects.....	12
4.2.1 wmanSsSystem	12
4.2.1.1 wmanIfSsConfigFileEncodingTable.....	12
4.2.2 wmanIfSsCps	12
4.2.2.1 wmanIfSsConfigurationTable.....	12
4.2.2.2 wmanIfSsStatisticsCountersTable.....	12
4.2.3 wmanIfSsPkm.....	13
4.2.3.1 wmanIfSsPkmAuthTable.....	13
4.2.3.2 wmanIfSsPkmTekTable.....	13
4.2.3.3 wmanIfSsPkmCertificatesTable	13
4.2.4 wmanIfSsTraps.....	13

4.3	wmanIfCommonObjects	13
4.3.1	wmanIfCmnPacketCs	13
4.3.1.1	wmanIfCmnClassifierRuleTable	13
4.3.2	wmanIfCmnCps	13
4.3.2.1	wmanIfCmnServiceFlowTable	13
4.3.2.2	wmanIfCmnBsSsConfigurationTable	14
4.3.2.3	wmanIfCmnSsChMeasurementTable	14
4.3.3	wmanIfCmnPrivacy	14
4.3.3.1	wmanIfCmnCryptoSuiteTable	14
4.3.4	wmanIfCmnOfdmPhy	14
4.3.4.1	wmanIfOfdmUplinkChannelTable	14
4.3.4.2	wmanIfOfdmDownlinkChannelTable	14
4.3.4.3	wmanIfOfdmUcdBurstProfileTable	14
4.3.4.4	wmanIfOfdmDcdBurstProfileTable	15
5.	<i>ASN.1 Definition of 802.16 MIB</i>	16

1. Introduction

IEEE 802.16 working group defines WirelessMAN and WirelessHUMAN air interface specifications for the development of standard based Base Station (BS) and Subscriber Station (SS) to provide broadband wireless services to Metropolitan Area Networks (MANs). This contribution defines the 802.16 MIB for MAC and PHY layers to achieve management interoperability and provide the remote management capability that are urgently needed for massive WirelessMAN and WirelessHUMAN deployment.

1.1 Scope

The scope of this contribution is to define the 802.16 MAC and PHY MIB for SS and BS, based on IEEE 802.16REVd/D3 specification [3]. The definition of managed objects in this MIB is based on SNMPv2 Structure of Management Information (SMI) [4] and Textual Conventions [5]. Therefore, 802.16 MIB is compliant to SNMPv2, but is backward compatible to SNMPv1 through appropriate translation. It is also the intent to support SNMPv3.

Since 802.16 MIB has to be accessed through MIB tree, its relationship with Interface MIB—RFC2863 [7] will be described. Additional MIBs may be necessary to manage other interfaces in SS or BS, such as Ethernet, T1/E1, and ATM, but they are outside the scope of this contribution.

1.2 References

- [1] IEEE 802.16-2001, "IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Wireless Access Systems".
- [2] IEEE 802.16a-2003, "IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Wireless Access Systems – Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz.
- [3] IEEE 802.16REVd/D3-2004, "Draft IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems".
- [4] RFC1902, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", January 1996
- [5] RFC1903, "Textual Convention for Version 2 of the Simple Network Management Protocol (SNMPv2)", January 1996
- [6] RFC 1213, " Management Information Base for Network Management of TCP/IP-based internets: MIB-II", IETF, March 1991
- [7] RFC2863, "The Interfaces Group MIB", June, 2000
- [8] RFC2515, "Definitions of Managed Objects for ATM Management", February, 1999
- [9] IEEE P802.16-REVd/D4-2004, "Draft IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems", March 29, 2004.

2. BWA Network Management Reference Model

Figure 1 shows the management reference model of Broadband Wireless Access (BWA) networks. It consists of a network Management System (NMS), performing the network manager role, and managed nodes, which provide access to managed objects via MIB or virtual information store. SSs and BSs are managed nodes that act in the SNMP agent role. Furthermore, managed SSs, which have a secondary management CID, may be managed indirectly through the BS to which they are registered. In this case, the BS acts in an SNMP Proxy role on behalf of managed SSs. SS can be managed by NMS directly as well.

The management information between SS and BS will be carried over Second Management CID for managed SS. If the 2nd management CID does not exist, the SNMP messages shall go through another interface in the customer premise. The SNMP agent in the SS can be managed directly, or via a SNMP proxy in the BS.

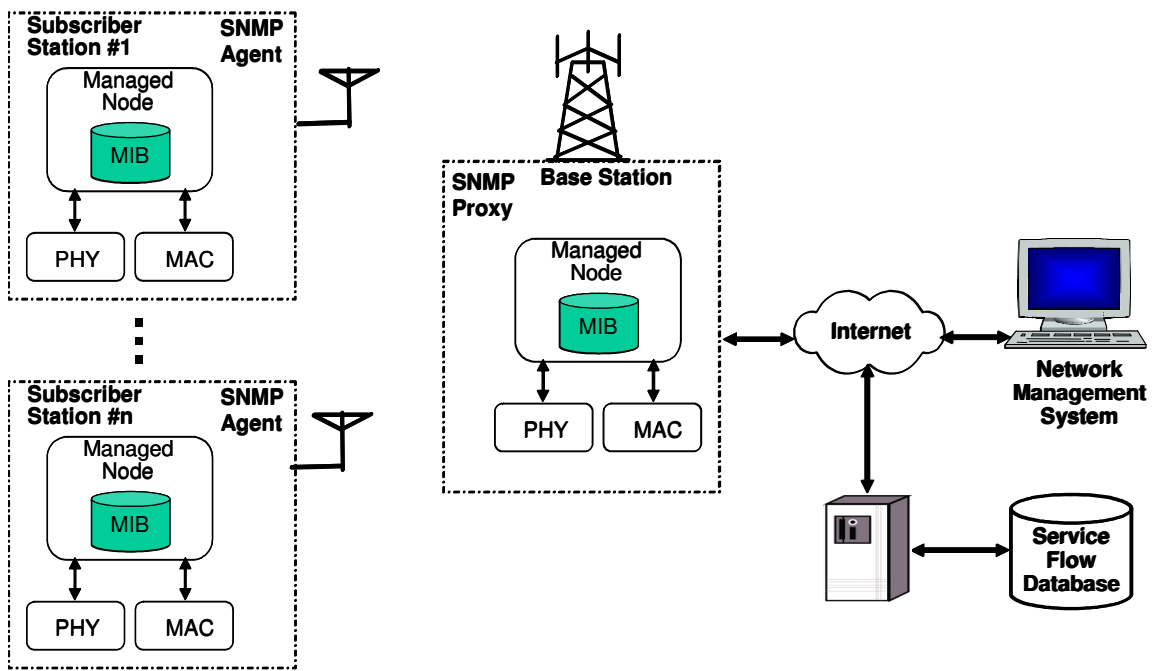


Figure 1 - BWA Network Management Reference Model

1 **3. Relationship with Interface MIB**

2 This section describes the integration with MIB-II [6] under Interface Group MIB defined in
 3 RFC2863, as 802.16 MIB will need to be integrated in the MIB tree. It describes where
 4 802.16 MIB is located in the MIB-II subtree, and how it can be accessed by NMS.

5 **3.1 MIB-2 Integration**

6 The IANA has assigned the following `ifType` to point to multipoint broadband wireless
 7 access.

```
8     IANAifType ::= TEXTUAL-COVENTION
9     SYNTAX INTEGER {
10         propBWAp2Mp (184) -- prop broadband wireless access
11         -- point to multipoint
12     }
```

13 Therefore, upon 802.16 MIB being approved by the IETF, this MIB can be accessed
 14 through

```
15     iso.org.dod.internet.mgmt.mib-2.transmission.ifType
16     (1.3.6.1.2.1.10.184)
```

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

```
18     wmanMIB ::= {transmission 184} -- WMAN interface table
```

19 Before the approval of the IETF; however, 802.16 MIB is temporary located under
 20 enterprise via

```
21     iso.org.dod.internet.private.enterprise.wmanMIB
22     (1.3.6.1.4.1.n)
```

23 Or

```
24     iso.org.dod.internet.private.enterprise.vendorID.wmanMIB
25     (1.3.6.1.4.1.xxx.n)
```

26 **3.2 Usage of MIB-II Tables**

27 “Interfaces” group of MIB-II, in RFC1573, has been designed to manage various sub-
 28 layers (e.g. MAC and PHY) beneath the internetwork-layer for numerous media-specific
 29 interfaces. `ifTable` in MIB-II is used to access the `wmanIfMib`.

30 Table 1 describes some key attributes in the `ifTable` that will be reused in the BS
 31 `wmanIfMib`. When the SNMP agent is implemented in a common base station controller,
 32 each BS sector will have an entry in the `ifTable`. When the SNMP agent is implemented in
 33 the sector controller, there is only one entry for the BS sector in the `ifTable`.

34

35

36

37

38

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

<i>ifTable</i>	<i>ifIndex</i>	<i>ifType (IANA)</i>	<i>ifSpeed</i>	<i>ifPhysAddress</i>	<i>ifAdminStatus</i>	<i>ifOperStatus</i>
BS Sector 1	An ifEntry per BS sector (1)	propBWA2Mp	Null	MAC address of BS sector	Administration Status	Operational Status
BS Sector 2	An ifEntry per BS sector (2)	propBWA2Mp	Null	MAC address of BS sector	Administration Status	Operational Status
BS Sector 3	An ifEntry per BS sector (3)	propBWA2Mp	Null	MAC address of BS sector	Administration Status	Operational Status
Ethernet			Null	MAC address	Administration Status	Operational Status

Table 1 – Usage of ifTable objects for Base Station

Table 2 show the usage of ifTable for SS. There is only one entry for the SS itself. Additional entries may be necessary to support other network interfaces, such as Ethernet.

<i>ifTable</i>	<i>ifIndex</i>	<i>ifType (IANA)</i>	<i>ifSpeed</i>	<i>ifPhysAddress</i>	<i>ifAdminStatus</i>	<i>ifOperStatus</i>
SS	An ifEntry for SS	propBWA2Mp	Null	MAC address of SS	Administration Status	Operational Status
Ethernet			Null	MAC address	Administration Status	Operational Status

Table 2– Usage of ifTable objects for Subscriber Station

4. 802.16 MIB Structure

Figure 2 shows the MIB structure of wmanIfMib for 802.16 [3]. The MIB structure is organized based on the the reference model as defined in IEEE 802.16REVd/D3 standard [3].

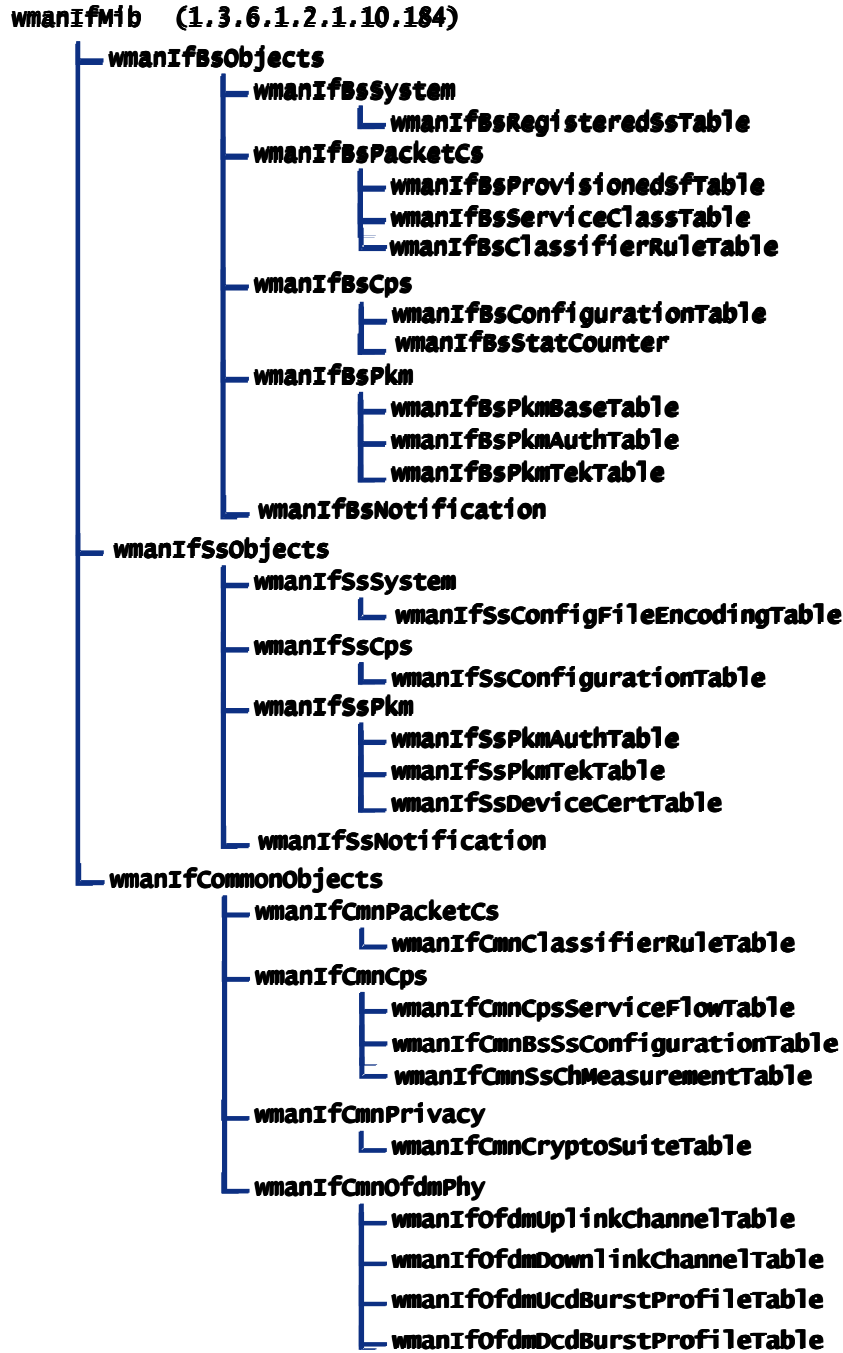


Figure 2 – wmanIfMib Structure

- 1 wmanIfMib is composed of three groups:
- 2 ▪ wmanIfBsObjects: This group contains managed objects to be implemented
 - 3 in the SNMP agent in BS.
 - 4 ▪ wmanIfSsObjects: This group contains managed objects to be implemented
 - 5 in the SNMP agent in SS.
 - 6 ▪ wmanIfCommonObjects: This group contains common managed objects to be
 - 7 implemented in the SNMP agent in BS and SS.

8 **4.1 wmanIfBsObjects**

9 4.1.1 **wmanIfBsSystem**

10 wmanIfBsSystem group contains system level BS managed objects.

11 4.1.1.1 wmanIfBsRegisteredSsTable

12 This table is indexed by BS ifIndex and wmanIfBsSsIdIndex that contains SS
13 information obtained from REG-REQ message as defined in section 6.3.2.3.7 in [9].
14 Each entry in the table may contain the following objects.

- 15 ▪ Basic CID
- 16 ▪ Primary management CID
- 17 ▪ Secondary Management CID
- 18 ▪ HMAC tuple
- 19 ▪ Uplink CID support
- 20 ▪ SS management support
- 21 ▪ SS capability
- 22 ▪ IP version
- 23 ▪ CS sublayer capabilities

24 4.1.2 **wmanIfBsPacketCs**

25 wmanIfBsPacketCs group contains BS managed objects relating to the Packet CS
26 management entity layer in figure 1 of [3].

27 4.1.2.1 wmanIfBsProvisionedSfTable

28 This table is doubly indexed by SS MAC address and Service Flow ID and contains
29 provisioned service flow profiles, Per SS. These connection parameters shall be
30 provisioned for the SS using DSA messages, as specified in [9], 6.3.2.3.10. Admittance
31 and activation of provisioned service flow may be postponed.

32 4.1.2.2 wmanIfBsServiceClassTable

33 This table is provisioned and is indexed by QoS profile index. Each entry of the table
34 contains corresponding service flow characteristic attributes (e.g. QoS parameter set)
35 as defined in section 6.13.4 in [9].

1 To facilitate the NMS task of provisioning service flow attributes for hundreds or even
 2 thousands of subscriber stations supported by each BS, the concept of Provisioned
 3 Service Classes are devised. Figure 3 shows an example of QoS profiles that are
 4 created to define the service flow attributes that can be shared by multiple service
 5 flows. For example, Basic CID UL for SSs A1, B1, and X1 uses profile 1. Service flow
 6 attribute profiles can be added or deleted dynamically to meet different QoS demands
 7 from subscribers.

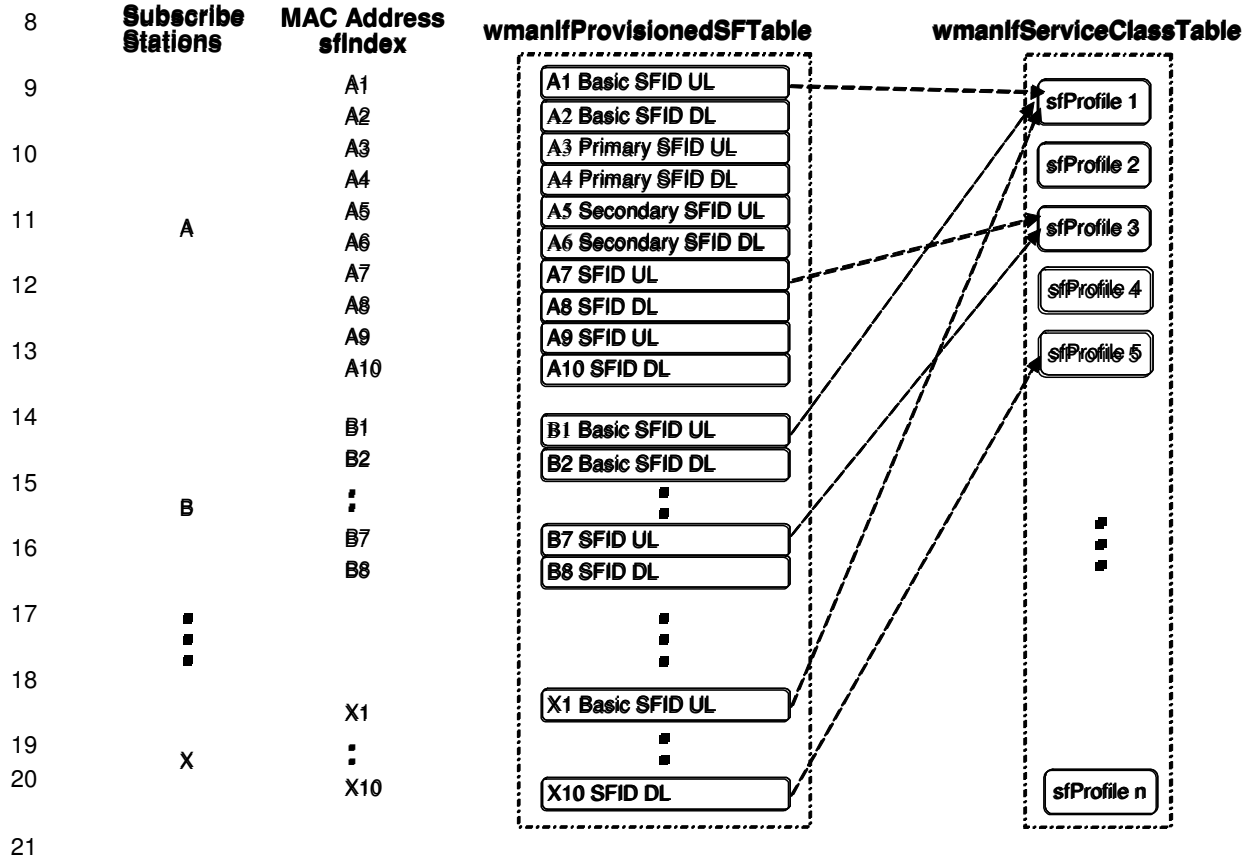


Figure 3 – Service Classes – Service Flows Mapping

4.1.2.3 wmanIfBsClassifierRuleTable

This table is indexed by service flow index and classifier rule index, and contains the packet classifier rules.

4.1.3 wmanIfBsCps

wmanIfBsCpsParameters group contains BS managed objects relating to the MAC CPS management entity layer in figure 1 of [3].

4.1.3.1 wmanIfBsConfigurationTable

This table contains objects for BS system parameters and constants as defined in section 10.1, Table 295 of [9]. It is indexed by BS Id.

- 1 4.1.3.2 **wmanIfBsChMeasurementTable**
2 This table is indexed by BS ifIndex and contains statistics about the channel
3 measurement.
- 4 4.1.4 **wmanIfBsPkm**
5 wmanIfBsPkm group contains BS managed objects relating to the MAC CPS privacy
6 management entity section in figure 1 of [3].
- 7 4.1.4.1 **wmanIfBsPkmBaselineTable**
8 This table is indexed by BS ifIndex and contains base station PKM operational
9 parameters described in section 10.2 and table 296 of [9].
- 10 4.1.4.2 **wmanIfBsPkmAuthTable**
11 This table is double indexed by ifIndex and SsMacAddress and contains runtime
12 subscriber station authentication and authorization parameters for each base station.
- 13 4.1.4.3 **wmanIfBsPkmTekTable**
14 This table is double indexed by ifIndex and SAId and contains runtime Security
15 association parameters for each base station.
- 16 4.1.5 **wmanIfBsNotification**
17 wmanIfBsNotification group contains BS traps to report fault events and exceptions,
18 such as link-up and link-down.
- 19 **4.2 wmanIfSsObjects**
- 20 4.2.1 **wmanSsSystem**
21 wmanIfSsSystem group contains subscriber station system level objects.
- 22 4.2.1.1 **wmanIfSsConfigFileEncodingTable**
23 This table is indexed by SS index, and contain configuration file information about the
24 subscriber station such as manufacturer, hardware model, serial number, and software
25 or firmware revision.
- 26 4.2.2 **wmanIfSsCps**
27 wmanIfSsCpsParameters group contains subscriber station manageable objects
28 relating to the MAC CPS management entity layer in figure 1 of [3].
- 29 4.2.2.1 **wmanIfSsConfigurationTable**
30 This table is indexed by SS Id and contains objects for SS system parameters and
31 constants as defined in section 10.1, Table 295 of [9].
- 32 4.2.2.2 **wmanIfSsStatisticsCountersTable**
33 This object contains the performance monitoring data for SS.

1 4.2.3 **wmanIfSsPkm**

2 wmanIfSsPkmParameters group contains subscriber station manageable objects
3 relating to the MAC CPS privacy management entity section in figure 1 of [3].

4 4.2.3.1 **wmanIfSsPkmAuthTable**

5 This table is indexed by SS MAC address and contains subscriber station
6 authentication and authorization parameters including those described in section 10.2
7 and table 296 of [9].

8 4.2.3.2 **wmanIfSsPkmTekTable**

9 This table is doubly indexed by SS MAC address and SAId and contains subscriber
10 station runtime parameters for each active security association.

11 4.2.3.3 **wmanIfSsPkmCertificatesTable**

12 This table is indexed by SS MAC address and contains subscriber station and SS
13 manufacturer certificates.

14 4.2.4 **wmanIfSsTraps**

15 wmanIfBsTraps group contains SS traps to report fault events and exceptions, such as
16 link-up and link-down.

17 **4.3 wmanIfCommonObjects**

18 4.3.1 **wmanIfCmnPacketCs**

19 4.3.1.1 **wmanIfCmnClassifierRuleTable**

20 wmanIfClassifierRuleTable is indexed by service flow ID and contains runtime classifier
21 rules screening criteria for each service flow as described in section 11.13.21 of [9].

22 4.3.2 **wmanIfCmnCps**

23 4.3.2.1 **wmanIfCmnServiceFlowTable**

24 This table is doubly indexed by ifIndex and service flow ID. In the BS, it represents the
25 totality of all provisioned, admitted, and active service flow for both DL and UL
26 directions. In the SS, this table should contain the service flows, both DL and UL, being
27 allocated to a specific SS.

28 A Service Flow is represented by parameters, such as

- 29
 - Service Flow common parameters, like SFID and CID
 - 30 - Classifiers associated with Service Flow, see [9] , 5.2.2, 5.2.5 – 5.2.7
 - 31 - Service Flow QoS parameters like QoS parameters of specific Service Flow,
32 like Max Sustained Traffic Rate, QoS status (admitted etc.)
 - 33 - Service Flow Header Suppression parameters like associated classifier and
34 PHS rule, see [9] , 5.2.4

1 4.3.2.2 **wmanIfCmnBsSsConfigurationTable**

2 This table is indexed by SS Id and contains objects for SS system parameters and
3 constants as defined in section 10.1, Table 295 of [9].

4 4.3.2.3 **wmanIfCmnSsChMeasurementTable**

5 This object contains the channel measurement table for SS.

6 4.3.3 **wmanIfCmnPrivacy**

7 4.3.3.1 **wmanIfCmnCryptoSuiteTable**

8 This table is doubly indexed by ifIndex and wmanIfCryptoSuiteIndex and contains
9 supported crypto suites for the particular SS and other crypto parameters such as key
10 lifetimes. See sections 11.9.14 and 11.9.15 of [9].

11 4.3.4 **wmanIfCmnOfdmPhy**

12 wmanIfOfdmPhy is a group containing objects specific to OFDM PHY.

13 4.3.4.1 **wmanIfOfdmUplinkChannelTable**

14 This table contains the uplink channels that the BS is able to receive. In the SS, this
15 table should have an entry indicating the uplink channel that the SS can transmit. Each
16 entry contains the parameters needed to describe uplink channel descriptor as defined
17 in section 11, Table 302 and 305 of [9], and include the following objects.

- 18 ▪ Uplink center frequency (KHz)
- 19 ▪ Subchannelization REQ Region-Full Parameters
- 20 ▪ Bandwidth request opportunity size
- 21 ▪ Ranging request opportunity size

22 4.3.4.2 **wmanIfOfdmDownlinkChannelTable**

23 This table contains the downlink channels that the BS is able to transmit. In the SS, this
24 table should have an entry indicating the downlink channel that the SS can receive.
25 Each entry contains the parameters needed to describe downlink channel descriptor as
26 defined in section 11, Table 312 of [9], and including the following.

- 27 ▪ channel number (for license exempt operation only)
- 28 ▪ Frequency (downlink center frequency (kHz))
- 29 ▪ BS EIRP
- 30 ▪ TTG
- 31 ▪ RTG
- 32 ▪ MAC Version

33 4.3.4.3 **wmanIfOfdmUcdBurstProfileTable**

34 Each entry in this table contains the parameters needed for the UCD burst profile as
35 defined in section 11, Table 310 of [9].

- 1 4.3.4.4 wmanIfOfdmDcdBurstProfileTable
- 2 wmanIfDcdBurstProfileTable – Each entry in this table contains the parameters
- 3 needed for the UCD burst profile as defined in section 11, Table 316 of [9].

1 **5. ASN.1 Definition of 802.16 MIB**

```

2  WMAN-IF-MIB DEFINITIONS ::= BEGIN
3
4  IMPORTS
5      MODULE-IDENTITY,
6      OBJECT-TYPE,
7      NOTIFICATION-TYPE,
8      Unsigned32,
9      Integer32,
10     Counter32,
11     Counter64,
12     TimeTicks,
13     IpAddress,
14     transmission
15         FROM SNMPv2-SMI
16     SnmpAdminString
17         FROM SNMP-FRAMEWORK-MIB
18     TEXTUAL-CONVENTION,
19     MacAddress,
20     RowStatus,
21     TruthValue,
22     DateAndTime,
23     DisplayString,
24     TimeInterval,
25     TimeStamp
26         FROM SNMPv2-TC
27     InetAddressType, InetAddress
28         FROM INET-ADDRESS-MIB
29     OBJECT-GROUP,
30
31     MODULE-COMPLIANCE
32         FROM SNMPv2-CONF
33     ifIndex, InterfaceIndexOrZero
34         FROM IF-MIB;
35
36 wmanIfMib MODULE-IDENTITY
37     LAST-UPDATED      "0405120000Z" -- May 12, 2004
38     ORGANIZATION      "IETF IPCDN Working Group"
39     CONTACT-INFO
40         "              Joey Chou
41         Postal: Intel Corporation
42         5000 W. Chandler Blvd, Chandler, AZ 85227, USA
43         E-mail: joey.chou@intel.com
44
45         Russ Reynolds
46         Postal: Proxim Corporation
47         935 Stewart Drive, Sunnyvale, CA 94085, USA
48         E-mail: RReynolds@proxim.com
49
50         Shlomi Eini
51         Postal: Airspan Networks

```



```

1             Airport city 70100,Israel
2             E-mail: seini@airspan.com
3
4             Bogdan Moldoveanu
5             Postal: Redline Communications Inc.
6                 302 Town Centre Blvd., Markham, ON L3R 0E8, Canada
7             E-mail: b moldoveanu@redlinecommunications.com"
8
9             DESCRIPTION
10            "This MIB Module defines managed objects for 802.16 based
11            Subscriber Station and Base Station."
12            ::= { transmission 184 }
13
14            -- Textual Conventions
15
16            wmanIfMibObjects OBJECT IDENTIFIER ::= { wmanIfMib 1 }
17            wmanIfBsObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 1 }
18            wmanIfSsObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 2 }
19            wmanIfCommonObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 3 }
20
21            --
22            -- BS object group - containing tables and objects to be implemented in
23            -- the Base station
24            --
25            -- wmanIfBsSystem contain the Base Station system objects
26            wmanIfBsSystem OBJECT IDENTIFIER ::= { wmanIfBsObjects 1 }
27
28            wmanIfBsRegisteredSsTable OBJECT-TYPE
29                SYNTAX SEQUENCE OF wmanIfBsRegisteredSsEntry
30                MAX-ACCESS not-accessible
31                STATUS current
32                DESCRIPTION
33                    "This table contains entries of Ss that have been
34                    registered through REG-REQ message"
35                REFERENCE
36                    "Section 6.4.3.2.7 in IEEE 802.16REVd/D3-2004"
37                ::= { wmanIfBsSystem 1 }
38
39            wmanIfBsRegisteredSsEntry OBJECT-TYPE
40                SYNTAX wmanIfBsRegisteredSsEntry
41                MAX-ACCESS not-accessible
42                STATUS current
43                DESCRIPTION
44                    "This table provides one row for each SS that has been
45                    registered in the BS, and is indexed by
46                    wmanIfBsSsIdIndex. The primary index is the ifIndex
47                    with an ifType of propBwAp2Mp. The ifIndex identifies
48                    which BS sector with which the SS is associated."
49                INDEX { ifIndex, wmanIfBsSsIdIndex }
50                ::= { wmanIfBsRegisteredSsTable 1 }
51
52            wmanIfBsRegisteredSsEntry ::= SEQUENCE {
53                wmanIfBsSsIdIndex Unsigned32,
54                wmanIfBsSsMacAddress MacAddress,

```

```

1      wmanIfBsSsBasicCid          INTEGER,
2      wmanIfBsSsPrimaryCid       INTEGER,
3      wmanIfBsSsSecondaryCid     INTEGER,
4      wmanIfBsHmacTuple          OCTET STRING,
5      wmanIfBsUlCidSupport        INTEGER,
6      wmanIfBsSsManagementSupport INTEGER,
7      wmanIfBsSsArqSupport        INTEGER,
8      wmanIfBsSsDsxFowControl     INTEGER,
9      wmanIfBsSsMacCrcSupport     INTEGER,
10     wmanIfBsSsMcaFlowControl     INTEGER,
11     wmanIfBsSsMcpGroupCidSupport INTEGER,
12     wmanIfBsSsPkmFlowControl     INTEGER,
13     wmanIfBsIpVersion            INTEGER,
14     wmanIfBsSsMacCsSupportBitMap BITS,
15     wmanIfBsSsMaxNumOfClassifier INTEGER,
16     wmanIfBsSsPhsSupport         INTEGER
17     }
18
19     wmanIfBsSsIdIndex OBJECT-TYPE
20         SYNTAX      Unsigned32 (1 .. 4294967295)
21         MAX-ACCESS  read-only
22         STATUS      current
23         DESCRIPTION
24             "wmanIfBsSsIdIndex identifies the SS that is registered."
25         REFERENCE
26             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
27         ::= { wmanIfBsRegisteredSsEntry 1 }
28
29     wmanIfBsSsMacAddress OBJECT-TYPE
30         SYNTAX      MacAddress
31         MAX-ACCESS  read-only
32         STATUS      current
33         DESCRIPTION
34             "The MAC address of SS is received from the RNG-REQ
35             message. This MAC address can be used as the
36             index to find out the BS and its associated SSS."
37         REFERENCE
38             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
39         ::= { wmanIfBsRegisteredSsEntry 2 }
40
41     wmanIfBsSsBasicCid OBJECT-TYPE
42         SYNTAX      INTEGER
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "The value of this object indicates the SS's basic CID
47             that was sent in the RNG-RSP message."
48         REFERENCE
49             "Section 6.4.9.5 in IEEE 802.16REvD/D3-2004"
50         ::= { wmanIfBsRegisteredSsEntry 3 }
51
52     wmanIfBsSsPrimaryCid OBJECT-TYPE
53         SYNTAX      INTEGER
54         MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object indicates the basic CID of the
4          SS received from the RNG-RSP message."
5      REFERENCE
6          "Section 6.4.9.5 in IEEE 802.16REVd/D3-2004"
7      ::= { wmanIfBsRegisteredSsEntry 4 }
8
9      wmanIfBsSsSecondaryCid OBJECT-TYPE
10     SYNTAX      INTEGER
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "The value of this object indicates the secondary
15         management CID present in the REG-REQ message. The value
16         should be null indicating the 2nd management CID doesn't
17         exist."
18     REFERENCE
19         "Section 6.4.2.3.8 in IEEE 802.16REVd/D3-2004"
20     ::= { wmanIfBsRegisteredSsEntry 5 }
21
22     wmanIfBsHmacTuple OBJECT-TYPE
23     SYNTAX      OCTET STRING
24     MAX-ACCESS  read-only
25     STATUS      current
26     DESCRIPTION
27         "This parameter contains the HMAC Key Sequence Number
28         concatenated with an HMAC-Digest used for message
29         authentication. The HMAC Key Sequence Number is stored
30         in the four least significant bits of the first byte of
31         the HMAC Tuple, and the most significant four bits
32         are reserved."
33     REFERENCE
34         "Section 11.1.2 in IEEE 802.16REVd/D3-2004"
35     ::= { wmanIfBsRegisteredSsEntry 6 }
36
37     wmanIfBsUlCidSupport OBJECT-TYPE
38     SYNTAX      INTEGER
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "This object shows the number of Uplink CIDs the SS can
43         support."
44     REFERENCE
45         "Section 11.7.4 in IEEE 802.16REVd/D3-2004"
46     ::= { wmanIfBsRegisteredSsEntry 7 }
47
48     wmanIfBsSsManagementSupport OBJECT-TYPE
49     SYNTAX      INTEGER {unmanagedSs(0),
50                 managedSs(1)}
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "This object indicates whether or not the SS is managed."

```

```

1      REFERENCE
2      "Section 11.7.1.1 in IEEE 802.16REvd/D3-2004"
3      ::= { wmanIfBsRegisteredSsEntry 8 }
4
5      wmanIfBsSsArqSupport OBJECT-TYPE
6      SYNTAX      INTEGER {arqOn(0),
7                  arqOff(1)}
8      MAX-ACCESS  read-only
9      STATUS      current
10     DESCRIPTION
11     "This object indicates whether the SS support ARQ."
12     REFERENCE
13     "Section 11.7.6.1 in IEEE 802.16REvd/D3-2004"
14     ::= { wmanIfBsRegisteredSsEntry 9 }
15
16     wmanIfBsSsDsxFwControl OBJECT-TYPE
17     SYNTAX      INTEGER (0..255)
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21     "This object specifies the maximum number of concurrent
22     DSA, DSC, or DSD transactions that may be outstanding."
23     REFERENCE
24     "Section 11.7.6.2 in IEEE 802.16REvd/D3-2004"
25     ::= { wmanIfBsRegisteredSsEntry 10 }
26
27     wmanIfBsSsMacCrcSupport OBJECT-TYPE
28     SYNTAX      INTEGER {noMacCrcSupport(0),
29                  macCrcSupport(1)}
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33     "This object indicates whether or not the SS supports MAC
34     level CRC."
35     REFERENCE
36     "Section 11.7.6.3 in IEEE 802.16REvd/D3-2004"
37     ::= { wmanIfBsRegisteredSsEntry 11 }
38
39     wmanIfBsSsMcaFlowControl OBJECT-TYPE
40     SYNTAX      INTEGER (0..255)
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44     "This object specifies the maximum number of concurrent
45     MCA transactions that may be outstanding."
46     REFERENCE
47     "Section 11.7.6.4 in IEEE 802.16REvd/D3-2004"
48     ::= { wmanIfBsRegisteredSsEntry 12 }
49
50     wmanIfBsSsMcpGroupCidSupport OBJECT-TYPE
51     SYNTAX      INTEGER (0..255)
52     MAX-ACCESS  read-only
53     STATUS      current
54     DESCRIPTION

```

```

1           "This object indicates the maximum number of
2           simultaneous Multicast Polling Groups the SS is
3           capable of belonging to."
4     REFERENCE
5           "Section 11.7.6.5 in IEEE 802.16REvD/D3-2004"
6     ::= { wmanIfBsRegisteredSsEntry 13 }
7
8     wmanIfBsSsPkmFlowControl OBJECT-TYPE
9       SYNTAX      INTEGER (0..255)
10      MAX-ACCESS  read-only
11      STATUS      current
12      DESCRIPTION
13        "This object specifies the maximum number of concurrent PKM
14        transactions that may be outstanding."
15      REFERENCE
16        "Section 11.7.6.6 in IEEE 802.16REvD/D3-2004"
17      ::= { wmanIfBsRegisteredSsEntry 14 }
18
19     wmanIfBsIpVersion OBJECT-TYPE
20       SYNTAX      INTEGER {ipv4(1),
21                          ipv6(2)}
22       MAX-ACCESS  read-only
23       STATUS      current
24       DESCRIPTION
25        "This object indicates the version of IP used on the
26        Secondary Management Connection. The value should be numm
27        if the 2nd management CID doesn't exist."
28       REFERENCE
29        "Section 11.7.2.1 in IEEE 802.16REvD/D3-2004"
30       ::= { wmanIfBsRegisteredSsEntry 15 }
31
32     wmanIfBsSSMacCsSupportBitMap OBJECT-TYPE
33       SYNTAX      BITS {atm(0),
34                          packetIpv4(1),
35                          packetIpv6(2),
36                          packet802-3(3),
37                          packet802-1Q(4),
38                          packetIpv4Over802-3(5),
39                          packetIpv6Over802-3(6),
40                          packetIpv4Over802-1Q(7),
41                          packetIpv6Over802-1Q(8)}
42       MAX-ACCESS  read-only
43       STATUS      current
44       DESCRIPTION
45        "This object indicates the set of MAC convergence
46        sublayer support. When a bit is set, it indicates
47        the corresponding CS feature is supported."
48       REFERENCE
49        "Section 11.7.5.1 in IEEE 802.16REvD/D3-2004"
50       ::= { wmanIfBsRegisteredSsEntry 16 }
51
52     wmanIfBsSSMaxNumOfClassifier OBJECT-TYPE
53       SYNTAX      INTEGER
54       MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object indicates the maximum number of admitted
4          Classifiers that the SS is allowed to have."
5      REFERENCE
6          "Section 11.7.5.2 in IEEE 802.16REvD/D3-2004"
7      ::= { wmanIfBsRegisteredSsEntry 17 }
8
9      wmanIfBsSSPhsSupport OBJECT-TYPE
10     SYNTAX      INTEGER {noPhsSupport(0),
11                 atmPhsSupport(1),
12                 packetPhsSupport(2)}
13     MAX-ACCESS  read-only
14     STATUS      current
15     DESCRIPTION
16         "This object indicates indicates the level of PHS support."
17     REFERENCE
18         "Section 11.7.5.3 in IEEE 802.16REvD/D3-2004"
19     ::= { wmanIfBsRegisteredSsEntry 18 }
20
21     --
22     -- wmanIfBsPacketCs contain the Base Station Packet Convergence Sublayer
23     -- objects
24     wmanIfBsPacketCs OBJECT IDENTIFIER ::= { wmanIfBsObjects 2 }
25
26     wmanUlschedulingType ::= TEXTUAL-CONVENTION
27         STATUS      current
28         DESCRIPTION
29             "The scheduling service provided by a SC for an
30             upstream service flow. If the parameter is omitted
31             from an upstream QOS Parameter Set, this object takes
32             the value of bestEffort (2). This parameter must be
33             reported as undefined (1) for downstream QOS Parameter
34             Sets."
35         SYNTAX      INTEGER {undefined(1),
36                             bestEffort(2),
37                             nonRealTimePollingService(3),
38                             realTimePollingService(4),
39                             unsolicitedGrantService(6)}
40
41     wmanIfBsProvisionedSfTable OBJECT-TYPE
42     SYNTAX      SEQUENCE OF wmanIfBsProvisionedSfEntry
43     MAX-ACCESS  not-accessible
44     STATUS      current
45     DESCRIPTION
46         "This table is doubly indexed (SS MAC address, SF ID) and
47         contains pre-provisioned service flow profiles, Per SS.
48         These connection parameters shall be provisioned for the SS
49         using DSA messages. NMS shall pre-provisioning the service
50         class table - wmanIfBsServiceClassTable by using
51         wmanIfBsServiceClassIndex, and packet classifier rule table
52         - wmanIfBsClassifierRuleTable by using wmanIfBSSfId"
53     REFERENCE
54         "Section 6.4.13 in IEEE 802.16REvD/D3-2004"

```

```

1      ::= { wmanIfBsPacketCs 1 }
2
3  wmanIfBsProvisionedSfEntry OBJECT-TYPE
4      SYNTAX      WmanIfBsProvisionedSfEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8          "This table provides one row for each service flow been
9          pre-provisioned by NMS."
10     INDEX { wmanIfBsSsProvMacAddress, wmanIfBsSfId}
11     ::= { wmanIfBsProvisionedSfTable 1 }
12
13  wmanIfBsProvisionedSfEntry ::= SEQUENCE {
14      wmanIfBsSfId                Unsigned32,
15      wmanIfBsSsProvMacAddress    MacAddress,
16      wmanIfBsSfDirection        INTEGER,
17      wmanIfBsServiceClassIndex  INTEGER,
18      wmanIfBsSfState            INTEGER,
19      wmanIfBsSfProvisionedTime  TimeStamp,
20      wmanIfBsProvisionedSfRowStatus RowStatus
21  }
22
23  wmanIfBsSfId OBJECT-TYPE
24      SYNTAX      Unsigned32 (1 .. 4294967295)
25      MAX-ACCESS  not-accessible
26      STATUS      current
27      DESCRIPTION
28          "A 32 bit quantity that uniquely identifies a service flow
29          to both the subscriber station and base station (BS)."
30      ::= { wmanIfBsProvisionedSfEntry 1 }
31
32  wmanIfBsSsProvMacAddress OBJECT-TYPE
33      SYNTAX      MacAddress
34      MAX-ACCESS  not-accessible
35      STATUS      current
36      DESCRIPTION
37          "The MAC address of the SS, where the service flow resides.
38          It can be used as the index to associate service flows
39          with the SS."
40      ::= { wmanIfBsProvisionedSfEntry 2 }
41
42  wmanIfBsSfDirection OBJECT-TYPE
43      SYNTAX      INTEGER {downstream(1),
44                      upstream(2)}
45      MAX-ACCESS  read-create
46      STATUS      current
47      DESCRIPTION
48          "An attribute indicating the service flow is downstream or
49          upstream."
50      ::= { wmanIfBsProvisionedSfEntry 3 }
51
52  wmanIfBsServiceClassIndex OBJECT-TYPE
53      SYNTAX      INTEGER
54      MAX-ACCESS  read-create

```

```

1      STATUS      current
2      DESCRIPTION
3          "The index in wmanIfBsServiceClasTable describing the
4          service class or QoS parameters for such service flow.
5          If no associated entry in wmanIfBsServiceClasTable
6          exists, this object returns a value of zero."
7      ::= { wmanIfBsProvisionedSfEntry 4 }
8
9      wmanIfBsSfState OBJECT-TYPE
10     SYNTAX      INTEGER {provisionedState(1),
11                 admittedState(2),
12                 activeState(3)}
13     MAX-ACCESS  read-create
14     STATUS      current
15     DESCRIPTION
16         "wmanIfBsSfState determines how the service flow will be
17         transitioned to the Admitted or Active state.
18         Admitted or Active state: The pre-provisioned service flow
19         will be transitioned to the Admitted or Active state, as
20         soon as the SS passes the network entry procedure, and
21         connection admission control. An entry will be created
22         in the SS and BS service flow tables.
23         Provisioned state: After SS enters the network; the
24         pre-provisioned service flow will remain in the Provisioned
25         state until NMS set it different state. An entry will be
26         created in the SS and BS service flow tables"
27     REFERENCE
28         "Section 6.4.13.6, in IEEE 802.16REvD/D3-2004"
29     ::= { wmanIfBsProvisionedSfEntry 5 }
30
31     wmanIfBsSfProvisionedTime OBJECT-TYPE
32     SYNTAX      TimeStamp
33     MAX-ACCESS  read-create
34     STATUS      current
35     DESCRIPTION
36         "Indicates the data and time when the service flow is
37         provisioned."
38     ::= { wmanIfBsProvisionedSfEntry 6 }
39
40     wmanIfBsProvisionedSfRowStatus OBJECT-TYPE
41     SYNTAX      RowStatus
42     MAX-ACCESS  read-create
43     STATUS      current
44     DESCRIPTION
45         "This object is used to create a new row or modify or
46         delete an existing row in this table.
47
48         If the implementator of this MIB has chosen not
49         to implement 'dynamic assignment' of profiles, this
50         object is not useful and should return noSuchName
51         upon SNMP request."
52     ::= { wmanIfBsProvisionedSfEntry 7 }
53
54     wmanIfBsServiceClasTable OBJECT-TYPE

```



```

1      SYNTAX      SEQUENCE OF WmanIfBsServiceClassEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table is provisioned and is indexed by
6          wmanIfBsQoSProfileIndex. Each entry of the table contains
7          corresponding service flow characteristic attributes
8          (e.g. QoS parameter set). The value of
9          wmanIfBsQoSProfileIndex is obtained from
10         wmanIfBsServiceClassIndex in wmanIfBsProvisionedSfTable"
11     REFERENCE
12         "Section 6.4.13.4 in IEEE 802.16REvD/D3-2004"
13     ::= { wmanIfBsPacketCs 2 }
14
15     WmanIfBsServiceClassEntry OBJECT-TYPE
16         SYNTAX      WmanIfBsServiceClassEntry
17         MAX-ACCESS  not-accessible
18         STATUS      current
19         DESCRIPTION
20             "This table provides one row for each service class"
21         INDEX { wmanIfBsQoSProfileIndex }
22         ::= { wmanIfBsServiceClassTable 1 }
23
24     WmanIfBsServiceClassEntry ::= SEQUENCE {
25         wmanIfBsQoSProfileIndex          INTEGER,
26         wmanIfBsQoSServiceClassName     DisplayString,
27         wmanIfBsQoSTrafficPriority       INTEGER,
28         wmanIfBsQoSMaxSustainedRate     INTEGER,
29         wmanIfBsQoSMaxTrafficBurst      INTEGER,
30         wmanIfBsQoSMinReservedRate     INTEGER,
31         wmanIfBsQOSToleratedJitter     INTEGER,
32         wmanIfBsQoSMaxLatency           INTEGER,
33         wmanIfBsQoSFixedVsVariableSduInd INTEGER,
34         wmanIfBsQOSSduSize              INTEGER,
35         wmanIfBsQoSScSchedulingType     WmanUlsSchedulingType,
36         wmanIfBsQoSScArqEnable          TruthValue,
37         wmanIfBsQoSScArqWindowSize     INTEGER,
38         wmanIfBsQoSScArqFragmentLifetime INTEGER,
39         wmanIfBsQoSScArqSyncLossTimeout INTEGER,
40         wmanIfBsQoSScArqDeliverInOrder TruthValue,
41         wmanIfBsQoSScArqRxPurgeTimeout INTEGER,
42         wmanIfBsQoSScFragmentLen       INTEGER,
43         wmanIfBsQoSScMinRsvdTolerableRate INTEGER,
44         wmanIfBsQOSServiceClassRowStatus RowStatus
45     }
46
47     WmanIfBsQoSProfileIndex OBJECT-TYPE
48         SYNTAX      INTEGER (1 .. 1000)
49         MAX-ACCESS  not-accessible
50         STATUS      current
51         DESCRIPTION
52             "The index value which uniquely identifies an entry
53             in the wmanIfBsServiceClassTable"
54         ::= { wmanIfBsServiceClassEntry 1 }

```

```

1
2  wmanIfBsQoSServiceClassName OBJECT-TYPE
3      SYNTAX      DisplayString
4      MAX-ACCESS  read-create
5      STATUS      current
6      DESCRIPTION "Refers to the Service Class Name"
7      REFERENCE
8          "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
9      ::= { wmanIfBsServiceClassEntry 2 }
10
11 wmanIfBsQoSTrafficPriority OBJECT-TYPE
12     SYNTAX      INTEGER (0..7)
13     MAX-ACCESS  read-create
14     STATUS      current
15     DESCRIPTION
16         "The value of this parameter specifies the priority
17         assigned to a service flow. For uplink service flows,
18         the BS should use this parameter when determining
19         precedence in request service and grant generation,
20         and the SS shall preferentially select contention
21         Request opportunities for Priority Request CIDs
22         based on this priority. Higher numbers indicate higher
23         priority"
24     REFERENCE
25         "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
26     ::= { wmanIfBsServiceClassEntry 3 }
27
28 wmanIfBsQoSMaxSustainedRate OBJECT-TYPE
29     SYNTAX      INTEGER
30     UNITS       "bps"
31     MAX-ACCESS  read-create
32     STATUS      current
33     DESCRIPTION
34         "This parameter defines the peak information rate
35         of the service. The rate is expressed in bits per
36         second and pertains to the SDUs at the input to
37         the system."
38     REFERENCE
39         "Section 11.13.8 in IEEE 802.16REVd/D3-2004"
40     ::= { wmanIfBsServiceClassEntry 4 }
41
42 wmanIfBsQoSMaxTrafficBurst OBJECT-TYPE
43     SYNTAX      INTEGER
44     MAX-ACCESS  read-create
45     STATUS      current
46     DESCRIPTION
47         "This parameter defines the maximum burst size that
48         must be accommodated for the service."
49     REFERENCE
50         "Section 11.13.9 in IEEE 802.16REVd/D3-2004"
51     ::= { wmanIfBsServiceClassEntry 5 }
52
53 wmanIfBsQoSMinReservedRate OBJECT-TYPE
54     SYNTAX      INTEGER

```

```

1      UNITS      "bps"
2      MAX-ACCESS read-create
3      STATUS     current
4      DESCRIPTION
5          "This parameter specifies the minimum rate reserved
6          for this service flow."
7      REFERENCE
8          "Section 11.13.10 in IEEE 802.16REVd/D3-2004"
9      ::= { wmanIfBsServiceClassEntry 6 }
10
11     wmanIfBsQoSoleratedJitter OBJECT-TYPE
12         SYNTAX     INTEGER
13         UNITS      "millisecond"
14         MAX-ACCESS read-create
15         STATUS     current
16         DESCRIPTION
17             "This parameter defines the Maximum delay
18             variation (jitter) for the connection."
19         REFERENCE
20             "Section 11.13.15 in IEEE 802.16REVd/D3-2004"
21         ::= { wmanIfBsServiceClassEntry 7 }
22
23     wmanIfBsQoSMaxLatency OBJECT-TYPE
24         SYNTAX     INTEGER
25         UNITS      "millisecond"
26         MAX-ACCESS read-create
27         STATUS     current
28         DESCRIPTION
29             "The value of this parameter specifies the maximum
30             latency between the reception of a packet by the BS
31             or SS on its network interface and the forwarding
32             of the packet to its RF Interface."
33         REFERENCE
34             "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
35         ::= { wmanIfBsServiceClassEntry 8 }
36
37     wmanIfBsQoSFixedVsVariableSduInd OBJECT-TYPE
38         SYNTAX     INTEGER (0..1)
39         MAX-ACCESS read-create
40         STATUS     current
41         DESCRIPTION
42             "The value of this parameter specifies whether the SDUs
43             on the service flow are fixed-length (0) or
44             variable-length (1). The parameter is used only if
45             packing is on for the service flow. The default value
46             is 0, i.e., variable-length SDUs."
47         REFERENCE
48             "Section 11.13.16 in IEEE 802.16REVd/D4-2004"
49         DEFVAL     { 0 }
50         ::= { wmanIfBsServiceClassEntry 9 }
51
52     wmanIfBsQoSsduSize OBJECT-TYPE
53         SYNTAX     INTEGER
54         MAX-ACCESS read-create

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this parameter specifies the length of the
4          SDU for a fixed-length SDU service flow. This parameter
5          is used only if packing is on and the service flow is
6          indicated as carrying fixed-length SDUs. The default
7          value is 49 bytes, i.e., VC-switched ATM cells with PHS.
8          The parameter is relevant for both ATM and Packet
9          Convergence Sublayers."
10     REFERENCE
11         "Section 11.13.17 in IEEE 802.16REvd/D4-2004"
12     DEFVAL      { 49 }
13     ::= { wmanIfBsServiceClassEntry 10 }
14
15     wmanIfBsQosScSchedulingType OBJECT-TYPE
16         SYNTAX      WmanUlSchedulingType
17         MAX-ACCESS  read-create
18         STATUS      current
19         DESCRIPTION
20             "Specifies the upstream scheduling service used for
21             upstream service flow. If the referenced parameter
22             is not present in the corresponding 802.16 QoS
23             Parameter Set of an upstream service flow, the
24             default value of this object is bestEffort(2)."

```

```

1      STATUS      current
2      DESCRIPTION
3          "The maximum time interval an ARQ fragment will be
4          managed by the transmitter ARQ machine, once
5          initial transmission of the fragment has occurred.
6          If transmission or retransmission of the fragment
7          is not acknowledged by the receiver before the
8          time limit is reached, the fragment is discarded.
9          A value of 0 means Infinite."
10     REFERENCE
11         "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
12     DEFVAL      {0}
13     ::= { wmanIfBsServiceClassEntry 14 }
14
15     wmanIfBsQosScArqSyncLosTimeout OBJECT-TYPE
16         SYNTAX      INTEGER (0 .. 65535 )
17         UNITS       "10 us"
18         MAX-ACCESS  read-create
19         STATUS      current
20         DESCRIPTION
21             "The maximum interval before declaring a loss
22             of synchronization of the sender and receiver
23             state machines. A value of 0 means Infinite."
24         REFERENCE
25             "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
26         DEFVAL      {0}
27         ::= { wmanIfBsServiceClassEntry 15 }
28
29     wmanIfBsQosScArqDeliverInOrder OBJECT-TYPE
30         SYNTAX      TruthValue
31         MAX-ACCESS  read-create
32         STATUS      current
33         DESCRIPTION
34             "Indicates whether or not data is to be delivered
35             by the receiving MAC to its client application
36             in the order in which data was handed off to the
37             originating MAC."
38         REFERENCE
39             "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
40         ::= { wmanIfBsServiceClassEntry 16 }
41
42     wmanIfBsQosScArqRXPurgeTimeout OBJECT-TYPE
43         SYNTAX      INTEGER (0 .. 65535)
44         UNITS       "10 us"
45         MAX-ACCESS  read-create
46         STATUS      current
47         DESCRIPTION
48             "Indicates the time interval the ARQ window is advanced
49             after a fragment is received. A value of 0 means
50             Infinite."
51         REFERENCE
52             "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
53         DEFVAL      {0}
54         ::= { wmanIfBsServiceClassEntry 17 }

```

```

1
2  wmanIfBsQosScFragmentLen OBJECT-TYPE
3      SYNTAX      INTEGER (32 .. 2040)
4      MAX-ACCESS  read-create
5      STATUS      current
6      DESCRIPTION
7          "The maximum size fragment a transmitter shall form
8          or a receiver shall expect to receive."
9      ::= { wmanIfBsServiceClassEntry 18 }
10
11 wmanIfBsQosSCMinRsvdTolerableRate OBJECT-TYPE
12     SYNTAX      INTEGER
13     UNITS       "bps"
14     MAX-ACCESS  read-create
15     STATUS      current
16     DESCRIPTION
17         "Minimum Tolerable Traffic Rate = R (bits/sec) with
18         time base T(sec) means the following. Let S denote
19         additional demand accumulated at the MAC SAP of the
20         transmitter during an arbitrary time interval of the
21         length T. Then the amount of data forwarded at the
22         receiver to CS (in bits) during this interval should
23         be not less than min {S, R * T}."
24     REFERENCE   "Section 11.13.11 in IEEE 802.16REvd/D3-2004"
25     ::= { wmanIfBsServiceClassEntry 19 }
26
27 wmanIfBsQoSServiceClassRowStatus OBJECT-TYPE
28     SYNTAX      RowStatus
29     MAX-ACCESS  read-create
30     STATUS      current
31     DESCRIPTION
32         "This object is used to create a new row or modify or
33         delete an existing row in this table.
34
35         If the implementator of this MIB has chosen not
36         to implement 'dynamic assignment' of profiles, this
37         object is not useful and should return noSuchName
38         upon SNMP request."
39     ::= { wmanIfBsServiceClassEntry 20 }
40
41 wmanIfBsClassifierRuleTable OBJECT-TYPE
42     SYNTAX      SEQUENCE OF wmanIfBsClassifierRuleEntry
43     MAX-ACCESS  not-accessible
44     STATUS      current
45     DESCRIPTION
46         "This table contains packet classifier rules associated
47         with service flows."
48     REFERENCE   "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
49     ::= { wmanIfBsPacketCs 3 }
50
51
52 wmanIfBsClassifierRuleEntry OBJECT-TYPE
53     SYNTAX      wmanIfBsClassifierRuleEntry
54     MAX-ACCESS  not-accessible

```

```

1      STATUS      current
2      DESCRIPTION
3          "This table provides one row for each packet classifier
4          rule, and is indexed by wmanIfBsSfId and
5          wmanIfBsClassifierRuleIndex. wmanIfBsSfIndex
6          identifies the service flow, while
7          wmanIfBsClassifierRuleIndex identifies the packet
8          classifier rule."
9      INDEX { wmanIfBsSfIndex, wmanIfBsClassifierRuleIndex }
10     ::= { wmanIfBsClassifierRuleTable 1 }
11
12     wmanIfBsClassifierRuleEntry ::= SEQUENCE {
13         wmanIfBsSfIndex                Unsigned32,
14         wmanIfBsClassifierRuleIndex     Unsigned32,
15         wmanIfBsClassifierRulePriority   INTEGER,
16         wmanIfBsClassifierRuleIpTosLow  OCTET STRING,
17         wmanIfBsClassifierRuleIpTosHigh OCTET STRING,
18         wmanIfBsClassifierRuleIpTosMask OCTET STRING,
19         wmanIfBsClassifierRuleIpProtocol Integer32,
20         wmanIfBsClassifierRuleInetAddressType InetAddressType,
21         wmanIfBsClassifierRuleInetAddress InetAddress,
22         wmanIfBsClassifierRuleInetAddressMask InetAddress,
23         wmanIfBsClassifierRuleInetAddress InetAddress,
24         wmanIfBsClassifierRuleInetAddressMask InetAddress,
25         wmanIfBsClassifierRuleSourcePortStart Integer32,
26         wmanIfBsClassifierRuleSourcePortEnd Integer32,
27         wmanIfBsClassifierRuleDestPortStart Integer32,
28         wmanIfBsClassifierRuleDestPortEnd Integer32,
29         wmanIfBsClassifierRuleDestMacAddr MacAddress,
30         wmanIfBsClassifierRuleDestMacMask MacAddress,
31         wmanIfBsClassifierRuleSourceMacAddr MacAddress,
32         wmanIfBsClassifierRuleSourceMacMask MacAddress,
33         wmanIfBsClassifierRuleEnetProtocolType INTEGER,
34         wmanIfBsClassifierRuleEnetProtocol Integer32,
35         wmanIfBsClassifierRuleUserPriLow Integer32,
36         wmanIfBsClassifierRuleUserPriHigh Integer32,
37         wmanIfBsClassifierRuleVlanId Integer32,
38         wmanIfBsClassifierRuleState INTEGER,
39         wmanIfBsClassifierRulePkts Counter64,
40         wmanIfBsClassifierRuleRowStatus RowStatus
41     }
42
43     wmanIfBsSfIndex OBJECT-TYPE
44         SYNTAX      Unsigned32 (1 .. 4294967295)
45         MAX-ACCESS  not-accessible
46         STATUS      current
47         DESCRIPTION
48             "A 32 bit quantity that uniquely identifies a service flow
49             to both the subscriber station and base station (BS)."
```

```

50     ::= { wmanIfBsClassifierRuleEntry 1 }
51
52     wmanIfBsClassifierRuleIndex OBJECT-TYPE
53         SYNTAX      Unsigned32 (1..4294967295)
54         MAX-ACCESS  not-accessible
```

```

1      STATUS      current
2      DESCRIPTION
3          "An index is assigned to a classifier in BS classifiers
4          table"
5      ::= { wmanIfBsClassifierRuleEntry 2 }
6
7      wmanIfBsClassifierRulePriority OBJECT-TYPE
8          SYNTAX      INTEGER
9          MAX-ACCESS  read-create
10         STATUS      current
11         DESCRIPTION
12             "The value specifies the priority for the Classifier, which
13             is used for determining the order of the Classifier. A
14             higher value indicates higher priority. Classifiers may have
15             priorities in the range 0..255 with the default value = 0."
16         REFERENCE
17             "Section 11.13.21.3.4.3 in IEEE 802.16REvd/D4-2004"
18         DEFVAL      { 0 }
19         ::= { wmanIfBsClassifierRuleEntry 3 }
20
21         wmanIfBsClassifierRuleIpTosLow OBJECT-TYPE
22             SYNTAX      OCTET STRING (SIZE(1))
23             MAX-ACCESS  read-create
24             STATUS      current
25             DESCRIPTION
26                 "The low value of a range of TOS byte values. If the
27                 referenced parameter is not present in a classifier, this
28                 object reports the value of 0."
29             REFERENCE
30                 "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
31             ::= { wmanIfBsClassifierRuleEntry 4 }
32
33         wmanIfBsClassifierRuleIpTosHigh OBJECT-TYPE
34             SYNTAX      OCTET STRING (SIZE(1))
35             MAX-ACCESS  read-create
36             STATUS      current
37             DESCRIPTION
38                 "The 8-bit high value of a range of TOS byte values.
39                 If the referenced parameter is not present in a classifier,
40                 this object reports the value of 0."
41             REFERENCE
42                 "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
43             ::= { wmanIfBsClassifierRuleEntry 5 }
44
45         wmanIfBsClassifierRuleIpTosMask OBJECT-TYPE
46             SYNTAX      OCTET STRING (SIZE(1))
47             MAX-ACCESS  read-create
48             STATUS      current
49             DESCRIPTION
50                 "The mask value is bitwise ANDed with TOS byte in an IP
51                 packet and this value is used check range checking of
52                 TosLow and TosHigh. If the referenced parameter is not
53                 present in a classifier, this object reports the value
54                 of 0."

```



```
1      REFERENCE
2          "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
3      ::= { wmanIfBsClassifierRuleEntry 6 }
4
5  wmanIfBsClassifierRuleIpProtocol OBJECT-TYPE
6      SYNTAX      Integer32 (0..255)
7      MAX-ACCESS  read-create
8      STATUS      current
9      DESCRIPTION
10         "This object indicates the value of the IP Protocol field
11         required for IP packets to match this rule. If the
12         referenced parameter is not present in a classifier, this
13         object reports the value of 0."
14     REFERENCE
15         "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
16     ::= { wmanIfBsClassifierRuleEntry 7 }
17
18  wmanIfBsClassifierRuleInetAddressType OBJECT-TYPE
19      SYNTAX      InetAddressType
20      MAX-ACCESS  read-create
21      STATUS      current
22      DESCRIPTION
23         "The type of the internet address for
24         wmanIfBsClassifierRuleInetAddressSource,
25         wmanIfBsClassifierRuleInetAddressSourceMask,
26         wmanIfBsClassifierRuleInetAddressDest, and
27         wmanIfBsClassifierRuleInetAddressDestMask.
28         If the referenced parameter is not present in a classifier,
29         this object reports the value of ipv4(1)."
30     REFERENCE
31         "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
32     ::= { wmanIfBsClassifierRuleEntry 8 }
33
34  wmanIfBsClassifierRuleInetAddressSource OBJECT-TYPE
35      SYNTAX      InetAddress
36      MAX-ACCESS  read-create
37      STATUS      current
38      DESCRIPTION
39         "This object specifies the value of the IP Source Address
40         required for packets to match this rule. An IP packet
41         matches the rule when the packet ip source address bitwise
42         ANDed with the wmanIfBsClassifierRuleInetAddressSourceMask value
43         equals the wmanIfBsClassifierRuleInetAddressSource value.
44         If the referenced parameter is not present n a classifier,
45         this object reports the value of 0.0.0.0."
46     REFERENCE
47         "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
48     ::= { wmanIfBsClassifierRuleEntry 9 }
49
50  wmanIfBsClassifierRuleInetAddressSourceMask OBJECT-TYPE
51      SYNTAX      InetAddress
52      MAX-ACCESS  read-create
53      STATUS      current
54      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         wmanIfBsClassifierRuleInetSourceMask value equals the
6         wmanIfBsClassifierRuleInetSourceAddr value.
7         If the referenced parameter is not present in a classifier,
8         this object reports the value of 0.0.0.0."
9     REFERENCE
10        "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
11        ::= { wmanIfBsClassifierRuleEntry 10 }
12
13     wmanIfBsClassifierRuleInetDestAddr OBJECT-TYPE
14         SYNTAX      InetAddress
15         MAX-ACCESS  read-create
16         STATUS      current
17         DESCRIPTION
18             "This object specifies the value of the IP Destination
19             Address required for packets to match this rule. An IP
20             packet matches the rule when the packet IP destination
21             address bitwise ANDED with the
22             wmanIfBsClassifierRuleInetDestMask value equals the
23             wmanIfBsClassifierRuleInetDestAddr value.
24             If the referenced parameter is not present in a
25             classifier, this object reports the value of 0.0.0.0."
26         REFERENCE
27             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
28             ::= { wmanIfBsClassifierRuleEntry 11 }
29
30     wmanIfBsClassifierRuleInetDestMask OBJECT-TYPE
31         SYNTAX      InetAddress
32         MAX-ACCESS  read-create
33         STATUS      current
34         DESCRIPTION
35             "This object specifies which bits of a packet's IP
36             Destination Address that are compared to match this rule.
37             An IP packet matches the rule when the packet destination
38             address bitwise ANDED with the
39             wmanIfBsClassifierRuleInetDestMask value equals the
40             wmanIfBsClassifierRuleInetDestAddr value.
41             If the referenced parameter is not present in a classifier
42             , this object reports the value of 0.0.0.0."
43         REFERENCE
44             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
45             ::= { wmanIfBsClassifierRuleEntry 12 }
46
47     wmanIfBsClassifierRuleSourcePortStart OBJECT-TYPE
48         SYNTAX      Integer32 (0..65535)
49         MAX-ACCESS  read-create
50         STATUS      current
51         DESCRIPTION
52             "This object specifies the low end inclusive range of
53             TCP/UDP source port numbers to which a packet is compared
54             . This object is irrelevant for non-TCP/UDP IP packets.

```

```

1           If the referenced parameter is not present in a
2           classifier, this object reports the value of 0."
3   REFERENCE
4           "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
5   ::= { wmanIfBsClassifierRuleEntry 13 }
6
7   wmanIfBsClassifierRuleSourcePortEnd OBJECT-TYPE
8       SYNTAX      Integer32 (0..65535)
9       MAX-ACCESS  read-create
10      STATUS      current
11      DESCRIPTION
12          "This object specifies the high end inclusive range of
13          TCP/UDP source port numbers to which a packet is compared.
14          This object is irrelevant for non-TCP/UDP IP packets.
15          If the referenced parameter is not present in a classifier,
16          this object reports the value of 65535."
17      REFERENCE
18          "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
19      ::= { wmanIfBsClassifierRuleEntry 14 }
20
21   wmanIfBsClassifierRuleDestPortStart OBJECT-TYPE
22      SYNTAX      Integer32 (0..65535)
23      MAX-ACCESS  read-create
24      STATUS      current
25      DESCRIPTION
26          "This object specifies the low end inclusive range of
27          TCP/UDP destination port numbers to which a packet is
28          compared. If the referenced parameter is not present
29          in a classifier, this object reports the value of 0."
30      REFERENCE
31          "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
32      ::= { wmanIfBsClassifierRuleEntry 15 }
33
34   wmanIfBsClassifierRuleDestPortEnd OBJECT-TYPE
35      SYNTAX      Integer32 (0..65535)
36      MAX-ACCESS  read-create
37      STATUS      current
38      DESCRIPTION
39          "This object specifies the high end inclusive range of
40          TCP/UDP destination port numbers to which a packet is
41          compared. If the referenced parameter is not present
42          in a classifier, this object reports the value of
43          65535."
44      REFERENCE
45          "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
46      ::= { wmanIfBsClassifierRuleEntry 16 }
47
48   wmanIfBsClassifierRuleDestMacAddr OBJECT-TYPE
49      SYNTAX      MacAddress
50      MAX-ACCESS  read-create
51      STATUS      current
52      DESCRIPTION
53          "An Ethernet packet matches an entry when its destination
54          MAC address bitwise ANDed with

```

```

1         wmanIfBsClassifierRuleDestMacMask equals the value of
2         wmanIfBsClassifierRuleDestMacAddr. If the referenced
3         parameter is not present in a classifier, this object
4         reports the value of '000000000000'H."
5     REFERENCE
6         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
7     ::= { wmanIfBsClassifierRuleEntry 17 }
8
9     wmanIfBsClassifierRuleDestMacMask OBJECT-TYPE
10        SYNTAX      MacAddress
11        MAX-ACCESS  read-create
12        STATUS      current
13        DESCRIPTION
14            "An Ethernet packet matches an entry when its destination
15            MAC address bitwise ANDed with
16            wmanIfBsClassifierRuleDestMacMask equals the value of
17            wmanIfBsClassifierRuleDestMacAddr. If the referenced
18            parameter is not present in a classifier, this object
19            reports the value of '000000000000'H."
20        REFERENCE
21            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
22        ::= { wmanIfBsClassifierRuleEntry 18 }
23
24     wmanIfBsClassifierRuleSourceMacAddr OBJECT-TYPE
25        SYNTAX      MacAddress
26        MAX-ACCESS  read-create
27        STATUS      current
28        DESCRIPTION
29            "An Ethernet packet matches this entry when its source
30            MAC address bitwise ANDed with
31            wmanIfBsClassifierRuleSourceMacMask equals the value
32            of wmanIfBsClassifierRuleSourceMacAddr. If the
33            referenced parameter is not present in a classifier,
34            this object reports the value of '000000000000'H."
35        REFERENCE
36            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
37        ::= { wmanIfBsClassifierRuleEntry 19 }
38
39     wmanIfBsClassifierRuleSourceMacMask OBJECT-TYPE
40        SYNTAX      MacAddress
41        MAX-ACCESS  read-create
42        STATUS      current
43        DESCRIPTION
44            "An Ethernet packet matches an entry when its destination
45            MAC address bitwise ANDed with
46            wmanIfBsClassifierRuleSourceMacMask equals the value of
47            wmanIfBsClassifierRuleSourceMacAddr. If the referenced
48            parameter is not present in a classifier, this object
49            reports the value of '000000000000'H."
50        REFERENCE
51            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
52        ::= { wmanIfBsClassifierRuleEntry 20 }
53
54     wmanIfBsClassifierRuleEnetProtocolType OBJECT-TYPE

```

```

1      SYNTAX      INTEGER {none(0),
2                      ethertype(1),
3                      dsap(2)}
4      MAX-ACCESS  read-create
5      STATUS      current
6      DESCRIPTION
7          "This object indicates the format of the layer 3 protocol
8          id in the Ethernet packet. A value of none(0) means that
9          the rule does not use the layer 3 protocol type as a
10         matching criteria. A value of ethertype(1) means that the
11         rule applies only to frames which contains an EtherType
12         value. Ethertype values are contained in packets using
13         the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
14         Sub-Network Access Protocol (SNAP) encapsulation formats.
15         A value of dsap(2) means that the rule applies only to
16         frames using the IEEE802.3 encapsulation format with a
17         Destination Service Access Point (DSAP) other than 0xAA
18         (which is reserved for SNAP). If the Ethernet frame
19         contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
20         this object applies to the embedded EtherType field within
21         the 802.1P/Q header. If the referenced parameter is not
22         present in a classifier, this object reports the value of
23         0."
24      REFERENCE
25          "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
26      ::= { wmanIfBsClassifierRuleEntry 21 }
27
28      wmanIfBsClassifierRuleEnetProtocol OBJECT-TYPE
29          SYNTAX      Integer32 (0..65535)
30          MAX-ACCESS  read-create
31          STATUS      current
32          DESCRIPTION
33              "If wmanIfBsClassifierRuleEnetProtocolType is none(0),
34              this object is ignored when considering whether a packet
35              matches the current rule.
36              If wmanIfBsClassifierRuleEnetProtocolType is ethertype(1),
37              this object gives the 16-bit value of the EtherType that
38              the packet must match in order to match the rule.
39              If wmanIfBsClassifierRuleEnetProtocolType is dsap(2), the
40              lower 8 bits of this object's value must match the DSAP
41              byte of the packet in order to match the rule.
42              If wmanIfBsClassifierRuleEnetProtocolType is mac(3), the
43              lower 8 bits of this object value represent a lower bound
44              (inclusive) of MAC management message type codes matched,
45              and the upper 8 bits of this object value represent the
46              upper bound (inclusive) of matched MAC message type codes.
47              Certain message type codes are excluded from matching, as
48              specified in the reference.
49              If the Ethernet frame contains an 802.1P/Q Tag header
50              (i.e. EtherType 0x8100), this object applies to the
51              embedded EtherType field within the 802.1P/Q header.
52              If the referenced parameter is not present in the
53              classifier, the value of this object is reported as 0."
54      REFERENCE

```

```

1      "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
2      ::= { wmanIfBsClassifierRuleEntry 22 }
3
4  wmanIfBsClassifierRuleUserPriLow OBJECT-TYPE
5      SYNTAX      Integer32 (0..7)
6      MAX-ACCESS  read-create
7      STATUS      current
8      DESCRIPTION
9          "This object applies only to Ethernet frames using the
10         802.1P/Q tag header (indicated with EtherType 0x8100).
11         Such frames include a 16-bit Tag that contains a 3 bit
12         Priority field and a 12 bit VLAN number.
13         Tagged Ethernet packets must have a 3-bit Priority field
14         within the range of wmanIfBsClassifierRulePriLow and
15         wmanIfBsClassifierRulePriHigh in order to match this
16         rule.
17         If the referenced parameter is not present in the
18         classifier, the value of this object is reported as 0."
19      REFERENCE
20          "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
21      ::= { wmanIfBsClassifierRuleEntry 23 }
22
23  wmanIfBsClassifierRuleUserPriHigh OBJECT-TYPE
24      SYNTAX      Integer32 (0..7)
25      MAX-ACCESS  read-create
26      STATUS      current
27      DESCRIPTION
28          "This object applies only to Ethernet frames using the
29         802.1P/Q tag header (indicated with EtherType 0x8100).
30         Such frames include a 16-bit Tag that contains a 3 bit
31         Priority field and a 12 bit VLAN number.
32         Tagged Ethernet packets must have a 3-bit Priority
33         field within the range of wmanIfBsClassifierRulePriLow
34         and wmanIfBsClassifierRulePriHigh in order to match
35         this rule.
36         If the referenced parameter is not present in the
37         classifier, the value of this object is reported as 7."
38      REFERENCE
39          "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
40      ::= { wmanIfBsClassifierRuleEntry 24 }
41
42  wmanIfBsClassifierRuleVlanId OBJECT-TYPE
43      SYNTAX      Integer32 (0..4095)
44      MAX-ACCESS  read-create
45      STATUS      current
46      DESCRIPTION
47          "This object applies only to Ethernet frames using the
48         802.1P/Q tag header.
49         If this object's value is nonzero, tagged packets must
50         have a VLAN Identifier that matches the value in order
51         to match the rule.
52         Only the least significant 12 bits of this object's
53         value are valid.
54         If the referenced parameter is not present in the

```

```

1         classifier, the value of this object is reported as 0."
2     REFERENCE
3         "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
4     ::= { wmanIfBsClassifierRuleEntry 25 }
5
6     wmanIfBsClassifierRuleState OBJECT-TYPE
7         SYNTAX      INTEGER {active(1),
8                     inactive(2)}
9         MAX-ACCESS  read-create
10        STATUS      current
11        DESCRIPTION
12            "This object indicates whether or not the classifier is
13            enabled to classify packets to a Service Flow.
14            If the referenced parameter is not present in the
15            classifier, the value of this object is reported
16            as active(1)."
17        REFERENCE
18            "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
19        ::= { wmanIfBsClassifierRuleEntry 26 }
20
21        wmanIfBsClassifierRulePkts OBJECT-TYPE
22            SYNTAX      Counter64
23            MAX-ACCESS  read-create
24            STATUS      current
25            DESCRIPTION
26                "This object counts the number of packets that have
27                been classified using this entry."
28            REFERENCE
29                "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
30            ::= { wmanIfBsClassifierRuleEntry 27 }
31
32        wmanIfBsClassifierRuleRowStatus OBJECT-TYPE
33            SYNTAX      RowStatus
34            MAX-ACCESS  read-create
35            STATUS      current
36            DESCRIPTION
37                "This object is used to create a new row or modify or
38                delete an existing row in this table.
39
40                If the implementator of this MIB has chosen not
41                to implement 'dynamic assignment' of profiles, this
42                object is not useful and should return noSuchName
43                upon SNMP request."
44            ::= { wmanIfBsClassifierRuleEntry 28 }
45
46        --
47        -- wmanIfBsCps contain the Base Station Common Part Sublayer objects
48        wmanIfBsCps OBJECT IDENTIFIER ::= { wmanIfBsObjects 3 }
49
50        --
51        -- wmanIfBsConfigurationTable contains global parameters common in BS
52        --
53        wmanIfBsConfigurationTable OBJECT-TYPE
54            SYNTAX      SEQUENCE OF WmanIfBsConfigurationEntry

```

```

1      MAX-ACCESS  not-accessible
2      STATUS      current
3      DESCRIPTION
4          "This table provides one row for each BS sector that
5          contains the BS system parameters as defined in section
6          10.1 of [3]."
7      ::= { wmanIfBsCps 1 }
8
9      wmanIfBsConfigurationEntry OBJECT-TYPE
10     SYNTAX      WmanIfBsConfigurationEntry
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "This table is indexed by ifIndex with an iftype of
15         propBWA2Mp."
16     INDEX { ifIndex }
17     ::= { wmanIfBsConfigurationTable 1 }
18
19     wmanIfBsConfigurationEntry ::= SEQUENCE {
20         wmanIfBsDcdInterval          INTEGER,
21         wmanIfBsUcdInterval          INTEGER,
22         wmanIfBsUcdTransition        INTEGER,
23         wmanIfBsDcdTransition        INTEGER,
24         wmanIfBsMaxMAPPending        INTEGER,
25         wmanIfBsInitialRangingInterval  INTEGER,
26         wmanIfBsClkCmpInterval       INTEGER,
27         wmanIfBsSsULMapProcTime      Unsigned32,
28         wmanIfBsSsRangRespProcTime   Unsigned32,
29         wmanIfBsT5Timeout             INTEGER,
30         wmanIfBsT9Timeout             INTEGER,
31         wmanIfBsT13Timeout           INTEGER,
32         wmanIfBsT15Timeout           INTEGER,
33         wmanIfBsT17Timeout           INTEGER,
34         wmanIfBsT27IdleTimer         INTEGER,
35         wmanIfBsT27ActiveTimer       INTEGER,
36         wmanIfBsConfigurationRowStatus  RowStatus
37     }
38
39     wmanIfBsDcdInterval OBJECT-TYPE
40     SYNTAX      INTEGER(0..10000)
41     UNITS       "milliseconds"
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "Time between transmission of DCD messages in ms."
46     ::= { wmanIfBsConfigurationEntry 1 }
47
48     wmanIfBsUcdInterval OBJECT-TYPE
49     SYNTAX      INTEGER(0..10000)
50     UNITS       "milliseconds"
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54         "Time between transmission of UCD messages in ms."

```



```

1      ::= { wmanIfBsConfigurationEntry 2 }
2
3  wmanIfBsUcdTransition OBJECT-TYPE
4      SYNTAX      INTEGER
5      UNITS       "Number of MAC Frames"
6      MAX-ACCESS  read-write
7      STATUS      current
8      DESCRIPTION
9          "The time the BS shall wait after repeating a UCD message
10         with an incremented Configuration Change Count before
11         issuing a UL-MAP message referring to
12         Downlink_Burst_Profiles defined in that UCD message."
13     ::= { wmanIfBsConfigurationEntry 3 }
14
15  wmanIfBsDcdTransition OBJECT-TYPE
16      SYNTAX      INTEGER
17      UNITS       "Number of MAC Frames"
18      MAX-ACCESS  read-write
19      STATUS      current
20      DESCRIPTION
21          "The time the BS shall wait after repeating a DCD message
22         with an incremented Configuration Change Count before
23         issuing a DL-MAP message referring to
24         Uplink_Burst_Profiles
25         defined in that DCD message."
26     ::= { wmanIfBsConfigurationEntry 4 }
27
28  wmanIfBsMaxMAPPending OBJECT-TYPE
29      SYNTAX      INTEGER
30      MAX-ACCESS  read-write
31      STATUS      current
32      DESCRIPTION
33          "Maximum validity of map."
34     ::= { wmanIfBsConfigurationEntry 5 }
35
36  wmanIfBsInitialRangingInterval OBJECT-TYPE
37      SYNTAX      INTEGER(0..2000)
38      UNITS       "milliseconds"
39      MAX-ACCESS  read-write
40      STATUS      current
41      DESCRIPTION
42          "Time between Initial Ranging regions assigned by the BS
43         in ms."
44     ::= { wmanIfBsConfigurationEntry 6 }
45
46  wmanIfBsClkCmpInterval OBJECT-TYPE
47      SYNTAX      INTEGER(50..50)
48      UNITS       "milliseconds"
49      MAX-ACCESS  read-only
50      STATUS      current
51      DESCRIPTION
52          "Time between the clock compare measurements used for the
53         generation of CLK-CMP messages."
54     ::= { wmanIfBsConfigurationEntry 7 }

```

```

1
2  wmanIfBsSsULMapProcTime OBJECT-TYPE
3      SYNTAX      Unsigned32 (200 .. 4294967295)
4      UNITS       "micro seconds"
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8          "Time provided between arrival of the last bit of a UL-MAP
9          at an SS and effectiveness of that map in us."
10     ::= { wmanIfBsConfigurationEntry 8 }
11
12  wmanIfBsSsRangRespProcTime 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     ::= { wmanIfBsConfigurationEntry 9 }
22
23  wmanIfBsT5Timeout OBJECT-TYPE
24      SYNTAX      INTEGER(0 .. 2000)
25      UNITS       "milliseconds"
26      MAX-ACCESS  read-write
27      STATUS      current
28      DESCRIPTION
29          "Wait for Uplink Channel Change Response in ms."
30     ::= { wmanIfBsConfigurationEntry 10 }
31
32  wmanIfBsT9Timeout OBJECT-TYPE
33      SYNTAX      INTEGER(300 .. 65535)
34      UNITS       "milliseconds"
35      MAX-ACCESS  read-write
36      STATUS      current
37      DESCRIPTION
38          "Registration Timeout, the time allowed between the BS
39          sending a RNG-RSP (success) to an SS, and receiving a
40          SBC-REQ from that same SS in ms."
41     ::= { wmanIfBsConfigurationEntry 11 }
42
43  wmanIfBsT13Timeout OBJECT-TYPE
44      SYNTAX      INTEGER(15 .. 65535)
45      UNITS       "minutes"
46      MAX-ACCESS  read-write
47      STATUS      current
48      DESCRIPTION
49          "The time allowed for an SS, following receipt of a
50          REG-RSP message to send a TFTP-CPLT message to the BS
51          in min."
52     ::= { wmanIfBsConfigurationEntry 12 }
53
54  wmanIfBsT15Timeout OBJECT-TYPE

```

```

1          SYNTAX      INTEGER(20 .. 65535)
2          UNITS        "milliseconds"
3          MAX-ACCESS  read-write
4          STATUS      current
5          DESCRIPTION
6              "Wait for MCA-RSP in ms."
7          ::= { wmanIfBsConfigurationEntry 13 }
8
9  wmanIfBsT17Timeout OBJECT-TYPE
10         SYNTAX      INTEGER(5 .. 65535)
11         UNITS        "minutes"
12         MAX-ACCESS  read-write
13         STATUS      current
14         DESCRIPTION
15             "Time allowed for SS to complete SS Authorization and
16             Key Exchange in min ."
17         ::= { wmanIfBsConfigurationEntry 14 }
18
19  wmanIfBsT27IdleTimer OBJECT-TYPE
20         SYNTAX      INTEGER
21         UNITS        "milliseconds"
22         MAX-ACCESS  read-write
23         STATUS      current
24         DESCRIPTION
25             "Maximum time between unicast grants to SS when BS believes
26             SS uplink transmission quality is good enough."
27         ::= { wmanIfBsConfigurationEntry 15 }
28
29  wmanIfBsT27ActiveTimer OBJECT-TYPE
30         SYNTAX      INTEGER
31         UNITS        "milliseconds"
32         MAX-ACCESS  read-write
33         STATUS      current
34         DESCRIPTION
35             "Maximum time between unicast grants to SS when BS believes
36             SS uplink transmission quality is not good enough."
37         ::= { wmanIfBsConfigurationEntry 16 }
38
39  wmanIfBsConfigurationRowStatus OBJECT-TYPE
40         SYNTAX      RowStatus
41         MAX-ACCESS  read-create
42         STATUS      current
43         DESCRIPTION
44             "This object is used to create a new row or modify or
45             delete an existing row in this table.
46
47             If the implementator of this MIB has chosen not
48             to implement 'dynamic assignment' of profiles, this
49             object is not useful and should return noSuchName
50             upon SNMP request."
51         ::= { wmanIfBsConfigurationEntry 17 }
52
53  --
54  -- Base Station statistics counters

```

```

1  --
2  wmanIfBsStatCounter OBJECT IDENTIFIER ::= { wmanIfBsCps 2 }
3
4  wmanIfBsChMeasurementTable OBJECT-TYPE
5      SYNTAX      SEQUENCE OF      wmanIfBsChMeasurementEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "This table contains channel measurement information
10         on the uplink signal received from SS. The table shall
11         be maintained as FIFO to store measurement samples that
12         can be used to create RSSI and CINR histogram report.
13         When the measurement entry for a SS reaches the limit,
14         the oldest entry shall be deleted as the new entry is
15         added to the table."
16     ::= { wmanIfBsStatCounter 1 }
17
18  wmanIfBsChMeasurementEntry OBJECT-TYPE
19      SYNTAX      wmanIfBsChMeasurementEntry
20      MAX-ACCESS  not-accessible
21      STATUS      current
22      DESCRIPTION
23          "Each entry in the table contains RSSI and CINR
24          signal quality measurement on signal received from the SS.
25          The primary index is the ifIndex with ifType of propBWA2Mp
26          identifying the BS sector. wmanIfChSsidIndex identifies
27          the SS from which the signal was received.
28          wmanIfBsHistogramIndex is the index to histogram samples.
29          Since there is no time stamp in the table,
30          wmanIfBsHistogramIndex should be increased monotonically,
31          and wraps around when it reaches the limit. "
32      INDEX      { ifIndex, wmanIfBsChSsidIndex,
33                  wmanIfBsHistogramIndex }
34     ::= { wmanIfBsChMeasurementTable 1 }
35
36  wmanIfBsChMeasurementEntry ::= SEQUENCE {
37      wmanIfBsChSsidIndex      Unsigned32,
38      wmanIfBsHistogramIndex   Unsigned32,
39      wmanIfBsChannelNumber    INTEGER,
40      wmanIfBsStartFrame       INTEGER,
41      wmanIfBsDuration          INTEGER,
42      wmanIfBsBasicReport       BITS,
43      wmanIfBsMeanCinrReport    INTEGER,
44      wmanIfBsMeanRssiReport    INTEGER
45  }
46
47  wmanIfBsChSsidIndex OBJECT-TYPE
48      SYNTAX      Unsigned32 (1 .. 4294967295)
49      MAX-ACCESS  read-only
50      STATUS      current
51      DESCRIPTION
52          "wmanIfBsChIdIndex identifies the SS providing the
53          channel measurement."
54      REFERENCE

```

```

1         "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
2         ::= { wmanIfBsChMeasurementEntry 1 }
3
4     wmanIfBsHistogramIndex OBJECT-TYPE
5         SYNTAX      Unsigned32 (1 .. 4294967295)
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "wmanIfBsHistogramIndex identifies the histogram samples
10            in the table for each subscriber station."
11         ::= { wmanIfBsChMeasurementEntry 2 }
12
13     wmanIfBsChannelNumber OBJECT-TYPE
14         SYNTAX      INTEGER
15         MAX-ACCESS  read-only
16         STATUS      current
17         DESCRIPTION
18             "Physical channel number to be reported on is only
19            applicable to licence exempt band. For licenced band,
20            this parameter should be null."
21         REFERENCE
22             "Section 8.5.1 in IEEE 802.16REvD/D3-2004"
23         ::= { wmanIfBsChMeasurementEntry 3 }
24
25     wmanIfBsStartFrame OBJECT-TYPE
26         SYNTAX      INTEGER
27         MAX-ACCESS  read-only
28         STATUS      current
29         DESCRIPTION
30             "Frame number in which measurement for this channel
31            started."
32         REFERENCE
33             "Section 11.12 in IEEE 802.16REvD/D3-2004"
34         ::= { wmanIfBsChMeasurementEntry 4 }
35
36     wmanIfBsDuration OBJECT-TYPE
37         SYNTAX      INTEGER
38         MAX-ACCESS  read-only
39         STATUS      current
40         DESCRIPTION
41             "Cumulative measurement duration on the channel in
42            multiples of Ts. For any value exceeding 0xFFFFFFFF,
43            report 0xFFFFFFFF."
44         REFERENCE
45             "Section 11.12 in IEEE 802.16REvD/D3-2004"
46         ::= { wmanIfBsChMeasurementEntry 5 }
47
48     wmanIfBsBasicReport OBJECT-TYPE
49         SYNTAX      BITS {wirelessHuman(0),
50                       unknownTransmission(1),
51                       primaryUser(2),
52                       channegNotMeasured(3)}
53         MAX-ACCESS  read-only
54         STATUS      current

```

```

1      DESCRIPTION
2          "Bit #0: WirelessHUMAN detected on the channel
3          Bit #1: Unknown transmissions detected on the channel
4          Bit #2: Primary User detected on the channel
5          Bit #3: Unmeasured. Channel not measured"
6      REFERENCE
7          "Section 11.12 in IEEE 802.16REvD/D3-2004"
8      ::= { wmanIfBsChMeasurementEntry 6 }
9
10     wmanIfBsMeanCinrReport OBJECT-TYPE
11         SYNTAX      INTEGER
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "Mean CINR report."
16         REFERENCE
17             "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
18             802.16REvD/D3-2004"
19         ::= { wmanIfBsChMeasurementEntry 7 }
20
21     wmanIfBsMeanRssiReport OBJECT-TYPE
22         SYNTAX      INTEGER
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "Mean RSSI report."
27         REFERENCE
28             "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
29             802.16REvD/D3-2004"
30         ::= { wmanIfBsChMeasurementEntry 8 }
31
32     --
33     -- Base station PKM group
34     -- wmanIfBsPkmObjects contain the Base Station Privacy Sublayer objects
35     wmanIfBsPkmObjects OBJECT IDENTIFIER ::= { wmanIfBsObjects 4 }
36
37     --
38     -- Table wmanIfBsPkmBaseTable
39     --
40     wmanIfBsPkmBaseTable OBJECT-TYPE
41         SYNTAX      SEQUENCE OF wmanIfBsPkmBaseEntry
42         MAX-ACCESS  not-accessible
43         STATUS      current
44         DESCRIPTION
45             "This table describes the basic PKM attributes of each Base
46             Station wireless interface."
47         ::= { wmanIfBsPkmObjects 1 }
48
49     wmanIfBsPkmBaseEntry OBJECT-TYPE
50         SYNTAX      wmanIfBsPkmBaseEntry
51         MAX-ACCESS  not-accessible
52         STATUS      current
53         DESCRIPTION
54             "Each entry contains objects describing attributes of one

```

```

1           BS wireless interface."
2     INDEX      { ifIndex }
3     ::= { wmanIfBsPkmBaseTable 1 }
4
5     wmanIfBsPkmBaseEntry ::= SEQUENCE {
6         wmanIfBsPkmDefaultAuthLifetime      Integer32,
7         wmanIfBsPkmDefaultTEKLifetime      Integer32,
8         wmanIfBsPkmDefaultSelfSigManufCertTrust INTEGER,
9         wmanIfBsPkmCheckCertValidityPeriods TruthValue,
10        wmanIfBsPkmAuthentInfos             Counter32,
11        wmanIfBsPkmAuthRequests             Counter32,
12        wmanIfBsPkmAuthReplies             Counter32,
13        wmanIfBsPkmAuthRejects             Counter32,
14        wmanIfBsPkmAuthInvalids            Counter32
15    }
16
17    wmanIfBsPkmDefaultAuthLifetime OBJECT-TYPE
18        SYNTAX      Integer32 (86400..604800)
19        UNITS       "seconds"
20        MAX-ACCESS  read-write
21        STATUS      current
22        DESCRIPTION
23            "The value of this object is the default lifetime, in
24            seconds, the BS assigns to a new authorization key."
25        REFERENCE
26            "IEEE 802.16 standard; Table 270"
27        DEFVAL      { 604800 }
28        ::= { wmanIfBsPkmBaseEntry 1 }
29
30    wmanIfBsPkmDefaultTEKLifetime OBJECT-TYPE
31        SYNTAX      Integer32 (1800..604800)
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 Traffic Encryption
38            Key(TEK)."
39        REFERENCE
40            "IEEE 802.16 standard; Table 270"
41        DEFVAL      { 43200 }
42        ::= { wmanIfBsPkmBaseEntry 2 }
43
44
45    wmanIfBsPkmDefaultSelfSigManufCertTrust OBJECT-TYPE
46        SYNTAX      INTEGER { trusted (1),
47                        untrusted (2) }
48        MAX-ACCESS  read-write
49        STATUS      current
50        DESCRIPTION
51            "This object determines the default trust of all (new)
52            self-signed manufacturer certificates obtained after
53            setting the object."
54        ::= { wmanIfBsPkmBaseEntry 3 }

```

```

1
2 wmanIfBsPkmCheckCertValidityPeriods OBJECT-TYPE
3     SYNTAX      TruthValue
4     MAX-ACCESS  read-write
5     STATUS      current
6     DESCRIPTION
7         "Setting this object to TRUE causes all certificates
8         received? thereafter to have their validity periods (and
9         their chain's validity periods) checked against the current
10        time of day. A FALSE setting will cause all certificates
11        received? Thereafter to not have their validity periods
12        (nor their chain's validity periods) checked against the
13        current time of day."
14        ::= { wmanIfBsPkmBaseEntry 4 }
15
16 wmanIfBsPkmAuthentInfos OBJECT-TYPE
17     SYNTAX      Counter32
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21         "The value of this object is the count of times the BS has
22         received an Authentication Information message from any
23         SS."
24        ::= { wmanIfBsPkmBaseEntry 5 }
25
26 wmanIfBsPkmAuthRequests OBJECT-TYPE
27     SYNTAX      Counter32
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31         "The value of this object is the count of times the BS has
32         received an Authorization Request message from any SS"
33        ::= { wmanIfBsPkmBaseEntry 6 }
34
35 wmanIfBsPkmAuthReplies OBJECT-TYPE
36     SYNTAX      Counter32
37     MAX-ACCESS  read-only
38     STATUS      current
39     DESCRIPTION
40         "The value of this object is the count of times the BS has
41         transmitted an Authorization Reply message to any SS."
42        ::= { wmanIfBsPkmBaseEntry 7 }
43
44 wmanIfBsPkmAuthRejects OBJECT-TYPE
45     SYNTAX      Counter32
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "The value of this object is the count of times the BS has
50         transmitted an Authorization Reject message to any SS."
51        ::= { wmanIfBsPkmBaseEntry 8 }
52
53 wmanIfBsPkmAuthInvalids OBJECT-TYPE
54     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 an Authorization Invalid message to any SS."
6      ::= { wmanIfBSPkmBaseEntry 9 }
7
8      --
9      -- Table wmanIfBSPkmAuthTable
10     --
11     wmanIfBSPkmAuthTable OBJECT-TYPE
12         SYNTAX      SEQUENCE OF  wmanIfBSPkmAuthEntry
13         MAX-ACCESS  not-accessible
14         STATUS      current
15         DESCRIPTION
16             "This table describes the attributes of each SS
17             authorization association. The BS maintains one
18             authorization association with each Baseline
19             Privacy-enabled SS on each BS wireless interface."
20         ::= { wmanIfBSPkmObjects 2 }
21
22     wmanIfBSPkmAuthEntry OBJECT-TYPE
23         SYNTAX      wmanIfBSPkmAuthEntry
24         MAX-ACCESS  not-accessible
25         STATUS      current
26         DESCRIPTION
27             "Each entry contains objects describing attributes of one
28             authorization association. The BS MUST create one entry per
29             SS per wireless interface, based on the receipt of an
30             Authorization Request message, and MUST not delete the
31             entry before the SS authorization permanently expires."
32         INDEX      { ifIndex, wmanIfBSPkmAuthSsMacAddress }
33         ::= { wmanIfBSPkmAuthTable 1 }
34
35     wmanIfBSPkmAuthEntry ::= SEQUENCE {
36         wmanIfBSPkmAuthSsMacAddress      MacAddress,
37         wmanIfBSPkmAuthSsPublicKey       OCTET STRING,
38         wmanIfBSPkmAuthSsKeySequenceNumber Integer32,
39         wmanIfBSPkmAuthSsExpiresOld      DateAndTime,
40         wmanIfBSPkmAuthSsExpiresNew      DateAndTime,
41         wmanIfBSPkmAuthSsLifetime        Integer32,
42         wmanIfBSPkmAuthSsGraceTime       Integer32,
43         wmanIfBSPkmAuthSsReset           INTEGER,
44         wmanIfBSPkmAuthSsInfos           Counter64,
45         wmanIfBSPkmAuthSsRequests        Counter64,
46         wmanIfBSPkmAuthSsReplies         Counter64,
47         wmanIfBSPkmAuthSsRejects         Counter64,
48         wmanIfBSPkmAuthSsInvalids        Counter64,
49         wmanIfBSPkmAuthRejectErrorCode   INTEGER,
50         wmanIfBSPkmAuthRejectErrorString SnmpAdminString,
51         wmanIfBSPkmAuthInvalidErrorCode  INTEGER,
52         wmanIfBSPkmAuthInvalidErrorString SnmpAdminString,
53         wmanIfBSPkmAuthPrimarySAId       Integer32,
54         wmanIfBSPkmAuthBpkmSsCertValid   INTEGER,

```

```

1          wmanIfBsPkmAuthBpkmSsCert          OCTET STRING
2      }
3
4  wmanIfBsPkmAuthSsMacAddress OBJECT-TYPE
5      SYNTAX      MacAddress
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "The value of this object is the physical address of the SS
10         to which the authorization association applies."
11      ::= { wmanIfBsPkmAuthEntry 1 }
12
13 wmanIfBsPkmAuthSsPublicKey OBJECT-TYPE
14     SYNTAX      OCTET STRING (SIZE (140))
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is a DER-encoded RSAPublicKey
19         ASN.1 type string, as defined in the RSA Encryption
20         Standard (PKCS #1) [10], corresponding to the public key of
21         the SS. The 74, 106, 140, 204, and 270 byte key encoding
22         lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
23         and 2048 public moduli respectively. This is a zero-length
24         string if the BS does not retain the public key."
25     ::= { wmanIfBsPkmAuthEntry 2 }
26
27 wmanIfBsPkmAuthSsKeySequenceNumber OBJECT-TYPE
28     SYNTAX      Integer32 (0..15)
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32         "The value of this object is the most recent authorization
33         key sequence number for this SS."
34     ::= { wmanIfBsPkmAuthEntry 3 }
35
36 wmanIfBsPkmAuthSsExpiresOld OBJECT-TYPE
37     SYNTAX      DateAndTime
38     MAX-ACCESS  read-only
39     STATUS      current
40     DESCRIPTION
41         "The value of this object is the actual clock time for
42         expiration of the immediate predecessor of the most recent
43         authorization key for this FSM. If this FSM has only one
44         authorization key, then the value is the time of activation
45         of this FSM."
46     ::= { wmanIfBsPkmAuthEntry 4 }
47
48 wmanIfBsPkmAuthSsExpiresNew OBJECT-TYPE
49     SYNTAX      DateAndTime
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The value of this object is the actual clock time for
54         expiration of the most recent authorization key for this

```

```

1         FSM"
2         ::= { wmanIfBsPkmAuthEntry 5 }
3
4     wmanIfBsPkmAuthSsLifetime OBJECT-TYPE
5         SYNTAX      Integer32 (86400..6048000)
6         UNITS        "seconds"
7         MAX-ACCESS  read-write
8         STATUS      current
9         DESCRIPTION
10            "The vaue of this object is the lifetime, in seconds, the
11            BS assigns to an authorization key for this SS."
12         REFERENCE
13            "IEEE 802.16 standard; Table 270"
14         DEFVAL      { 604800 }
15         ::= { wmanIfBsPkmAuthEntry 6 }
16
17     wmanIfBsPkmAuthSsGraceTime OBJECT-TYPE
18         SYNTAX      Integer32 (300..3024000)
19         UNITS        "seconds"
20         MAX-ACCESS  read-only
21         STATUS      current
22         DESCRIPTION
23            "The value of this object is the grace time for the
24            authorization key in seconds. The SS is expected to start
25            trying to get a new authorization key beginning
26            AuthGraceTime seconds before the authorization key actually
27            expires."
28         REFERENCE
29            "IEEE 802.16 standard; Table 270"
30         DEFVAL      { 600 }
31         ::= { wmanIfBsPkmAuthEntry 7 }
32
33     wmanIfBsPkmAuthSsReset OBJECT-TYPE
34         SYNTAX      INTEGER { noResetRequested(1),
35                             invalidateAuth(2),
36                             sendAuthInvalid(3),
37                             invalidateTeks(4) }
38         MAX-ACCESS  read-write
39         STATUS      current
40         DESCRIPTION
41            "Setting this object to invalidateAuth(2) causes the BS to
42            invalidate the current SS authorization key(s), but not to
43            transmit an Authorization Invalid message nor to invalidate
44            unicast TEKS. Setting this object to sendAuthInvalid(3)
45            causes the BS to invalidate the current SS authorization
46            key(s), and to transmit an Authorization Invalid message to
47            the SS, but not to invalidate unicast TEKS. Setting this
48            object to invalidateTeks(4) causes the BS to invalidate the
49            current SS authorization key(s), to transmit an
50            Authorization Invalid message to the SS, and to
51            invalidate all unicast TEKS associated with this SS
52            authorization. Reading this object returns the
53            most-recently-set value of this object, or returns
54            noResetRequested(1) if the object has not been set since

```

```

1         the last BS reboot."
2     ::= { wmanIfBsPkmAuthEntry 8 }
3
4     wmanIfBsPkmAuthSsInfos OBJECT-TYPE
5         SYNTAX      Counter64
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "The value of this object is the count of times the BS has
10            received an Authentication Information message from this
11            SS."
12     ::= { wmanIfBsPkmAuthEntry 9 }
13
14     wmanIfBsPkmAuthSsRequests OBJECT-TYPE
15         SYNTAX      Counter64
16         MAX-ACCESS  read-only
17         STATUS      current
18         DESCRIPTION
19             "The value of this object is the count of times the BS has
20            received an Authorization Request message from this SS."
21     ::= { wmanIfBsPkmAuthEntry 10 }
22
23     wmanIfBsPkmAuthSsReplies OBJECT-TYPE
24         SYNTAX      Counter64
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "The value of this object is the count of times the BS has
29            transmitted an Authorization Reply message to this SS."
30     ::= { wmanIfBsPkmAuthEntry 11 }
31
32     wmanIfBsPkmAuthSsRejects 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            transmitted an Authorization Reject message to this SS."
39     ::= { wmanIfBsPkmAuthEntry 12 }
40
41     wmanIfBsPkmAuthSsInvalids OBJECT-TYPE
42         SYNTAX      Counter64
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            transmitted an Authorization Invalid message to this SS."
48     ::= { wmanIfBsPkmAuthEntry 13 }
49
50     wmanIfBsPkmAuthRejectErrorCode OBJECT-TYPE
51         SYNTAX      INTEGER {none(1),
52                        unknown(2),
53                        unauthorizedSs(3),
54                        unauthorizedSaid(4),

```

```

1             permanentAuthorizationFailure(8),
2             timeOfDayNotAcquired(11)}
3     MAX-ACCESS    read-only
4     STATUS        current
5     DESCRIPTION
6         "The value of this object is the enumerated description of
7         the Error-Code in most recent Authorization Reject message
8         transmitted to the SS. This has value unknown(2) if the
9         last Error-Code value was 0, and none(1) if no
10        Authorization Reject message has been transmitted to the
11        SS."
12    ::= { wmanIfBSPkmAuthEntry 14 }
13
14    wmanIfBSPkmAuthRejectErrorString OBJECT-TYPE
15        SYNTAX      SnmpAdminString (SIZE (0..128))
16        MAX-ACCESS  read-only
17        STATUS      current
18        DESCRIPTION
19            "The value of this object is the Display-String in most
20            recent Authorization Reject message transmitted to the SS.
21            This is a zero length string if no Authorization Reject
22            message has been transmitted to the SS."
23        ::= { wmanIfBSPkmAuthEntry 15 }
24
25    wmanIfBSPkmAuthInvalidErrorCode OBJECT-TYPE
26        SYNTAX      INTEGER {none(1),
27                    unknown(2),
28                    unauthorizedSs(3),
29                    unsolicited(5),
30                    invalidKeySequence(6),
31                    keyRequestAuthenticationFailure(7)}
32        MAX-ACCESS  read-only
33        STATUS      current
34        DESCRIPTION
35            "The value of this object is the enumerated description of
36            the Error-Code in most recent Authorization Invalid message
37            transmitted to the SS. This has value unknown(2) if the
38            last Error-Code value was 0, and none(1) if no
39            Authorization Invalid message has been transmitted to the
40            SS."
41        ::= { wmanIfBSPkmAuthEntry 16 }
42
43    wmanIfBSPkmAuthInvalidErrorString OBJECT-TYPE
44        SYNTAX      SnmpAdminString (SIZE (0..128))
45        MAX-ACCESS  read-only
46        STATUS      current
47        DESCRIPTION
48            "The value of this object is the Display-String in most
49            recent Authorization Invalid message transmitted to the SS.
50            This is a zero length string if no Authorization Invalid
51            message has been transmitted to the SS."
52        ::= { wmanIfBSPkmAuthEntry 17 }
53
54    wmanIfBSPkmAuthPrimarySAId OBJECT-TYPE

```

```

1          SYNTAX      Integer32 (0..16383)
2          MAX-ACCESS  read-only
3          STATUS      current
4          DESCRIPTION
5              "The value of this object is the Primary Security
6              Association identifier."
7          ::= { wmanIfBsPkmAuthEntry 18 }
8
9  wmanIfBsPkmAuthBpkmSsCertValid OBJECT-TYPE
10         SYNTAX      INTEGER {unknown (0),
11                     validSsChained (1),
12                     validSsTrusted (2),
13                     invalidSsUntrusted (3),
14                     invalidCAUntrusted (4),
15                     invalidSsOther (5),
16                     invalidCAOther (6) }
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "Contains the reason why a SS's certificate is deemed valid
21             or invalid. Return unknown if the SS is running PKM mode.
22             ValidSsChained means the certificate is valid because it
23             chains to a valid certificate. ValidSsTrusted means the
24             certificate is valid because it has been provisioned to be
25             trusted. InvalidSsUntrusted means the certificate is
26             invalid because it has been provisioned to be untrusted.
27             InvalidCAUntrusted means the certificate is invalid
28             because it chains to an untrusted certificate.
29             InvalidSsOther and InvalidCAOther refer to errors in
30             parsing, validity periods, etc, which are attributable to
31             the SS certificate or its chain respectively."
32         ::= { wmanIfBsPkmAuthEntry 19 }
33
34  wmanIfBsPkmAuthBpkmSsCert OBJECT-TYPE
35         SYNTAX      OCTET STRING
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "The X509 SS Certificate sent as part of a PKM
40             Authorization Request."
41         ::= { wmanIfBsPkmAuthEntry 20 }
42
43  --
44  -- Table wmanIfBsPkmTEKTable
45  --
46  wmanIfBsPkmTEKTable OBJECT-TYPE
47         SYNTAX      SEQUENCE OF wmanIfBsPkmTEKEntry
48         MAX-ACCESS  not-accessible
49         STATUS      current
50         DESCRIPTION
51             "This table describes the attributes of each Traffic
52             Encryption Key (TEK) association. The BS maintains one TEK
53             association per SAID on each BS wireless interface."
54         ::= { wmanIfBsPkmObjects 3 }

```

```

1
2  wmanIfBsPkmTEKEntry OBJECT-TYPE
3      SYNTAX      WmanIfBsPkmTEKEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "Each entry contains objects describing attributes of one
8          TEK association on a particular BS wireless interface. The
9          BS MUST create one entry per SAID per wireless interface,
10         based on the receipt of a Key Request message, and MUST not
11         delete the entry before the SS authorization for the SAID
12         permanently expires."
13     INDEX      { ifIndex, wmanIfBsPkmTEKSAID }
14     ::= { wmanIfBsPkmTEKTable 1 }
15
16  wmanIfBsPkmTEKEntry ::= SEQUENCE {
17      wmanIfBsPkmTEKSAID          Integer32,
18      wmanIfBsPkmTEKSAType        INTEGER,
19      wmanIfBsPkmTEKDataEncryptAlg  INTEGER,
20      wmanIfBsPkmTEKDataAuthentAlg  INTEGER,
21      wmanIfBsPkmTEKEncryptAlg      INTEGER,
22      wmanIfBsPkmTEKLifetime        Integer32,
23      wmanIfBsPkmTEKGraceTime       Integer32,
24      wmanIfBsPkmTEKKeySequenceNumber Integer32,
25      wmanIfBsPkmTEKExpiresOld      DateAndTime,
26      wmanIfBsPkmTEKExpiresNew      DateAndTime,
27      wmanIfBsPkmTEKReset           TruthValue,
28      wmanIfBsPkmKeyRequests         Counter32,
29      wmanIfBsPkmKeyReplies          Counter32,
30      wmanIfBsPkmKeyRejects         Counter32,
31      wmanIfBsPkmTEKInvalids        Counter32,
32      wmanIfBsPkmKeyRejectErrorCode  INTEGER,
33      wmanIfBsPkmKeyRejectErrorString SnmpAdminString,
34      wmanIfBsPkmTEKInvalidErrorCode INTEGER,
35      wmanIfBsPkmTEKInvalidErrorString SnmpAdminString
36  }
37
38  wmanIfBsPkmTEKSAID OBJECT-TYPE
39      SYNTAX      Integer32 (0..16383)
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "The value of this object is the WiMAX Security Association
44          ID (SAID)."
```

```

45     ::= { wmanIfBsPkmTEKEntry 1 }
46
47  wmanIfBsPkmTEKSAType OBJECT-TYPE
48      SYNTAX      INTEGER {none(0),
49                          primary(1),
50                          static(2),
51                          dynamic(3) }
52      MAX-ACCESS  read-only
53      STATUS      current
54      DESCRIPTION

```

```

1           "The value of this object is the type of security
2           association. Dynamic does not apply to SSS running in PKM
3           mode."
4       ::= { wmanIfBsPkmTEKEntry 2 }
5
6   wmanIfBsPkmTEKDataEncryptAlg OBJECT-TYPE
7       SYNTAX      INTEGER { none(0),
8                   des56CbcMode(1) }
9       MAX-ACCESS  read-only
10      STATUS      current
11      DESCRIPTION
12          "The value of this object is the data encryption algorithm
13          being utilized."
14      REFERENCE
15          "IEEE 802.16 standard; Table 301"
16      ::= { wmanIfBsPkmTEKEntry 3 }
17
18   wmanIfBsPkmTEKDataAuthentAlg OBJECT-TYPE
19      SYNTAX      INTEGER { none(0) }
20      MAX-ACCESS  read-only
21      STATUS      current
22      DESCRIPTION
23          "The value of this object is the data authentication
24          algorithm being utilized."
25      REFERENCE
26          "IEEE 802.16 standard; Table 302"
27      ::= { wmanIfBsPkmTEKEntry 4 }
28
29   wmanIfBsPkmTEKEncryptAlg OBJECT-TYPE
30      SYNTAX      INTEGER { tripleDES(0),
31                   rsa1024(1) }
32      MAX-ACCESS  read-only
33      STATUS      current
34      DESCRIPTION
35          "The value of this object is the TEK key encryption
36          algorithm being utilized."
37      REFERENCE
38          "IEEE 802.16 standard; Table 303"
39      ::= { wmanIfBsPkmTEKEntry 5 }
40
41   wmanIfBsPkmTEKLifetime OBJECT-TYPE
42      SYNTAX      Integer32 (1800..604800)
43      UNITS       "seconds"
44      MAX-ACCESS  read-write
45      STATUS      current
46      DESCRIPTION
47          "The value of this object is the lifetime, in seconds, the
48          BS assigns to keys for this TEK association."
49      REFERENCE
50          "IEEE 802.16 standard; Table 270"
51      DEFVAL     { 43200 }
52      ::= { wmanIfBsPkmTEKEntry 6 }
53
54   wmanIfBsPkmTEKGraceTime OBJECT-TYPE

```



```

1      SYNTAX      Integer32 (300..302399)
2      UNITS       "seconds"
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The value of this object is the grace time for the TEK in
7          seconds. The SS is expected to start trying to acquire a
8          new TEK beginning TEK GraceTime seconds before the TEK
9          actually expires."
10     REFERENCE
11         "IEEE 802.16 standard; Table 270"
12     DEFVAL      { 3600 }
13     ::= { wmanIfBsPkmTEKEntry 7 }
14
15     wmanIfBsPkmTEKKeySequenceNumber OBJECT-TYPE
16         SYNTAX      Integer32 (0..15)
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "The value of this object is the most recent TEK key
21             sequence number for this SAID."
22         ::= { wmanIfBsPkmTEKEntry 8 }
23
24     wmanIfBsPkmTEKExpiresOld OBJECT-TYPE
25         SYNTAX      DateAndTime
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "The value of this object is the actual clock time for
30             expiration of the immediate predecessor of the most recent
31             TEK for this FSM. If this FSM has only one TEK, then the
32             value is the time of activation of this FSM."
33         ::= { wmanIfBsPkmTEKEntry 9 }
34
35     wmanIfBsPkmTEKExpiresNew OBJECT-TYPE
36         SYNTAX      DateAndTime
37         MAX-ACCESS  read-only
38         STATUS      current
39         DESCRIPTION
40             "The value of this object is the actual clock time for
41             expiration of the most recent TEK for this FSM."
42         ::= { wmanIfBsPkmTEKEntry 10 }
43
44     wmanIfBsPkmTEKReset OBJECT-TYPE
45         SYNTAX      TruthValue
46         MAX-ACCESS  read-write
47         STATUS      current
48         DESCRIPTION
49             "Setting this object to TRUE causes the BS to invalidate
50             the current active TEK(s) (plural due to key transition
51             periods), and to generate a new TEK for the associated
52             SAID; the BS MAY also generate an unsolicited TEK Invalid
53             message, to optimize the TEK synchronization between the BS
54             and the SS. Reading this object always returns FALSE."

```

```

1      ::= { wmanIfBsPkmTEKEntry 11 }
2
3      wmanIfBsPkmKeyRequests OBJECT-TYPE
4          SYNTAX      Counter32
5          MAX-ACCESS  read-only
6          STATUS      current
7          DESCRIPTION
8              "The value of this object is the count of times the BS has
9              received a Key Request message."
10         ::= { wmanIfBsPkmTEKEntry 12 }
11
12         wmanIfBsPkmKeyReplies OBJECT-TYPE
13             SYNTAX      Counter32
14             MAX-ACCESS  read-only
15             STATUS      current
16             DESCRIPTION
17                 "The value of this object is the count of times the BS has
18                 transmitted a Key Reply message."
19         ::= { wmanIfBsPkmTEKEntry 13 }
20
21         wmanIfBsPkmKeyRejects OBJECT-TYPE
22             SYNTAX      Counter32
23             MAX-ACCESS  read-only
24             STATUS      current
25             DESCRIPTION
26                 "The value of this object is the count of times the BS has
27                 transmitted a Key Reject message."
28         ::= { wmanIfBsPkmTEKEntry 14 }
29
30         wmanIfBsPkmTEKInvalids OBJECT-TYPE
31             SYNTAX      Counter32
32             MAX-ACCESS  read-only
33             STATUS      current
34             DESCRIPTION
35                 "The value of this object is the count of times the BS has
36                 transmitted a TEK Invalid message."
37         ::= { wmanIfBsPkmTEKEntry 15 }
38
39         wmanIfBsPkmKeyRejectErrorCode OBJECT-TYPE
40             SYNTAX      INTEGER {none(1),
41                             unknown(2),
42                             unauthorizedSaid(4)}
43             MAX-ACCESS  read-only
44             STATUS      current
45             DESCRIPTION
46                 "The value of this object is the enumerated; description of
47                 the Error-Code in the most recent Key Reject message sent
48                 in response to a Key Request for this SAID. This has value
49                 unknown(2) if the last Error-Code value was 0, and none(1)
50                 if no Key Reject message has been received since reboot."
51         ::= { wmanIfBsPkmTEKEntry 16 }
52
53         wmanIfBsPkmKeyRejectErrorString OBJECT-TYPE
54             SYNTAX      SnmpAdminString (SIZE (0..128))

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the Display-String in the most
5          recent Key Reject message sent in response to a Key Request
6          for this SAID. This is a zero length string if no Key
7          Reject message has been received since reboot."
8      ::= { wmanIfBsPkmTEKEntry 17 }
9
10     wmanIfBsPkmTEKInvalidErrorCode OBJECT-TYPE
11         SYNTAX      INTEGER {none(1),
12                        unknown(2),
13                        invalidKeySequence(6)}
14         MAX-ACCESS  read-only
15         STATUS      current
16         DESCRIPTION
17             "The value of this object is the enumerated description of
18             the Error-Code in the most recent TEK Invalid message sent
19             in association with this SAID. This has value unknown(2)
20             if the last Error-Code value was 0, and none(1) if no TEK
21             Invalid message has been received since reboot."
22         ::= { wmanIfBsPkmTEKEntry 18 }
23
24     wmanIfBsPkmTEKInvalidErrorString OBJECT-TYPE
25         SYNTAX      SnmpAdminString (SIZE (0..128))
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "The value of this object is the Display-String in the most
30             recent TEK Invalid message sent in association with this
31             SAID. This is a zero length string if no TEK Invalid
32             message has been received since reboot."
33         ::= { wmanIfBsPkmTEKEntry 19 }
34
35     --
36     -- Base station Notification Group
37     -- wmanIfBsNotificationObjects contains the BS SNMP Trap objects
38     --
39     wmanIfBsNotification OBJECT IDENTIFIER ::= { wmanIfBsObjects 5 }
40     wmanIfBsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIfBsNotification 1 }
41     wmanIfBsTrapControl OBJECT IDENTIFIER ::= { wmanIfBsNotification 2 }
42
43     wmanIfBsTrapControlRegister OBJECT-TYPE
44         SYNTAX      BITS { wmanSwUpgradeInit           (0),
45                            wmanSwUpgradeFail          (1),
46                            wmanSwUpgradeSuccess        (2),
47                            wmanSwConfigurationChanged (3),
48                            wmanPswdChangeFail         (4),
49                            wmanPowerSupplyFail        (5),
50                            wmanTODFail                 (6),
51                            wmanBsInitRegReqSuccess    (7),
52                            wmanBsInitRegReqFail       (8),
53                            wmanBsDynServReqFail      (9),
54                            wmanBsDynServRspFail      (10),

```

```

1           wmanBsdynServAckFail           (11),
2           wmanBsBPKMFail                 (12),
3           wmanBsSsOnOffNotification      (13)
4       }
5   MAX-ACCESS read-write
6   STATUS      current
7   DESCRIPTION
8       "The object is used to enable Base Station traps. From left
9       to right, the set bit indicates the corresponding Base
10      Station trap is enabled."
11      ::= { wmanIfBsTrapControl 1 }
12
13  wmanIfBsSsStatusValue OBJECT-TYPE
14      SYNTAX      INTEGER {active(1),
15                      offline(2)}
16      MAX-ACCESS read-write
17      STATUS      current
18      DESCRIPTION
19          "This type defines the status of each subscriber station"
20      ::= { wmanIfBsTrapControl 2 }
21
22  wmanBsInitRegReqSuccessTrap NOTIFICATION-TYPE
23      OBJECTS      { wmanIfBsSsMacAddress }
24      STATUS      current
25      DESCRIPTION
26          "An event to report the success of a registration request
27          from Subscriber Station happened during the Subscriber
28          Station initialization process and detected in the Bs side"
29      ::= { wmanIfBsTrapDefinitions 1 }
30
31  wmanBsInitRegReqFailTrap NOTIFICATION-TYPE
32      OBJECTS      { wmanIfBsSsMacAddress }
33      STATUS      current
34      DESCRIPTION
35          "An event to report the failure of a registration request
36          from Subscriber Station happened during the Subscriber
37          Station initialization process and detected in the Bs side"
38      ::= { wmanIfBsTrapDefinitions 2 }
39
40  wmanBsdynServReqFailTrap NOTIFICATION-TYPE
41      OBJECTS      { wmanIfBsSsMacAddress }
42      STATUS      current
43      DESCRIPTION
44          "An event to report the failure of a dynamic service
45          request happened during the dynamic services process
46          and detected in the Bs side"
47      ::= { wmanIfBsTrapDefinitions 3 }
48
49  wmanBsdynServRspFailTrap NOTIFICATION-TYPE
50      OBJECTS      { wmanIfBsSsMacAddress }
51      STATUS      current
52      DESCRIPTION
53          "An event to report the failure of a dynamic service
54          response happened during the dynamic services process

```

```

1         and detected in the Bs side."
2     ::= { wmanIfBsTrapDefinitions 4 }
3
4     wmanBsDynServAckFailTrap    NOTIFICATION-TYPE
5         OBJECTS    { wmanIfBsSsMacAddress }
6         STATUS     current
7         DESCRIPTION
8             "An event to report the failure of a dynamic service
9             acknowledgement happened during the dynamic services
10            process and detected in the Bs side."
11     ::= { wmanIfBsTrapDefinitions 5 }
12
13     wmanBsBPKMFailTrap    NOTIFICATION-TYPE
14         OBJECTS    { wmanIfBsSsMacAddress }
15         STATUS     current
16         DESCRIPTION
17             "An event to report the failure of a BPKM operation
18             which is detected in the Bs side."
19     ::= { wmanIfBsTrapDefinitions 6 }
20
21     wmanBsSsOnOffNotificationTrap    NOTIFICATION-TYPE
22         OBJECTS    { wmanIfBsSsMacAddress,
23                    wmanIfBsSsStatusValue
24                    }
25         STATUS     current
26         DESCRIPTION
27             "Possible values for wmanIfBsSsStatusValue are: active(1)
28             and offline(2).
29             Based on this notification the NMS will issue an alarm
30             with Critical severity if the Subscriber Station with MAC
31             address wmanIfBsSsMacAddress becomes offline, and an
32             alarm with a Clear severity is generated when a
33             Subscriber Station comes active."
34     ::= { wmanIfBsTrapDefinitions 7 }
35
36     --
37     -- SS object group - containing tables and objects to be implemented in
38     -- the Subscriber station
39     --
40     -- wmanIfSsSystem contain the Subscriber Station System objects
41     wmanIfSsSystem OBJECT IDENTIFIER ::= { wmanIfSsObjects 1 }
42
43     wmanIfSsConfigFileEncodingTable OBJECT-TYPE
44         SYNTAX     SEQUENCE OF wmanIfSsConfigFileEncodingEntry
45         MAX-ACCESS not-accessible
46         STATUS     current
47         DESCRIPTION
48             "This table contains configuration file encoding
49             information of the SS."
50         REFERENCE
51             "Section 11.2 in IEEE 802.16REVD/D3-2004"
52     ::= { wmanIfSsSystem 1 }
53
54     wmanIfSsConfigFileEncodingEntry OBJECT-TYPE

```

```

1      SYNTAX      WmanIfSsConfigFileEncodingEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table provides one row for each SS, and is indexed
6          by wmanIfSsIdIndex."
7      INDEX { wmanIfSsIdIndex }
8      ::= { wmanIfSsConfigFileEncodingTable 1 }
9
10     WmanIfSsConfigFileEncodingEntry ::= SEQUENCE {
11         wmanIfSsIdIndex                Unsigned32,
12         wmanIfSsMicConfigSetting        OCTET STRING,
13         wmanIfSsVendorId                OCTET STRING,
14         wmanIfSsHwId                    OCTET STRING,
15         wmanIfSsSwVersion                OCTET STRING,
16         wmanIfSsUpgradeFileName         OCTET STRING,
17         wmanIfSsSwUpgradeTftpServer     InetAddress,
18         wmanIfSsTftpServerTimeStamp     DateAndTime
19     }
20
21     wmanIfSsIdIndex OBJECT-TYPE
22         SYNTAX      Unsigned32 (1 .. 4294967295)
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "wmanIfSsIdIndex identifies the SS that is registered."
27         REFERENCE
28             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
29         ::= { wmanIfSsConfigFileEncodingEntry 1 }
30
31     wmanIfSsMicConfigSetting OBJECT-TYPE
32         SYNTAX      OCTET STRING (SIZE(16))
33         MAX-ACCESS  read-only
34         STATUS      current
35         DESCRIPTION
36             "The value field contains the SS MIC code. This is used
37             to detect unauthorized modification or corruption of
38             the configuration file."
39         ::= { wmanIfSsConfigFileEncodingEntry 2 }
40
41     wmanIfSsVendorId OBJECT-TYPE
42         SYNTAX      OCTET STRING (SIZE(3))
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "This value identifies the managed SS vendor to which the
47             software upgrade is to be applied."
48         ::= { wmanIfSsConfigFileEncodingEntry 3 }
49
50     wmanIfSsHwId OBJECT-TYPE
51         SYNTAX      OCTET STRING
52         MAX-ACCESS  read-only
53         STATUS      current
54         DESCRIPTION

```

```

1           "This value identifies the hardware version to which the
2           software upgrade is to be applied."
3           ::= { wmanIfSsConfigFileEncodingEntry 4 }
4
5 wmanIfSsSwVersion OBJECT-TYPE
6     SYNTAX      OCTET STRING
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10            "The value field contains the SS MIC code. This is used
11            to detect unauthorized modification or corruption of
12            the configuration file."
13            ::= { wmanIfSsConfigFileEncodingEntry 5 }
14
15 wmanIfSsUpgradeFileName OBJECT-TYPE
16     SYNTAX      OCTET STRING
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20            "The filename is a fully qualified directory path
21            name which is in a format appropriate to the server."
22            ::= { wmanIfSsConfigFileEncodingEntry 6 }
23
24 wmanIfSsSwUpgradeTftpServer OBJECT-TYPE
25     SYNTAX      InetAddress
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29            "This object is the IP address of the TFTP server on
30            which the software upgrade file for the SS resides."
31            ::= { wmanIfSsConfigFileEncodingEntry 7 }
32
33 wmanIfSsTftpServerTimeStamp OBJECT-TYPE
34     SYNTAX      DateAndTime
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38            "This is the sending time of the configuration file in
39            seconds. The definition of time is as in IETF RFC 868."
40            ::= { wmanIfSsConfigFileEncodingEntry 8 }
41
42 --
43 -- wmanIfSsCps contain the Base Station Common Part Sublayer objects
44 wmanIfSsCps OBJECT IDENTIFIER ::= { wmanIfSsObjects 2 }
45
46 --
47 -- wmanIfSsConfigurationTable contains global parameters common
48 -- in BS and SS
49 wmanIfSsConfigurationTable OBJECT-TYPE
50     SYNTAX      SEQUENCE OF wmanIfSsConfigurationEntry
51     MAX-ACCESS  not-accessible
52     STATUS      current
53     DESCRIPTION
54            "This table provides one row for each SS that contains

```

```

1         the system parameters as defined in section 10.1 of [3].
2 ::= { wmanIfSsCps 1 }
3
4 wmanIfSsConfigurationEntry OBJECT-TYPE
5     SYNTAX      WmanIfSsConfigurationEntry
6     MAX-ACCESS  not-accessible
7     STATUS      current
8     DESCRIPTION
9         "This table is indexed by wmanIfCmnSsIdIndex."
10    INDEX { ifIndex }
11    ::= { wmanIfSsConfigurationTable 1 }
12
13 wmanIfSsConfigurationEntry ::= SEQUENCE {
14     wmanIfSsId                Unsigned32,
15     wmanIfSsLostDLMapInterval INTEGER,
16     wmanIfSsLostULMapInterval INTEGER,
17     wmanIfSsContentionRangRetries INTEGER,
18     wmanIfSsRequestRetries    INTEGER,
19     wmanIfSsRegRequestRetries INTEGER,
20     wmanIfSsTftpBackoffStart  INTEGER,
21     wmanIfSsTftpBackoffEnd    INTEGER,
22     wmanIfSsTftpRequestRetries INTEGER,
23     wmanIfSsTftpDownloadRetries INTEGER,
24     wmanIfSsTftpWait          INTEGER,
25     wmanIfSsToDRetries        INTEGER,
26     wmanIfSsToDRetryPeriod    INTEGER,
27     wmanIfSsT1Timeout         INTEGER,
28     wmanIfSsT2Timeout         INTEGER,
29     wmanIfSsT3Timeout         INTEGER,
30     wmanIfSsT4Timeout         INTEGER,
31     wmanIfSsT6Timeout         INTEGER,
32     wmanIfSsT12Timeout        INTEGER,
33     wmanIfSsT14Timeout        INTEGER,
34     wmanIfSsT16Timeout        INTEGER,
35     wmanIfSsT18Timeout        INTEGER,
36     wmanIfSsT19Timeout        INTEGER,
37     wmanIfSsT20Timeout        INTEGER,
38     wmanIfSsT21Timeout        INTEGER,
39     wmanIfSsSBCRequestRetries INTEGER,
40     wmanIfSsTftpCpltRetries   INTEGER,
41     wmanIfSsT26Timeout        INTEGER,
42     wmanIfSsDLManagProcTime   INTEGER,
43     wmanIfSsConfigurationRowStatus RowStatus
44 }
45
46 wmanIfSsId OBJECT-TYPE
47     SYNTAX      Unsigned32 (1 .. 4294967295)
48     MAX-ACCESS  read-only
49     STATUS      current
50     DESCRIPTION
51         "wmanIfSsId is the index to
52         wmanIfSsConfigurationTable."
53     REFERENCE
54         "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"

```



```

1      ::= { wmanIfSsConfigurationEntry 1 }
2
3      wmanIfSsLostDLMapInterval OBJECT-TYPE
4          SYNTAX      INTEGER(0..600)
5          UNITS       "milliseconds"
6          MAX-ACCESS  read-write
7          STATUS      current
8          DESCRIPTION
9              "Time since last received DL-MAP message before downlink
10             synchronization is considered lost in ms."
11      ::= { wmanIfSsConfigurationEntry 2 }
12
13      wmanIfSsLostULMapInterval OBJECT-TYPE
14          SYNTAX      INTEGER(0..600)
15          UNITS       "milliseconds"
16          MAX-ACCESS  read-write
17          STATUS      current
18          DESCRIPTION
19              "Time since last received UL-MAP message before downlink
20             synchronization is considered lost in ms."
21      ::= { wmanIfSsConfigurationEntry 3 }
22
23      wmanIfSsContentionRangRetries OBJECT-TYPE
24          SYNTAX      INTEGER(16..65535)
25          MAX-ACCESS  read-write
26          STATUS      current
27          DESCRIPTION
28              "Number of retries on contention Ranging Requests."
29      ::= { wmanIfSsConfigurationEntry 4 }
30
31      wmanIfSsRequestRetries OBJECT-TYPE
32          SYNTAX      INTEGER(16..65535)
33          MAX-ACCESS  read-write
34          STATUS      current
35          DESCRIPTION
36              "Number of retries on bandwidth allocation requests."
37      ::= { wmanIfSsConfigurationEntry 5 }
38
39      wmanIfSsRegRequestRetries OBJECT-TYPE
40          SYNTAX      INTEGER(3..65535)
41          MAX-ACCESS  read-write
42          STATUS      current
43          DESCRIPTION
44              "Number of retries on registration requests."
45      ::= { wmanIfSsConfigurationEntry 6 }
46
47      wmanIfSsTftpBackoffStart OBJECT-TYPE
48          SYNTAX      INTEGER(1..65535)
49          UNITS       "seconds"
50          MAX-ACCESS  read-write
51          STATUS      current
52          DESCRIPTION
53              "Initial value for TFTP backoff in s."
54      ::= { wmanIfSsConfigurationEntry 7 }

```

```

1
2  wmanIfsTftpBackoffEnd OBJECT-TYPE
3      SYNTAX      INTEGER(16..65535)
4      UNITS       "seconds"
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8          "Last value for TFTP backoff in s."
9      ::= { wmanIfsConfigurationEntry 8 }
10
11 wmanIfsTftpRequestRetries OBJECT-TYPE
12     SYNTAX      INTEGER(16..65535)
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Number of retries on TFTP request."
17     ::= { wmanIfsConfigurationEntry 9 }
18
19 wmanIfsTftpDownloadRetries OBJECT-TYPE
20     SYNTAX      INTEGER(3..65535)
21     MAX-ACCESS  read-write
22     STATUS      current
23     DESCRIPTION
24         "Number of retries on entire TFTP downloads."
25     ::= { wmanIfsConfigurationEntry 10 }
26
27 wmanIfsTftpWait OBJECT-TYPE
28     SYNTAX      INTEGER(2..65535)
29     UNITS       "minutes"
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "The duration between two consecutive TFTP retries in min."
34     ::= { wmanIfsConfigurationEntry 11 }
35
36 wmanIfsToDRetries OBJECT-TYPE
37     SYNTAX      INTEGER(3..65535)
38     MAX-ACCESS  read-write
39     STATUS      current
40     DESCRIPTION
41         "Number of Retries per Time of Day Retry Period."
42     ::= { wmanIfsConfigurationEntry 12 }
43
44 wmanIfsToDRetryPeriod OBJECT-TYPE
45     SYNTAX      INTEGER(5..65535)
46     MAX-ACCESS  read-write
47     STATUS      current
48     DESCRIPTION
49         "Time of Day Retry Period."
50     ::= { wmanIfsConfigurationEntry 13 }
51
52 wmanIfsT1Timeout OBJECT-TYPE
53     SYNTAX      INTEGER(0..65535)
54     UNITS       "milliseconds"

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "Wait for DCD timeout in ms."
5      ::= { wmanIfsConfigurationEntry 14 }
6
7      wmanIfsSt2Timeout OBJECT-TYPE
8          SYNTAX      INTEGER(0..65535)
9          UNITS        "milliseconds"
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "Wait for broadcast ranging timeout in ms."
14         ::= { wmanIfsConfigurationEntry 15 }
15
16         wmanIfsSt3Timeout OBJECT-TYPE
17             SYNTAX      INTEGER(0..200)
18             UNITS        "milliseconds"
19             MAX-ACCESS  read-write
20             STATUS      current
21             DESCRIPTION
22                 "Ranging Response reception timeout following the
23                 transmission of a Ranging Request in ms."
24             ::= { wmanIfsConfigurationEntry 16 }
25
26         wmanIfsSt4Timeout OBJECT-TYPE
27             SYNTAX      INTEGER(30..35)
28             UNITS        "seconds"
29             MAX-ACCESS  read-write
30             STATUS      current
31             DESCRIPTION
32                 "Wait for unicast ranging opportunity. If the pending until
33                 complete field was used earlier by this SS, then the value
34                 of that field shall be added to this interval in s."
35             ::= { wmanIfsConfigurationEntry 17 }
36
37         wmanIfsSt6Timeout OBJECT-TYPE
38             SYNTAX      INTEGER(0..3000)
39             UNITS        "milliseconds"
40             MAX-ACCESS  read-write
41             STATUS      current
42             DESCRIPTION
43                 "Wait for registration response in ms."
44             ::= { wmanIfsConfigurationEntry 18 }
45
46         wmanIfsSt12Timeout OBJECT-TYPE
47             SYNTAX      INTEGER
48             UNITS        "milliseconds"
49             MAX-ACCESS  read-write
50             STATUS      current
51             DESCRIPTION
52                 "Wait for UCD descriptor in ms."
53             ::= { wmanIfsConfigurationEntry 19 }
54

```

```

1  wmanIfsSt14Timeout OBJECT-TYPE
2      SYNTAX      INTEGER(0..200)
3      UNITS       "milliseconds"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "wait for DSX-RVD Timeout in ms."
8      ::= { wmanIfsConfigurationEntry 20 }
9
10 wmanIfsSt16Timeout OBJECT-TYPE
11     SYNTAX      INTEGER(10..65535)
12     UNITS       "milliseconds"
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "wait for bandwidth request grant in ms."
17     ::= { wmanIfsConfigurationEntry 21 }
18
19 wmanIfsSt18Timeout OBJECT-TYPE
20     SYNTAX      INTEGER(0..65535)
21     UNITS       "milliseconds"
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         "wait for SBC-RSP timeout in ms."
26     ::= { wmanIfsConfigurationEntry 22 }
27
28 wmanIfsSt19Timeout OBJECT-TYPE
29     SYNTAX      INTEGER(0..65535)
30     UNITS       "milliseconds"
31     MAX-ACCESS  read-write
32     STATUS      current
33     DESCRIPTION
34         "Time DL-channel remains unusable in ms."
35     ::= { wmanIfsConfigurationEntry 23 }
36
37 wmanIfsSt20Timeout OBJECT-TYPE
38     SYNTAX      INTEGER(0..65535)
39     UNITS       "milliseconds"
40     MAX-ACCESS  read-write
41     STATUS      current
42     DESCRIPTION
43         "Time SS searches for preambles on a given channel in ms."
44     ::= { wmanIfsConfigurationEntry 24 }
45
46 wmanIfsSt21Timeout OBJECT-TYPE
47     SYNTAX      INTEGER(0..10000)
48     UNITS       "milliseconds"
49     MAX-ACCESS  read-write
50     STATUS      current
51     DESCRIPTION
52         "Time SS searches for DL-MAP on a given channel in ms."
53     ::= { wmanIfsConfigurationEntry 25 }
54

```

```

1  wmanIfSsSBCRequestRetries OBJECT-TYPE
2      SYNTAX      INTEGER(3..16)
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "Number of retries on SBC Request."
7      ::= { wmanIfSsConfigurationEntry 26 }
8
9  wmanIfSsTftpCpltRetries OBJECT-TYPE
10     SYNTAX      INTEGER(3..16)
11     MAX-ACCESS  read-write
12     STATUS      current
13     DESCRIPTION
14         "Number of retries on TFTP-CPLT."
15     ::= { wmanIfSsConfigurationEntry 27 }
16
17  wmanIfSsT26Timeout OBJECT-TYPE
18     SYNTAX      INTEGER(10..200)
19     UNITS       "milliseconds"
20     MAX-ACCESS  read-write
21     STATUS      current
22     DESCRIPTION
23         "Wait for TFTP-RSP in ms."
24     ::= { wmanIfSsConfigurationEntry 28 }
25
26  wmanIfSsDLManagProcTime OBJECT-TYPE
27     SYNTAX      INTEGER(0..200)
28     UNITS       "micro seconds"
29     MAX-ACCESS  read-write
30     STATUS      current
31     DESCRIPTION
32         "Max. time between reception of Fast Power Control
33         management message and compliance to its instructions
34         by SS in us."
35     ::= { wmanIfSsConfigurationEntry 29 }
36
37  wmanIfSsConfigurationRowStatus OBJECT-TYPE
38     SYNTAX      RowStatus
39     MAX-ACCESS  read-create
40     STATUS      current
41     DESCRIPTION
42         "This object is used to create a new row or modify or
43         delete an existing row in this table.
44
45         If the implementator of this MIB has chosen not
46         to implement 'dynamic assignment' of profiles, this
47         object is not useful and should return noSuchName
48         upon SNMP request."
49     ::= { wmanIfSsConfigurationEntry 30 }
50
51  -- Subscriber station PKM group
52  -- wmanIfSsPkmObjects contain the Subscriber Station Privacy Sublayer
53  -- objects
54  wmanIfSsPkmObjects OBJECT IDENTIFIER ::= { wmanIfSsObjects 3 }

```

```

1
2
3  --
4  -- Table wmanIfSsPkmBaseTable
5  --
6  wmanIfSsPkmBaseTable OBJECT-TYPE
7      SYNTAX      SEQUENCE OF  wmanIfSsPkmBaseEntry
8      MAX-ACCESS  not-accessible
9      STATUS      current
10     DESCRIPTION
11         "This table describes the basic and authorization related
12         PKM attributes of each SS wireless interface."
13     ::= { wmanIfSsPkmObjects 1 }
14
15  wmanIfSsPkmBaseEntry OBJECT-TYPE
16      SYNTAX      wmanIfSsPkmBaseEntry
17      MAX-ACCESS  not-accessible
18      STATUS      current
19      DESCRIPTION
20         "Each entry contains objects describing attributes of one
21         SS wireless interface."
22      INDEX      { ifIndex }
23      ::= { wmanIfSsPkmBaseTable 1 }
24
25  wmanIfSsPkmBaseEntry ::= SEQUENCE {
26      wmanIfSsPkmPrivacyEnable      TruthValue,
27      wmanIfSsPkmPublicKey          OCTET STRING,
28      wmanIfSsPkmAuthState          INTEGER,
29      wmanIfSsPkmAuthKeySequenceNumber Integer32,
30      wmanIfSsPkmAuthExpiresOld    DateAndTime,
31      wmanIfSsPkmAuthExpiresNew    DateAndTime,
32      wmanIfSsPkmAuthReset         TruthValue,
33      wmanIfSsPkmAuthGraceTime     Integer32,
34      wmanIfSsPkmTEKGraceTime     Integer32,
35      wmanIfSsPkmAuthWaitTimeout   Integer32,
36      wmanIfSsPkmReauthWaitTimeout Integer32,
37      wmanIfSsPkmOpWaitTimeout     Integer32,
38      wmanIfSsPkmRekeyWaitTimeout  Integer32,
39      wmanIfSsPkmAuthRejectWaitTimeout Integer32,
40      wmanIfSsPkmAuthentInfos      Counter32,
41      wmanIfSsPkmAuthRequests      Counter32,
42      wmanIfSsPkmAuthReplies       Counter32,
43      wmanIfSsPkmAuthRejects       Counter32,
44      wmanIfSsPkmAuthInvalids      Counter32,
45      wmanIfSsPkmAuthRejectErrorCode INTEGER,
46      wmanIfSsPkmAuthRejectErrorString SnmpAdminString,
47      wmanIfSsPkmAuthInvalidErrorCode INTEGER,
48      wmanIfSsPkmAuthInvalidErrorString SnmpAdminString
49  }
50
51  wmanIfSsPkmPrivacyEnable OBJECT-TYPE
52      SYNTAX      TruthValue
53      MAX-ACCESS  read-only
54      STATUS      current

```

```

1      DESCRIPTION
2          "This object identifies whether this SS is provisioned to
3          run Baseline Privacy Plus."
4      ::= { wmanIfSsPkmBaseEntry 1 }
5
6      wmanIfSsPkmPublicKey OBJECT-TYPE
7          SYNTAX      OCTET STRING (SIZE (140))
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "The value of this object is a DER-encoded RSAPublicKey
12             ASN.1 type string, as defined in the RSA Encryption
13             Standard (PKCS#1) [10], corresponding to the public key of
14             the SS. The 74, 106, 140, 204, and 270 byte key encoding
15             lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
16             and 2048 public moduli respectively."
17         ::= { wmanIfSsPkmBaseEntry 2 }
18
19         wmanIfSsPkmAuthState OBJECT-TYPE
20             SYNTAX      INTEGER {start(1),
21                         authwait(2),
22                         authorized(3),
23                         reauthwait(4),
24                         authRejectwait(5),
25                         silent(6)}
26             MAX-ACCESS  read-only
27             STATUS      current
28             DESCRIPTION
29                 "The value of this object is the state of the SS
30                 authorization FSM. The start state indicates that FSM is
31                 in its initial state."
32             ::= { wmanIfSsPkmBaseEntry 3 }
33
34         wmanIfSsPkmAuthKeySequenceNumber OBJECT-TYPE
35             SYNTAX      Integer32 (0..15)
36             MAX-ACCESS  read-only
37             STATUS      current
38             DESCRIPTION
39                 "The value of this object is the most recent authorization
40                 key sequence number for this FSM."
41             ::= { wmanIfSsPkmBaseEntry 4 }
42
43         wmanIfSsPkmAuthExpiresOld OBJECT-TYPE
44             SYNTAX      DateAndTime
45             MAX-ACCESS  read-only
46             STATUS      current
47             DESCRIPTION
48                 "The value of this object is the actual clock time for
49                 expiration of the immediate predecessor of the most recent
50                 authorization key for this FSM. If this FSM has only one
51                 authorization key, then the value is the time of activation
52                 of this FSM."
53             ::= { wmanIfSsPkmBaseEntry 5 }
54

```

```

1  wmanIfSsPkmAuthExpiresNew OBJECT-TYPE
2      SYNTAX      DateAndTime
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The value of this object is the actual clock time for
7          expiration of the most recent authorization key for this
8          FSM."
9      ::= { wmanIfSsPkmBaseEntry 6 }
10
11 wmanIfSsPkmAuthReset OBJECT-TYPE
12     SYNTAX      TruthValue
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Setting this object to TRUE generates a Reauthorize event
17         in the authorization FSM. Reading this object always
18         returns FALSE."
19     ::= { wmanIfSsPkmBaseEntry 7 }
20
21 wmanIfSsPkmAuthGraceTime OBJECT-TYPE
22     SYNTAX      Integer32 (300..3024000)
23     UNITS       "seconds"
24     MAX-ACCESS  read-only
25     STATUS      current
26     DESCRIPTION
27         "The value of this object is the grace time for an
28         authorization key. A SS is expected to start trying to get
29         a new authorization key beginning AuthGraceTime seconds
30         before the authorization key actually expires."
31     REFERENCE
32         "IEEE 802.16 standard; Table 270"
33     DEFVAL      { 600 }
34     ::= { wmanIfSsPkmBaseEntry 8 }
35
36 wmanIfSsPkmTEKGraceTime OBJECT-TYPE
37     SYNTAX      Integer32 (300..3024000)
38     UNITS       "seconds"
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "The value of this object is the grace time for the TEK in
43         seconds. The SS is expected to start trying to acquire a
44         new TEK beginning TEK GraceTime seconds before the
45         expiration of the most recent TEK."
46     REFERENCE
47         "IEEE 802.16 standard; Table 270"
48     DEFVAL      { 3600 }
49     ::= { wmanIfSsPkmBaseEntry 9 }
50
51 wmanIfSsPkmAuthwaitTimeout OBJECT-TYPE
52     SYNTAX      Integer32 (2..30)
53     UNITS       "seconds"
54     MAX-ACCESS  read-only

```



```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object is the Authorize wait Timeout."
4      REFERENCE
5          "IEEE 802.16 standard; Table 270"
6      DEFVAL      { 10 }
7      ::= { wmanIfSsPkmBaseEntry 10 }
8
9      wmanIfSsPkmReauthwaitTimeout OBJECT-TYPE
10     SYNTAX      Integer32 (2..30)
11     UNITS       "seconds"
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "The value of this object is the Reauthorize wait Timeout
16         in seconds."
17     REFERENCE
18         "IEEE 802.16 standard; Table 270"
19     DEFVAL      { 10 }
20     ::= { wmanIfSsPkmBaseEntry 11 }
21
22     wmanIfSsPkmOpwaitTimeout OBJECT-TYPE
23     SYNTAX      Integer32 (1..10)
24     UNITS       "seconds"
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28         "The value of this object is the Operational wait Timeout
29         in seconds."
30     REFERENCE
31         "IEEE 802.16 standard; Table 270"
32     DEFVAL      { 1 }
33     ::= { wmanIfSsPkmBaseEntry 12 }
34
35     wmanIfSsPkmRekeywaitTimeout OBJECT-TYPE
36     SYNTAX      Integer32 (1..10)
37     UNITS       "seconds"
38     MAX-ACCESS  read-only
39     STATUS      current
40     DESCRIPTION
41         "The value of this object is the Rekey wait Timeout in
42         seconds."
43     REFERENCE
44         "IEEE 802.16 standard; Table 270"
45     DEFVAL      { 1 }
46     ::= { wmanIfSsPkmBaseEntry 13 }
47
48     wmanIfSsPkmAuthRejectwaitTimeout OBJECT-TYPE
49     SYNTAX      Integer32 (10..600)
50     UNITS       "seconds"
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "The value of this object is the Authorization Reject wait

```

```

1         Timeout in seconds."
2     REFERENCE
3         "IEEE 802.16 standard; Table 270"
4     DEFVAL      { 60 }
5     ::= { wmanIfSsPkmBaseEntry 14 }
6
7     wmanIfSsPkmAuthentInfos 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            transmitted an Authentication Information message."
14        ::= { wmanIfSsPkmBaseEntry 15 }
15
16        wmanIfSsPkmAuthRequests OBJECT-TYPE
17            SYNTAX      Counter32
18            MAX-ACCESS  read-only
19            STATUS      current
20            DESCRIPTION
21                "The value of this object is the count of times the SS has
22                transmitted an Authorization Request message."
23            ::= { wmanIfSsPkmBaseEntry 16 }
24
25        wmanIfSsPkmAuthReplies OBJECT-TYPE
26            SYNTAX      Counter32
27            MAX-ACCESS  read-only
28            STATUS      current
29            DESCRIPTION
30                "The value of this object is the count of times the SS has
31                received an Authorization Reply message."
32            ::= { wmanIfSsPkmBaseEntry 17 }
33
34        wmanIfSsPkmAuthRejects OBJECT-TYPE
35            SYNTAX      Counter32
36            MAX-ACCESS  read-only
37            STATUS      current
38            DESCRIPTION
39                "The value of this object is the count of times the SS has
40                received an Authorization Reject message."
41            ::= { wmanIfSsPkmBaseEntry 18 }
42
43        wmanIfSsPkmAuthInvalids 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 the SS has
49                received an Authorization Invalid message."
50            ::= { wmanIfSsPkmBaseEntry 19 }
51
52        wmanIfSsPkmAuthRejectErrorCode OBJECT-TYPE
53            SYNTAX      INTEGER {none(1),
54                        unknown(2),

```

```

1             unauthorizedSs(3),
2             unauthorizedSaid(4),
3             permanentAuthorizationFailure(8),
4             timeOfDayNotAcquired(11)}
5     MAX-ACCESS    read-only
6     STATUS        current
7     DESCRIPTION
8         "The value of this object is the enumerated description of
9         the Error-Code in most recent Authorization Reject message
10        received by the SS. This has value unknown(2)if the last
11        Error-Code value was 0, and none(1) if no Authorization
12        Reject message has been received since reboot."
13    ::= { wmanIfSsPkmBaseEntry 20 }
14
15    wmanIfSsPkmAuthRejectErrorString 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 Reject message received by the SS.
22            This is a zero length string if no Authorization Reject
23            message has been received since reboot."
24        ::= { wmanIfSsPkmBaseEntry 21 }
25
26    wmanIfSsPkmAuthInvalidErrorCode OBJECT-TYPE
27        SYNTAX      INTEGER {none(1),
28                        unknown(2),
29                        unauthorizedSs(3),
30                        unsolicited(5),
31                        invalidKeySequence(6),
32                        keyRequestAuthenticationFailure(7)}
33        MAX-ACCESS  read-only
34        STATUS      current
35        DESCRIPTION
36            "The value of this object is the enumerated description of
37            the Error-Code in most recent Authorization Invald message
38            received by the SS. This has value unknown(2) if the last
39            Error-Code value was 0, and none(1) if no Authorization
40            Invalid message has been received since reboot."
41        ::= { wmanIfSsPkmBaseEntry 22 }
42
43    wmanIfSsPkmAuthInvalidErrorString OBJECT-TYPE
44        SYNTAX      SnmpAdminString (SIZE (0..128))
45        MAX-ACCESS  read-only
46        STATUS      current
47        DESCRIPTION
48            "The value of this object is the Display-String in most
49            recent Authorization Invalid message received by the SS.
50            This is a zero length string if no Authorization Invalid
51            message has been received since reboot."
52        ::= { wmanIfSsPkmBaseEntry 23 }
53
54    --

```

```

1  -- Table wmanIfSsPkmTEKTable
2  --
3  wmanIfSsPkmTEKTable OBJECT-TYPE
4      SYNTAX      SEQUENCE OF WmanIfSsPkmTEKEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8          "This table describes the attributes of each SS Traffic
9          Encryption Key(TEK) association. The SS maintains (no more
10         than) one TEK association per SAID per SS wireless
11         interface."
12     ::= { wmanIfSsPkmObjects 2 }
13
14  wmanIfSsPkmTEKEntry OBJECT-TYPE
15      SYNTAX      WmanIfSsPkmTEKEntry
16      MAX-ACCESS  not-accessible
17      STATUS      current
18      DESCRIPTION
19          "Each entry contains objects describing the TEK association
20          attributes of one SAID. The SS MUST create one entry per
21          SAID, regardless of whether the SAID was obtained from a
22          Registration Response message, from an Authorization Reply
23          message, or from any dynamic SAID establishment
24          mechanisms."
25      INDEX       { ifIndex, wmanIfSsPkmTEKSAID }
26     ::= { wmanIfSsPkmTEKTable 1 }
27
28  wmanIfSsPkmTEKEntry ::= SEQUENCE {
29      wmanIfSsPkmTEKSAID                Integer32,
30      wmanIfSsPkmTEKSAType              INTEGER,
31      wmanIfSsPkmTEKDataEncryptAlg     INTEGER,
32      wmanIfSsPkmTEKDataAuthentAlg    INTEGER,
33      wmanIfSsPkmTEKEncryptAlg        INTEGER,
34      wmanIfSsPkmTEKState               INTEGER,
35      wmanIfSsPkmTEKKeySequenceNumber Integer32,
36      wmanIfSsPkmTEKExpiresOld         DateAndTime,
37      wmanIfSsPkmTEKExpiresNew        DateAndTime,
38      wmanIfSsPkmTEKKeyRequests        Counter32,
39      wmanIfSsPkmTEKKeyReplies        Counter32,
40      wmanIfSsPkmTEKKeyRejects        Counter32,
41      wmanIfSsPkmTEKInvalids          Counter32,
42      wmanIfSsPkmTEKAuthPends         Counter32,
43      wmanIfSsPkmTEKKeyRejectErrorCode INTEGER,
44      wmanIfSsPkmTEKKeyRejectErrorString SnmpAdminString,
45      wmanIfSsPkmTEKInvalidErrorCode  INTEGER,
46      wmanIfSsPkmTEKInvalidErrorString SnmpAdminString
47  }
48
49  wmanIfSsPkmTEKSAID OBJECT-TYPE
50      SYNTAX      Integer32 (1..16383)
51      MAX-ACCESS  not-accessible
52      STATUS      current
53      DESCRIPTION
54          "The value of this object is the WiMAX Security Association

```

```

1         ID (SAID)."
```

::= { wmanIfSsPkmTEKEntry 1 }

```

3
4 wmanIfSsPkmTEKSAType OBJECT-TYPE
5     SYNTAX      INTEGER {none(0),
6                   primary(1),
7                   static(2),
8                   dynamic(3)}
9     MAX-ACCESS  read-only
10    STATUS      current
11    DESCRIPTION
12        "The value of this object is the type of security
13         association."
14    ::= { wmanIfSsPkmTEKEntry 2 }
```

::= { wmanIfSsPkmTEKEntry 2 }

```

15
16 wmanIfSsPkmTEKDataEncryptAlg OBJECT-TYPE
17     SYNTAX      INTEGER { none(0),
18                   des56CbcMode(1) }
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "The value of this object is the data encryption algorithm
23         being utilized."
24     REFERENCE
25         "IEEE 802.16 standard; Table 301"
26    ::= { wmanIfSsPkmTEKEntry 3 }
```

::= { wmanIfSsPkmTEKEntry 3 }

```

27
28 wmanIfSsPkmTEKDataAuthentAlg OBJECT-TYPE
29     SYNTAX      INTEGER { none(0) }
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33         "The value of this object is the data authentication
34         algorithm being utilized."
35     REFERENCE
36         "IEEE 802.16 standard; Table 302"
37    ::= { wmanIfSsPkmTEKEntry 4 }
```

::= { wmanIfSsPkmTEKEntry 4 }

```

38
39 wmanIfSsPkmTEKEncryptAlg OBJECT-TYPE
40     SYNTAX      INTEGER { tripleDES(0),
41                   rsa1024(1) }
42     MAX-ACCESS  read-only
43     STATUS      current
44     DESCRIPTION
45         "The value of this object is the TEK key encryption
46         algorithm for this cryptographic suite capability."
47     REFERENCE
48         "IEEE 802.16 standard; Table 303"
49    ::= { wmanIfSsPkmTEKEntry 5 }
```

::= { wmanIfSsPkmTEKEntry 5 }

```

50
51 wmanIfSsPkmTEKState OBJECT-TYPE
52     SYNTAX      INTEGER { start(1),
53                   opWait(2),
54                   opReauthWait(3),
```

```

1             operational(4),
2             rekeywait(5),
3             rekeyReauthwait(6) }
4     MAX-ACCESS    read-only
5     STATUS        current
6     DESCRIPTION
7         "The value of this object is the state of the indicated TEK
8         FSM. The start(1) state indicates that FSM is in its
9         initial state."
10    ::= { wmanIfSsPkmTEKEntry 6 }
11
12    wmanIfSsPkmTEKKeySequenceNumber OBJECT-TYPE
13        SYNTAX      Integer32 (0..15)
14        MAX-ACCESS  read-only
15        STATUS      current
16        DESCRIPTION
17            "The value of this object is the most recent TEK key
18            sequence number for this TEK FSM."
19        ::= { wmanIfSsPkmTEKEntry 7 }
20
21    wmanIfSsPkmTEKExpiresOld OBJECT-TYPE
22        SYNTAX      DateAndTime
23        MAX-ACCESS  read-only
24        STATUS      current
25        DESCRIPTION
26            "The value of this object is the actual clock time for
27            expiration of the immediate predecessor of the most recent
28            TEK for this FSM. If this FSM has only one TEK, then the
29            value is the time of activation of this FSM."
30        ::= { wmanIfSsPkmTEKEntry 8 }
31
32    wmanIfSsPkmTEKExpiresNew OBJECT-TYPE
33        SYNTAX      DateAndTime
34        MAX-ACCESS  read-only
35        STATUS      current
36        DESCRIPTION
37            "The value of this object is the actual clock time for
38            expiration of the most recent TEK for this FSM."
39        ::= { wmanIfSsPkmTEKEntry 9 }
40
41    wmanIfSsPkmTEKKeyRequests 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 SS has
47            transmitted a Key Request message."
48        ::= { wmanIfSsPkmTEKEntry 10 }
49
50    wmanIfSsPkmTEKKeyReplies OBJECT-TYPE
51        SYNTAX      Counter32
52        MAX-ACCESS  read-only
53        STATUS      current
54        DESCRIPTION

```

```

1           "The value of this object is the count of times the SS has
2             received a Key Reply message, including a message whose
3             authentication failed."
4       ::= { wmanIfSsPkmTEKEntry 11 }
5
6   wmanIfSsPkmTEKKeyRejects OBJECT-TYPE
7       SYNTAX      Counter32
8       MAX-ACCESS  read-only
9       STATUS      current
10      DESCRIPTION
11          "The value of this object is the count of times the SS has
12            received a Key Reject message, including a message whose
13            authentication failed."
14      ::= { wmanIfSsPkmTEKEntry 12 }
15
16   wmanIfSsPkmTEKInvalids OBJECT-TYPE
17       SYNTAX      Counter32
18       MAX-ACCESS  read-only
19       STATUS      current
20      DESCRIPTION
21          "The value of this object is the count of times the SS has
22            received a TEK Invalid message, including a message whose
23            authentication failed."
24      ::= { wmanIfSsPkmTEKEntry 13 }
25
26   wmanIfSsPkmTEKAuthPends OBJECT-TYPE
27       SYNTAX      Counter32
28       MAX-ACCESS  read-only
29       STATUS      current
30      DESCRIPTION
31          "The value of this object is the count of times an
32            Authorization Pending (Auth Pend) event occurred in this
33            FSM."
34      ::= { wmanIfSsPkmTEKEntry 14 }
35
36   wmanIfSsPkmTEKKeyRejectErrorCode OBJECT-TYPE
37       SYNTAX      INTEGER {none(1),
38                          unknown(2),
39                          unauthorizedSaid(4)}
40       MAX-ACCESS  read-only
41       STATUS      current
42      DESCRIPTION
43          "The value of this object is the enumerated description of
44            the Error-Code in most recent Key Reject message received
45            by the SS. This has value unknown(2) if the last Error-Code
46            value was 0, and none(1) if no Key Reject message has been
47            received since reboot."
48      ::= { wmanIfSsPkmTEKEntry 15 }
49
50   wmanIfSsPkmTEKKeyRejectErrorString OBJECT-TYPE
51       SYNTAX      SnmpAdminString (SIZE (0..128))
52       MAX-ACCESS  read-only
53       STATUS      current
54      DESCRIPTION

```

```

1         "The value of this object is the Display-String in most
2         recent Key Reject message received by the SS. This is a
3         zero length string if no Key Reject message has been
4         received since reboot."
5         ::= { wmanIfSsPkmTEKEntry 16 }
6
7 wmanIfSsPkmTEKInvalidErrorCode OBJECT-TYPE
8     SYNTAX      INTEGER {none(1),
9                 unknown(2),
10                invalidKeySequence(6)}
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "The value of this object is the enumerated description of
15         the Error-Code in most recent TEK Invalid message received
16         by the SS. This has value unknown(2) if the last
17         Error-Code value was 0, and none(1) if no TEK Invalid
18         message has been received since reboot."
19     ::= { wmanIfSsPkmTEKEntry 17 }
20
21 wmanIfSsPkmTEKInvalidErrorString OBJECT-TYPE
22     SYNTAX      SnmpAdminString (SIZE (0..128))
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "The value of this object is the Display-String in most
27         recent TEK Invalid message received by the SS. This is a
28         zero length string if no TEK Invalid message has been
29         received since reboot."
30     ::= { wmanIfSsPkmTEKEntry 18 }
31
32 --
33 -- Table wmanIfSsDeviceCertTable
34 --
35 wmanIfSsDeviceCertTable OBJECT-TYPE
36     SYNTAX      SEQUENCE OF wmanIfSsDeviceCertEntry
37     MAX-ACCESS  not-accessible
38     STATUS      current
39     DESCRIPTION
40         "This table describes the PKM device certificates for each
41         SS wireless interface."
42     ::= { wmanIfSsPkmObjects 3 }
43
44 wmanIfSsDeviceCertEntry OBJECT-TYPE
45     SYNTAX      wmanIfSsDeviceCertEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "Each entry contains the device certificate of one SS."
50     INDEX      { ifIndex }
51     ::= { wmanIfSsDeviceCertTable 1 }
52
53 wmanIfSsDeviceCertEntry ::= SEQUENCE {
54     wmanIfSsDeviceCert          OCTET STRING,

```



```

1          wmanIfSsDeviceManufCert          OCTET STRING
2      }
3
4
5  wmanIfSsDeviceCert OBJECT-TYPE
6      SYNTAX          OCTET STRING
7      MAX-ACCESS      read-only
8      STATUS          current
9      DESCRIPTION
10         "The X509 DER-encoded subscriber station certificate."
11         ::= { wmanIfSsDeviceCertEntry 1 }
12
13
14  wmanIfSsDeviceManufCert OBJECT-TYPE
15      SYNTAX          OCTET STRING
16      MAX-ACCESS      read-only
17      STATUS          current
18      DESCRIPTION
19         "The X509 DER-encoded manufacturer certificate which is
20         signed by the CA root authority certificate."
21         ::= { wmanIfSsDeviceCertEntry 2 }
22
23  --
24  -- Subscriber station Notification Group
25  -- wmanIfSsNotificationObjects contains the SS SNMP Trap objects
26  --
27  wmanIfSsNotification OBJECT IDENTIFIER ::= { wmanIfSsObjects 4 }
28  wmanIfSsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIfSsNotification 1 }
29  wmanIfSsTrapControl OBJECT IDENTIFIER ::= { wmanIfSsNotification 2 }
30
31  wmanIfSsTrapControlRegister OBJECT-TYPE
32      SYNTAX          BITS { wmanSwUpgradeInit(0),
33                           wmanSwUpgradeFail(1),
34                           wmanSwUpgradeSuccess(2),
35                           wmanSwConfigurationChanged(3),
36                           wmanPswdChangeFail(4),
37                           wmanPowerSupplyFail(5),
38                           wmanTODFail(6),
39                           wmanSsTLVUnknown(7),
40                           wmanSsDynServReqFail(8),
41                           wmanSsDynServRspFail(9),
42                           wmanSsDynServAckFail(10),
43                           wmanSsBPkMFail(11),
44                           wmanSsDHCPSuccess(12)
45                           }
46      MAX-ACCESS      read-write
47      STATUS          current
48      DESCRIPTION
49         "The object is used to enable Subscriber Station traps.
50         From left to right, the set bit indicates the corresponding
51         Subscriber Station trap is enabled."
52         ::= { wmanIfSsTrapControl 1 }
53
54  wmanBSTLVUnknown NOTIFICATION-TYPE

```

```

1      STATUS      current
2      DESCRIPTION
3          "Event that notifies detection of unknown TLV during
4  the
5          TLV parsing process."
6      ::= { wmanIfSsTrapDefinitions 1 }
7
8  wmanSsDynServReqFailTrap  NOTIFICATION-TYPE
9      STATUS      current
10     DESCRIPTION
11         "An event to report the failure of a dynamic service
12         request happened during the dynamic services process"
13     ::= { wmanIfSsTrapDefinitions 2 }
14
15  wmanSsDynServRspFailTrap  NOTIFICATION-TYPE
16     STATUS      current
17     DESCRIPTION
18         "An event to report the failure of a dynamic service
19         response happened during the dynamic services process."
20     ::= { wmanIfSsTrapDefinitions 3 }
21
22  wmanSsDynServAckFailTrap  NOTIFICATION-TYPE
23     STATUS      current
24     DESCRIPTION
25         "An event to report the failure of a dynamic service
26         acknowledgement happened during the dynamic services
27         process."
28     ::= { wmanIfSsTrapDefinitions 4 }
29
30  wmanSsBPKMFailTrap      NOTIFICATION-TYPE
31     STATUS      current
32     DESCRIPTION
33         "An event to report the failure of a BPKM operation."
34     ::= { wmanIfSsTrapDefinitions 5 }
35
36  wmanSsDHCPSSuccessTrap  NOTIFICATION-TYPE
37     STATUS      current
38     DESCRIPTION
39         "An event to report a successful DHCP Handshake for
40         the SS."
41     ::= { wmanIfSsTrapDefinitions 6 }
42
43  --
44  -- Common object group - containing common tables and objects to be
45  -- implemented in both Base Station and Subscriber Station
46  --
47  -- wmanIfCmnPacketCs contain the Packet Convergence Sublayer objects
48  -- that are common to both Base Station and Subscriber Station
49  wmanIfCmnPacketCs OBJECT IDENTIFIER ::= { wmanIfCommonObjects 1 }
50
51  wmanIfCmnClassifierRuleTable OBJECT-TYPE
52      SYNTAX      SEQUENCE OF wmanIfCmnClassifierRuleEntry
53      MAX-ACCESS  not-accessible
54      STATUS      current

```

```

1      DESCRIPTION
2          "This table contains packet classifier rules associated
3          with service flows."
4      ::= { wmanIfCmnPacketCs 1 }
5
6      wmanIfCmnClassifierRuleEntry OBJECT-TYPE
7          SYNTAX      wmanIfCmnClassifierRuleEntry
8          MAX-ACCESS  not-accessible
9          STATUS      current
10         DESCRIPTION
11             "This table provides one row for each packet classifier
12             rule, and is indexed by wmanIfCmnPcsSfIndex and
13             wmanIfCmnClassifierRuleIndex. wmanIfCmnPcsSfIndex identifies
14             the service flow, and wmanIfCmnClassifierRuleIndexAn
15             identifies the packet classifier rule."
16         INDEX { wmanIfCmnClassifierRuleIndex, wmanIfCmnPcsSfIndex }
17         ::= { wmanIfCmnClassifierRuleTable 1 }
18
19     wmanIfCmnClassifierRuleEntry ::= SEQUENCE {
20         wmanIfCmnClassifierRuleIndex      Unsigned32,
21         wmanIfCmnPcsSfIndex                Unsigned32,
22         wmanIfCmnClassifierRuleServiceFlowId Unsigned32,
23         wmanIfCmnClassifierRulePriority    INTEGER,
24         wmanIfCmnClassifierRuleIpTosLow   OCTET STRING,
25         wmanIfCmnClassifierRuleIpTosHigh  OCTET STRING,
26         wmanIfCmnClassifierRuleIpTosMask  OCTET STRING,
27         wmanIfCmnClassifierRuleIpProtocol Integer32,
28         wmanIfCmnClassifierRuleInetAddressType InetAddressType,
29         wmanIfCmnClassifierRuleInetAddress InetAddress,
30         wmanIfCmnClassifierRuleInetAddressMask InetAddress,
31         wmanIfCmnClassifierRuleInetAddress InetAddress,
32         wmanIfCmnClassifierRuleInetAddressMask InetAddress,
33         wmanIfCmnClassifierRuleSourcePortStart Integer32,
34         wmanIfCmnClassifierRuleSourcePortEnd Integer32,
35         wmanIfCmnClassifierRuleDestPortStart Integer32,
36         wmanIfCmnClassifierRuleDestPortEnd Integer32,
37         wmanIfCmnClassifierRuleDestMacAddr MacAddress,
38         wmanIfCmnClassifierRuleDestMacMask MacAddress,
39         wmanIfCmnClassifierRuleSourceMacAddr MacAddress,
40         wmanIfCmnClassifierRuleSourceMacMask MacAddress,
41         wmanIfCmnClassifierRuleEnetProtocolType INTEGER,
42         wmanIfCmnClassifierRuleEnetProtocol Integer32,
43         wmanIfCmnClassifierRuleUserPriLow Integer32,
44         wmanIfCmnClassifierRuleUserPriHigh Integer32,
45         wmanIfCmnClassifierRuleVlanId     Integer32,
46         wmanIfCmnClassifierRuleState      INTEGER,
47         wmanIfCmnClassifierRulePkts       Counter64,
48         wmanIfCmnClassifierRuleRowStatus  RowStatus
49     }
50
51     wmanIfCmnClassifierRuleIndex OBJECT-TYPE
52         SYNTAX      Unsigned32 (1..4294967295)
53         MAX-ACCESS  not-accessible
54         STATUS      current

```

```

1      DESCRIPTION
2          "An index is assigned to each classifier in the classifiers
3          table"
4      REFERENCE      ""
5      ::= { wmanIfCmnClassifierRuleEntry 1 }
6
7      wmanIfCmnPcsSfIndex OBJECT-TYPE
8          SYNTAX      Unsigned32 (1 .. 4294967295)
9          MAX-ACCESS  not-accessible
10         STATUS      current
11         DESCRIPTION
12             "A 32 bit quantity that uniquely identifies a service flow
13             to both the subscriber station and base station (BS)."
14         ::= { wmanIfCmnClassifierRuleEntry 2 }
15
16         wmanIfCmnClassifierRuleServiceFlowId OBJECT-TYPE
17             SYNTAX      Unsigned32 (1..4294967295)
18             MAX-ACCESS  read-write
19             STATUS      current
20             DESCRIPTION
21                 "An index assigned to a service flow by SC (SFID)."
22             REFERENCE  "802.16 Chapter 11.4.8"
23             ::= { wmanIfCmnClassifierRuleEntry 3 }
24
25         wmanIfCmnClassifierRulePriority OBJECT-TYPE
26             SYNTAX      INTEGER
27             MAX-ACCESS  read-write
28             STATUS      current
29             DESCRIPTION
30                 "The value specifies the order of evaluation of the
31                 classifiers. The higher the value the higher the
32                 priority. The value of 0 is used as default in
33                 provisioned service flows classifiers. The default
34                 value of 64 is used for dynamic service flow classifiers.
35                 If the referenced parameter is not present in a classifier,
36                 this object reports the default value as defined above"
37             ::= { wmanIfCmnClassifierRuleEntry 4 }
38
39         wmanIfCmnClassifierRuleIpTosLow OBJECT-TYPE
40             SYNTAX      OCTET STRING (SIZE(1))
41             MAX-ACCESS  read-write
42             STATUS      current
43             DESCRIPTION
44                 "The low value of a range of TOS byte values. If the
45                 referenced parameter is not present in a classifier, this
46                 object reports the value of 0."
47             REFERENCE  "802.16 Chapter 11.4.9"
48             ::= { wmanIfCmnClassifierRuleEntry 5 }
49
50         wmanIfCmnClassifierRuleIpTosHigh OBJECT-TYPE
51             SYNTAX      OCTET STRING (SIZE(1))
52             MAX-ACCESS  read-write
53             STATUS      current
54             DESCRIPTION

```

```

1           "The 8-bit high value of a range of TOS byte values.
2           If the referenced parameter is not present in a classifier,
3           this object reports the value of 0."
4     REFERENCE      "802.16 Chapter 11.4.9"
5     ::= { wmanIfCmnClassifierRuleEntry 6 }
6
7     wmanIfCmnClassifierRuleIpTosMask OBJECT-TYPE
8     SYNTAX          OCTET STRING (SIZE(1))
9     MAX-ACCESS      read-write
10    STATUS           current
11    DESCRIPTION
12    "The mask value is bitwise ANDed with TOS byte in an IP
13    packet and this value is used check range checking of
14    TosLow and TosHigh. If the referenced parameter is not
15    present in a classifier, this object reports the value
16    of 0."
17    REFERENCE      "802.16 Chapter 11.4.9"
18    ::= { wmanIfCmnClassifierRuleEntry 7 }
19
20    wmanIfCmnClassifierRuleIpProtocol OBJECT-TYPE
21    SYNTAX          Integer32 (0..255)
22    MAX-ACCESS      read-write
23    STATUS           current
24    DESCRIPTION
25    "This object indicates the value of the IP Protocol field
26    required for IP packets to match this rule. If the
27    referenced parameter is not present in a classifier, this
28    object reports the value of 0."
29    REFERENCE      "802.16 Chapter 11.4.9"
30    ::= { wmanIfCmnClassifierRuleEntry 8 }
31
32    wmanIfCmnClassifierRuleInetAddressType OBJECT-TYPE
33    SYNTAX          InetAddressType
34    MAX-ACCESS      read-write
35    STATUS           current
36    DESCRIPTION
37    "The type of the internet address for
38    wmanIfCmnClassifierRuleInetAddressSourceAddr,
39    wmanIfCmnClassifierRuleInetAddressSourceMask,
40    wmanIfCmnClassifierRuleInetAddressDestAddr, and
41    wmanIfCmnClassifierRuleInetAddressDestMask.
42    If the referenced parameter is not present in a classifier,
43    this object reports the value of ipv4(1)."
44    REFERENCE      ""
45    ::= { wmanIfCmnClassifierRuleEntry 9 }
46
47    wmanIfCmnClassifierRuleInetAddressSourceAddr OBJECT-TYPE
48    SYNTAX          InetAddress
49    MAX-ACCESS      read-write
50    STATUS           current
51    DESCRIPTION
52    "This object specifies the value of the IP Source Address
53    required for packets to match this rule. An IP packet
54    matches the rule when the packet ip source address bitwise

```

```

1         ANDED with the wmanIfCmnClassifierRuleInetSourceMask value
2         equals the wmanIfCmnClassifierRuleInetSourceAddr value.
3         If the referenced parameter is not present n a classifier,
4         this object reports the value of 0.0.0.0."
5     REFERENCE      "802.16 Chapter 11.4.9"
6     ::= { wmanIfCmnClassifierRuleEntry 10 }
7
8     wmanIfCmnClassifierRuleInetSourceMask OBJECT-TYPE
9         SYNTAX      InetAddress
10        MAX-ACCESS  read-write
11        STATUS      current
12        DESCRIPTION
13            "This object specifies which bits of a packet's IP Source
14            Address that are compared to match this rule. An IP packet
15            matches the rule when the packet source address bitwise
16            ANDED with the
17            wmanIfCmnClassifierRuleInetSourceMask value equals the
18            wmanIfCmnClassifierRuleInetSourceAddr value.
19            If the referenced parameter is not present in a classifier,
20            this object reports the value of 0.0.0.0."
21        REFERENCE   "802.16 Chapter 11.4.9"
22        ::= { wmanIfCmnClassifierRuleEntry 11 }
23
24     wmanIfCmnClassifierRuleInetDestAddr OBJECT-TYPE
25         SYNTAX      InetAddress
26         MAX-ACCESS  read-write
27         STATUS      current
28         DESCRIPTION
29             "This object specifies the value of the IP Destination
30             Address required for packets to match this rule. An IP
31             packet matches the rule when the packet IP destination
32             address bitwise ANDED with the
33             wmanIfCmnClassifierRuleInetDestMask value equals the
34             wmanIfCmnClassifierRuleInetDestAddr value.
35             If the referenced parameter is not present in a
36             classifier, this object reports the value of 0.0.0.0."
37         REFERENCE   "802.16 Chapter 11.4.9"
38         ::= { wmanIfCmnClassifierRuleEntry 12 }
39
40     wmanIfCmnClassifierRuleInetDestMask OBJECT-TYPE
41         SYNTAX      InetAddress
42         MAX-ACCESS  read-write
43         STATUS      current
44         DESCRIPTION
45             "This object specifies which bits of a packet's IP
46             Destination Address that are compared to match this rule.
47             An IP packet matches the rule when the packet destination
48             address bitwise ANDED with the
49             wmanIfCmnClassifierRuleInetDestMask value equals the
50             wmanIfCmnClassifierRuleInetDestAddr value.
51             If the referenced parameter is not present in a classifier
52             , this object reports the value of 0.0.0.0."
53         REFERENCE   "802.16 Chapter 11.4.9"
54         ::= { wmanIfCmnClassifierRuleEntry 13 }

```

```

1
2  wmanIfCmnClassifierRuleSourcePortStart OBJECT-TYPE
3      SYNTAX      Integer32 (0..65535)
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "This object specifies the low end inclusive range of
8          TCP/UDP source port numbers to which a packet is compared
9          . This object is irrelevant for non-TCP/UDP IP packets.
10         If the referenced parameter is not present in a
11         classifier, this object reports the value of 0."
12     REFERENCE    "802.16 Chapter 11.4.9"
13     ::= { wmanIfCmnClassifierRuleEntry 14 }
14
15  wmanIfCmnClassifierRuleSourcePortEnd OBJECT-TYPE
16     SYNTAX      Integer32 (0..65535)
17     MAX-ACCESS  read-write
18     STATUS      current
19     DESCRIPTION
20         "This object specifies the high end inclusive range of
21         TCP/UDP source port numbers to which a packet is compared.
22         This object is irrelevant for non-TCP/UDP IP packets.
23         If the referenced parameter is not present in a classifier,
24         this object reports the value of 65535."
25     REFERENCE    "802.16 Chapter 11.4.9"
26     ::= { wmanIfCmnClassifierRuleEntry 15 }
27
28  wmanIfCmnClassifierRuleDestPortStart OBJECT-TYPE
29     SYNTAX      Integer32 (0..65535)
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "This object specifies the low end inclusive range of
34         TCP/UDP destination port numbers to which a packet is
35         compared. If the referenced parameter is not present
36         in a classifier, this object reports the value of 0."
37     REFERENCE    "802.16 Chapter 11.4.9"
38     ::= { wmanIfCmnClassifierRuleEntry 16 }
39
40  wmanIfCmnClassifierRuleDestPortEnd OBJECT-TYPE
41     SYNTAX      Integer32 (0..65535)
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "This object specifies the high end inclusive range of
46         TCP/UDP destination port numbers to which a packet is
47         compared. If the referenced parameter is not present
48         in a classifier, this object reports the value of
49         65535."
50     REFERENCE    "802.16 Chapter 11.4.9"
51     ::= { wmanIfCmnClassifierRuleEntry 17 }
52
53  wmanIfCmnClassifierRuleDestMacAddr OBJECT-TYPE
54     SYNTAX      MacAddress

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "An Ethernet packet matches an entry when its destination
5          MAC address bitwise ANDed with
6          wmanIfCmnClassifierRuleDestMacMask equals the value of
7          wmanIfCmnClassifierRuleDestMacAddr. If the referenced
8          parameter is not present in a classifier, this object
9          reports the value of '000000000000'H."
10     REFERENCE    "802.16 Chapter 11.4.9"
11     ::= { wmanIfCmnClassifierRuleEntry 18 }
12
13     wmanIfCmnClassifierRuleDestMacMask OBJECT-TYPE
14     SYNTAX        MacAddress
15     MAX-ACCESS    read-write
16     STATUS        current
17     DESCRIPTION
18         "An Ethernet packet matches an entry when its destination
19         MAC address bitwise ANDed with
20         wmanIfCmnClassifierRuleDestMacMask equals the value of
21         wmanIfCmnClassifierRuleDestMacAddr. If the referenced
22         parameter is not present in a classifier, this object
23         reports the value of '000000000000'H."
24     REFERENCE    "802.16 Chapter 11.4.9"
25     ::= { wmanIfCmnClassifierRuleEntry 19 }
26
27     wmanIfCmnClassifierRuleSourceMacAddr OBJECT-TYPE
28     SYNTAX        MacAddress
29     MAX-ACCESS    read-write
30     STATUS        current
31     DESCRIPTION
32         "An Ethernet packet matches this entry when its source
33         MAC address bitwise ANDed with
34         wmanIfCmnClassifierRuleSourceMacMask equals the value
35         of wmanIfCmnClassifierRuleSourceMacAddr. If the
36         referenced parameter is not present in a classifier,
37         this object reports the value of 'FFFFFFFFFFFF'H."
38     REFERENCE    "802.16 Chapter 11.4.9"
39     ::= { wmanIfCmnClassifierRuleEntry 20 }
40
41     wmanIfCmnClassifierRuleSourceMacMask OBJECT-TYPE
42     SYNTAX        MacAddress
43     MAX-ACCESS    read-write
44     STATUS        current
45     DESCRIPTION
46         "An Ethernet packet matches an entry when its destination
47         MAC address bitwise ANDed with
48         wmanIfCmnClassifierRuleSourceMacMask equals the value of
49         wmanIfCmnClassifierRuleSourceMacAddr. If the referenced
50         parameter is not present in a classifier, this object
51         reports the value of '000000000000'H."
52     REFERENCE    "802.16 Chapter 11.4.9"
53     ::= { wmanIfCmnClassifierRuleEntry 21 }
54

```



```

1  wmanIfCmnClassifierRuleEnetProtocolType OBJECT-TYPE
2      SYNTAX      INTEGER {none(0),
3                  ethertype(1),
4                  dsap(2)}
5      MAX-ACCESS  read-write
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      REFERENCE   "802.16 Chapter 11.4.9"
26      ::= { wmanIfCmnClassifierRuleEntry 22 }
27
28  wmanIfCmnClassifierRuleEnetProtocol OBJECT-TYPE
29      SYNTAX      Integer32 (0..65535)
30      MAX-ACCESS  read-write
31      STATUS      current
32      DESCRIPTION
33          "If wmanIfCmnClassifierRuleEnetProtocolType is none(0),
34          this object is ignored when considering whether a packet
35          matches the current rule.
36          If wmanIfCmnClassifierRuleEnetProtocolType is ethertype(1),
37          this object gives the 16-bit value of the EtherType that
38          the packet must match in order to match the rule.
39          If wmanIfCmnClassifierRuleEnetProtocolType is dsap(2), the
40          lower 8 bits of this object's value must match the DSAP
41          byte of the packet in order to match the rule.
42          If wmanIfCmnClassifierRuleEnetProtocolType is mac(3), the
43          lower 8 bits of this object value represent a lower bound
44          (inclusive) of MAC management message type codes matched,
45          and the upper 8 bits of this object value represent the
46          upper bound (inclusive) of matched MAC message type codes.
47          Certain message type codes are excluded from matching, as
48          specified in the reference.
49          If the Ethernet frame contains an 802.1P/Q Tag header
50          (i.e. EtherType 0x8100), this object applies to the
51          embedded EtherType field within the 802.1P/Q header.
52          If the referenced parameter is not present in the
53          classifier, the value of this object is reported as 0."
54      REFERENCE   "802.16 Chapter 11.4.9"

```

```

1      ::= { wmanIfCmnClassifierRuleEntry 23 }
2
3  wmanIfCmnClassifierRuleUserPriLow OBJECT-TYPE
4      SYNTAX      Integer32 (0..7)
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8          "This object applies only to Ethernet frames using the
9          802.1P/Q tag header (indicated with EtherType 0x8100).
10         Such frames include a 16-bit Tag that contains a 3 bit
11         Priority field and a 12 bit VLAN number.
12         Tagged Ethernet packets must have a 3-bit Priority field
13         within the range of wmanIfCmnClassifierRulePriLow and
14         wmanIfCmnClassifierRulePriHigh in order to match this
15         rule.
16         If the referenced parameter is not present in the
17         classifier, the value of this object is reported as 0."
18     REFERENCE   "802.16 Chapter 11.4.9"
19     ::= { wmanIfCmnClassifierRuleEntry 24 }
20
21  wmanIfCmnClassifierRuleUserPriHigh OBJECT-TYPE
22     SYNTAX      Integer32 (0..7)
23     MAX-ACCESS  read-write
24     STATUS      current
25     DESCRIPTION
26         "This object applies only to Ethernet frames using the
27         802.1P/Q tag header (indicated with EtherType 0x8100).
28         Such frames include a 16-bit Tag that contains a 3 bit
29         Priority field and a 12 bit VLAN number.
30         Tagged Ethernet packets must have a 3-bit Priority
31         field within the range of wmanIfCmnClassifierRulePriLow
32         and wmanIfCmnClassifierRulePriHigh in order to match
33         this rule.
34         If the referenced parameter is not present in the
35         classifier, the value of this object is reported as 7."
36     REFERENCE   "802.16 Chapter 11.4.9"
37     ::= { wmanIfCmnClassifierRuleEntry 25 }
38
39  wmanIfCmnClassifierRuleVlanId OBJECT-TYPE
40     SYNTAX      Integer32 (0..4095)
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "This object applies only to Ethernet frames using the
45         802.1P/Q tag header.
46         If this object's value is nonzero, tagged packets must
47         have a VLAN Identifier that matches the value in order
48         to match the rule.
49         Only the least significant 12 bits of this object's
50         value are valid.
51         If the referenced parameter is not present in the
52         classifier, the value of this object is reported as 0."
53     REFERENCE   "802.16 Chapter 11.4.9"
54     ::= { wmanIfCmnClassifierRuleEntry 26 }

```

```

1
2 wmanIfCmnClassifierRuleState OBJECT-TYPE
3     SYNTAX      INTEGER {active(1),
4                 inactive(2)}
5     MAX-ACCESS  read-write
6     STATUS      current
7     DESCRIPTION
8         "This object indicates whether or not the classifier is
9         enabled to classify packets to a Service Flow.
10        If the referenced parameter is not present in the
11        classifier, the value of this object is reported
12        as active(1)."
13     REFERENCE   "802.16 Chapter 11.4.9"
14     ::= { wmanIfCmnClassifierRuleEntry 27 }
15
16 wmanIfCmnClassifierRulePkts OBJECT-TYPE
17     SYNTAX      Counter64
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21         "This object counts the number of packets that have
22         been classified using this entry."
23     ::= { wmanIfCmnClassifierRuleEntry 28 }
24
25 wmanIfCmnClassifierRuleRowStatus OBJECT-TYPE
26     SYNTAX      RowStatus
27     MAX-ACCESS  read-create
28     STATUS      current
29     DESCRIPTION
30         "This object is used to create a new row or modify or
31         delete an existing row in this table.
32
33         If the implementator of this MIB has chosen not
34         to implement 'dynamic assignment' of profiles, this
35         object is not useful and should return noSuchName
36         upon SNMP request."
37     ::= { wmanIfCmnClassifierRuleEntry 29 }
38
39 --
40 -- wmanIfCmnCps contain the Common Part Sublayer objects that are common
41 -- to both Base Station and Subscriber Station
42 wmanIfCmnCps OBJECT IDENTIFIER ::= { wmanIfCommonObjects 2 }
43
44 wmanIfCmnCpsServiceFlowTable OBJECT-TYPE
45     SYNTAX      SEQUENCE OF wmanIfCmnCpsServiceFlowEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "This table contains Service Flows that are created in both
50         BS and SS."
51     ::= { wmanIfCmnCps 1 }
52
53 wmanIfCmnCpsServiceFlowEntry OBJECT-TYPE
54     SYNTAX      wmanIfCmnCpsServiceFlowEntry

```

```

1      MAX-ACCESS not-accessible
2      STATUS      current
3      DESCRIPTION
4          "This table provides one row for each service flow, and is
5          indexed by wmanIfCmnCpsSfId. The value of wmanIfCmnCpsSfId
6          is obtained from wmanIfBssSfId."
7      INDEX      { wmanIfCmnCpsSfId }
8      ::= { wmanIfCmnCpsServiceFlowTable 1 }
9
10     wmanIfCmnCpsServiceFlowEntry ::= SEQUENCE {
11         wmanIfCmnCpsSfId                Unsigned32,
12         wmanIfCmnCpsSfCid                INTEGER,
13         wmanIfCmnCpsSfDirection          INTEGER,
14         wmanIfCmnCpsServiceClassIndex    INTEGER,
15         wmanIfCmnCpsSfState              INTEGER,
16         wmanIfCmnCpsServiceClassName     DisplayString,
17         wmanIfCmnCpsTrafficPriority       INTEGER,
18         wmanIfCmnCpsMaxSustainedRate     INTEGER,
19         wmanIfCmnCpsMaxTrafficBurst      INTEGER,
20         wmanIfCmnCpsMinReservedRate      INTEGER,
21         wmanIfCmnCpsToleratedJitter      INTEGER,
22         wmanIfCmnCpsMaxLatency           INTEGER,
23         wmanIfCmnCpsScSchedulingType     wmanUlSchedulingType,
24         wmanIfCmnCpsScArqEnable          TruthValue,
25         wmanIfCmnCpsScArqWindowSize      INTEGER,
26         wmanIfCmnCpsScArqFragmentLifetime INTEGER,
27         wmanIfCmnCpsScArqSyncLossTimeout INTEGER,
28         wmanIfCmnCpsScArqDeliverInOrder  TruthValue,
29         wmanIfCmnCpsScArqRxPurgeTimeout  INTEGER,
30         wmanIfCmnCpsScFragmentLen        INTEGER,
31         wmanIfCmnCpsSCMinRsvdTolerableRate INTEGER
32     }
33
34     wmanIfCmnCpsSfId OBJECT-TYPE
35         SYNTAX      Unsigned32 ( 1 .. 4294967295)
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "A 32 bit quantity that uniquely identifies a service flow
40             to both the subscriber station and base station (BS)."
41         ::= { wmanIfCmnCpsServiceFlowEntry 1 }
42
43     wmanIfCmnCpsSfCid OBJECT-TYPE
44         SYNTAX      INTEGER
45         MAX-ACCESS  read-only
46         STATUS      current
47         DESCRIPTION
48             "A 16 bit channel identifier to identify the connection
49             being created by DSA."
50         ::= { wmanIfCmnCpsServiceFlowEntry 2 }
51
52     wmanIfCmnCpsSfDirection OBJECT-TYPE
53         SYNTAX      INTEGER {downstream(1),
54             upstream(2)}

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "An attribute indicating the service flow is downstream or
5          upstream."
6      ::= { wmanIfCmnCpsServiceFlowEntry 3 }
7
8  wmanIfCmnCpsServiceClassIndex OBJECT-TYPE
9      SYNTAX      INTEGER
10     MAX-ACCESS  read-only
11     STATUS      current
12     DESCRIPTION
13         "The wmanIfCmnCpsServiceClassIndex associates this
14         service flow with QoS parameters.
15         If no associated entry in wmanIfCmnCpsServiceFlowTable
16         exists, this object returns a value of zero."
17     ::= { wmanIfCmnCpsServiceFlowEntry 4 }
18
19  wmanIfCmnCpsSfState OBJECT-TYPE
20     SYNTAX      INTEGER {provisionedState(1),
21                     admittedState(2),
22                     activeState(3)}
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "wmanIfCmnCpsSfState indicates the service flow state:
27         Provisioned, AdmittedState(2), and Active service flow
28         state."
29     REFERENCE
30         "Section 6.4.13.6, in IEEE 802.16REVd/D3-2004"
31     ::= { wmanIfCmnCpsServiceFlowEntry 5 }
32
33  wmanIfCmnCpsServiceClassName OBJECT-TYPE
34     SYNTAX      DisplayString
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38         "Refers to the Service Class Name"
39     REFERENCE   "802.16 Chapter 11.4.8"
40     ::= { wmanIfCmnCpsServiceFlowEntry 6 }
41
42  wmanIfCmnCpsTrafficPriority OBJECT-TYPE
43     SYNTAX      INTEGER
44     MAX-ACCESS  read-only
45     STATUS      current
46     DESCRIPTION
47         "The value of this parameter specifies the priority
48         assigned to a service flow. For uplink service flows,
49         the BS should use this parameter when determining
50         precedence in request service and grant generation,
51         and the SS shall preferentially select contention
52         Request opportunities for Priority Request CIDs
53         based on this priority"
54     REFERENCE

```

```

1         "Section 11.13.7 in IEEE 802.16REvD/D3-2004"
2         ::= { wmanIfCmnCpsServiceFlowEntry 7 }
3
4     wmanIfCmnCpsMaxSustainedRate OBJECT-TYPE
5         SYNTAX      INTEGER
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "This parameter defines the peak information rate
10            of the service. The rate is expressed in bits per
11            second and pertains to the SDUs at the input to
12            the system."
13        REFERENCE
14            "Section 11.13.8 in IEEE 802.16REvD/D3-2004"
15        ::= { wmanIfCmnCpsServiceFlowEntry 8 }
16
17    wmanIfCmnCpsMaxTrafficBurst OBJECT-TYPE
18        SYNTAX      INTEGER
19        MAX-ACCESS  read-only
20        STATUS      current
21        DESCRIPTION
22            "This parameter defines the maximum burst size that
23            must be accommodated for the service."
24        REFERENCE
25            "Section 11.13.9 in IEEE 802.16REvD/D3-2004"
26        ::= { wmanIfCmnCpsServiceFlowEntry 9 }
27
28    wmanIfCmnCpsMinReservedRate OBJECT-TYPE
29        SYNTAX      INTEGER
30        MAX-ACCESS  read-only
31        STATUS      current
32        DESCRIPTION
33            "This parameter specifies the minimum rate reserved
34            for this service flow."
35        REFERENCE
36            "Section 11.13.10 in IEEE 802.16REvD/D3-2004"
37        ::= { wmanIfCmnCpsServiceFlowEntry 10 }
38
39    wmanIfCmnCpsToleratedJitter OBJECT-TYPE
40        SYNTAX      INTEGER
41        MAX-ACCESS  read-only
42        STATUS      current
43        DESCRIPTION
44            "This parameter defines the Maximum delay
45            variation (jitter) for the connection."
46        REFERENCE
47            "Section 11.13.15 in IEEE 802.16REvD/D3-2004"
48        ::= { wmanIfCmnCpsServiceFlowEntry 11 }
49
50    wmanIfCmnCpsMaxLatency OBJECT-TYPE
51        SYNTAX      INTEGER
52        MAX-ACCESS  read-only
53        STATUS      current
54        DESCRIPTION

```

```

1           "The value of this parameter specifies the maximum
2           latency between the reception of a packet by the BS
3           or SS on its network interface and the forwarding
4           of the packet to its RF Interface."
5     REFERENCE
6           "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
7     ::= { wmanIfCmnCpsServiceFlowEntry 12 }
8
9     wmanIfCmnCpsScsSchedulingType OBJECT-TYPE
10      SYNTAX      wmanUlSchedulingType
11      MAX-ACCESS  read-only
12      STATUS      current
13      DESCRIPTION
14          "Specifies the upstream scheduling service used for
15          upstream service flow. If the referenced parameter
16          is not present in the corresponding 802.16 QoS
17          Parameter Set of an upstream service flow, the
18          default value of this object is bestEffort(2)."

```

```

1      SYNTAX      INTEGER (0 .. 65535 )
2      UNITS       "10 us"
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The maximum interval before declaring a loss
7            of synchronization of the sender and receiver
8            state machines. A value of 0 means Infinite."
9      ::= { wmanIfCmnCpsServiceFlowEntry 17}
10
11     wmanIfCmnCpsScArqDeliverInOrder OBJECT-TYPE
12         SYNTAX      TruthValue
13         MAX-ACCESS  read-only
14         STATUS      current
15         DESCRIPTION
16             "Indicates whether or not data is to be delivered
17               by the receiving MAC to its client application
18               in the order in which data was handed off to the
19               originating MAC."
20         ::= { wmanIfCmnCpsServiceFlowEntry 18 }
21
22     wmanIfCmnCpsScArqRXPurgeTimeout OBJECT-TYPE
23         SYNTAX      INTEGER (0 .. 65535)
24         UNITS       "10 us"
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "Indicates the time interval the ARQ window is advanced
29               after a fragment is received. A value of 0 means
30               Infinite."
31         ::= { wmanIfCmnCpsServiceFlowEntry 19}
32
33     wmanIfCmnCpsScFragmentLen OBJECT-TYPE
34         SYNTAX      INTEGER (32 .. 2040)
35         MAX-ACCESS  read-only
36         STATUS      current
37         DESCRIPTION
38             "The maximum size fragment a transmitter shall form
39               or a receiver shall expect to receive."
40         ::= { wmanIfCmnCpsServiceFlowEntry 20 }
41
42     wmanIfCmnCpsSCMinRsvdTolerableRate OBJECT-TYPE
43         SYNTAX      INTEGER
44         MAX-ACCESS  read-only
45         STATUS      current
46         DESCRIPTION
47             "Minimum Tolerable Traffic Rate = R (bits/sec) with
48               time base T(sec) means the following. Let S denote
49               additional demand accumulated at the MAC SAP of the
50               transmitter during an arbitrary time interval of the
51               length T. Then the amount of data forwarded at the
52               receiver to CS (in bits) during this interval should
53               be not less than min {S, R * T}."
54         REFERENCE   "Section 11.13.11 in IEEE 802.16REVD/D3-2004"

```



```

1      ::= { wmanIfCmnCpsServiceFlowEntry 21 }
2
3      --
4      -- wmanIfCmnBsSsConfigurationTable contains global parameters
5      -- common in BS and SS
6      wmanIfCmnBsSsConfigurationTable OBJECT-TYPE
7          SYNTAX      SEQUENCE OF wmanIfCmnBsSsConfigurationEntry
8          MAX-ACCESS  not-accessible
9          STATUS      current
10         DESCRIPTION
11             "This table provides one row for each SS that contains
12             the system parameters as defined in section 10.1 of [3]."
13         ::= { wmanIfCmnCps 2 }
14
15     wmanIfCmnBsSsConfigurationEntry OBJECT-TYPE
16         SYNTAX      wmanIfCmnBsSsConfigurationEntry
17         MAX-ACCESS  not-accessible
18         STATUS      current
19         DESCRIPTION
20             "This table is indexed by wmanIfCmnSsIdIndex."
21         INDEX { ifIndex }
22         ::= { wmanIfCmnBsSsConfigurationTable 1 }
23
24     wmanIfCmnBsSsConfigurationEntry ::= SEQUENCE {
25         wmanIfCmnSsId                Unsigned32,
26         wmanIfCmnInvitedRangRetries   INTEGER,
27         wmanIfCmnMinislotSize         INTEGER,
28         wmanIfCmnDSxReqRetries        INTEGER,
29         wmanIfCmnDSxRespRetries       INTEGER,
30         wmanIfCmnT7Timeout            INTEGER,
31         wmanIfCmnT8Timeout            INTEGER,
32         wmanIfCmnT10Timeout           INTEGER,
33         wmanIfCmnT22Timeout           INTEGER,
34         wmanIfCmnBsSsConfigurationRowStatus RowStatus
35     }
36
37     wmanIfCmnSsId OBJECT-TYPE
38         SYNTAX      Unsigned32 (1 .. 4294967295)
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "wmanIfCmnSsId is the index to
43             wmanIfCmnBsSsConfigurationTable."
44         REFERENCE
45             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
46         ::= { wmanIfCmnBsSsConfigurationEntry 1 }
47
48     wmanIfCmnInvitedRangRetries OBJECT-TYPE
49         SYNTAX      INTEGER(16..65535)
50         MAX-ACCESS  read-write
51         STATUS      current
52         DESCRIPTION
53             "Number of retries on inviting Ranging Requests."
54         ::= { wmanIfCmnBsSsConfigurationEntry 2 }

```

```
1
2 wmanIfCmnMinislotSize OBJECT-TYPE
3     SYNTAX      INTEGER
4     MAX-ACCESS  read-write
5     STATUS      current
6     DESCRIPTION
7         "Size of minislot for uplink transmission. Shall be a power
8         of 2 (in units of PS)."
```

```
9     ::= { wmanIfCmnBsSsConfigurationEntry 3 }
10
11 wmanIfCmnDSxReqRetries OBJECT-TYPE
12     SYNTAX      INTEGER
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Number of Timeout Retries on DSA/DSC/DSD Requests."
```

```
17     ::= { wmanIfCmnBsSsConfigurationEntry 4 }
18
19 wmanIfCmnDSxRespRetries OBJECT-TYPE
20     SYNTAX      INTEGER
21     MAX-ACCESS  read-write
22     STATUS      current
23     DESCRIPTION
24         "Number of Timeout Retries on DSA/DSC/DSD Responses."
```

```
25     ::= { wmanIfCmnBsSsConfigurationEntry 5 }
26
27 wmanIfCmnT7Timeout OBJECT-TYPE
28     SYNTAX      INTEGER(0 .. 1000)
29     UNITS       "milliseconds"
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "Wait for DSA/DSC/DSD Response Timeout in ms."
```

```
34     ::= { wmanIfCmnBsSsConfigurationEntry 6 }
35
36 wmanIfCmnT8Timeout OBJECT-TYPE
37     SYNTAX      INTEGER(0 .. 300)
38     UNITS       "milliseconds"
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "Wait for DSA/DSC/DSD Acknowledge Timeout in ms."
```

```
43     ::= { wmanIfCmnBsSsConfigurationEntry 7 }
44
45 wmanIfCmnT10Timeout OBJECT-TYPE
46     SYNTAX      INTEGER(0 .. 3000)
47     UNITS       "milliseconds"
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51         "Wait for Transaction End timeout in ms."
```

```
52     ::= { wmanIfCmnBsSsConfigurationEntry 8 }
53
54 wmanIfCmnT22Timeout OBJECT-TYPE
```

```

1          SYNTAX      INTEGER(0 .. 500)
2          UNITS        "milliseconds"
3          MAX-ACCESS   read-write
4          STATUS       current
5          DESCRIPTION
6              "Wait for ARQ Reset in ms."
7          ::= { wmanIfCmnBsSsConfigurationEntry 9 }
8
9  wmanIfCmnBsSsConfigurationRowStatus OBJECT-TYPE
10         SYNTAX      RowStatus
11         MAX-ACCESS   read-create
12         STATUS       current
13         DESCRIPTION
14             "This object is used to create a new row or modify or
15             delete an existing row in this table.
16
17             If the implementator of this MIB has chosen not
18             to implement 'dynamic assignment' of profiles, this
19             object is not useful and should return noSuchName
20             upon SNMP request."
21         ::= { wmanIfCmnBsSsConfigurationEntry 10 }
22
23         --
24         -- wmanIfCmnSsStatCounter contain the performance statistics information
25         wmanIfCmnSsStatCounter OBJECT IDENTIFIER ::= { wmanIfCmnCps 3 }
26
27         wmanIfCmnSsChMeasurementTable OBJECT-TYPE
28         SYNTAX      SEQUENCE OF wmanIfCmnSsChMeasurementEntry
29         MAX-ACCESS   not-accessible
30         STATUS       current
31         DESCRIPTION
32             "This table contains channel measurement information
33             for each SS. BS retrieves the channel measurement
34             information from REP-REQ/RSP messages. This table contains
35             channel measurement information on the downlink signal
36             sent to SS."
37         ::= { wmanIfCmnSsStatCounter 1 }
38
39         wmanIfCmnSsChMeasurementEntry OBJECT-TYPE
40         SYNTAX      wmanIfCmnSsChMeasurementEntry
41         MAX-ACCESS   not-accessible
42         STATUS       current
43         DESCRIPTION
44             "Each entry in the table contains RSSI and CINR
45             signal quality measurement taken from the SS. The primary
46             index is teh ifIndex with ifType propBWAp2Mp identifying
47             the BS sector. The primary index is the ifIndex with ifType
48             of propBWAp2Mp identifying the BS sector. wmanIfCmnSsIdIndex
49             identifies the SS where th emeasurement taking place.
50             wmanIfCmnHistogramIndex is the index to histogram samples.
51             Since there is no time stamp in the table,
52             wmanIfCmnHistogramIndex should be increased monotonically,
53             and warps around when it reaches the limit.
54             be maintained as FIFO to store measurement samples that

```

```

1         can be used to create RSSI and CINR histogram report.
2         when the measurement entry for a SS reaches the limit,
3         the oldest entry shall be deleted as the new entry is
4         added to the table."
5     INDEX        { ifIndex, wmanIfCmnSsidIndex,
6                   wmanIfCmnHistogramIndex }
7     ::= { wmanIfCmnSschMeasurementTable 1 }
8
9     wmanIfCmnSschMeasurementEntry ::= SEQUENCE {
10        wmanIfCmnSsidIndex           Unsigned32,
11        wmanIfCmnHistogramIndex      Unsigned32,
12        wmanIfCmnChannelNumber       INTEGER,
13        wmanIfCmnStartFrame          INTEGER,
14        wmanIfCmnDuration             INTEGER,
15        wmanIfCmnBasicReport          BITS,
16        wmanIfCmnMeanCinrReport       INTEGER,
17        wmanIfCmnStdDeviationCinrReport INTEGER,
18        wmanIfCmnMeanRssiReport       INTEGER,
19        wmanIfCmnStdDeviationRssiReport INTEGER
20    }
21
22    wmanIfCmnSsidIndex OBJECT-TYPE
23        SYNTAX      Unsigned32 (1 .. 4294967295)
24        MAX-ACCESS  read-only
25        STATUS      current
26        DESCRIPTION
27            "wmanIfCmnSsidIndex identifies the SS providing the
28             channel measurement."
29        REFERENCE
30            "Section 6.4.2.3.5 in IEEE 802.16REVd/D3-2004"
31        ::= { wmanIfCmnSschMeasurementEntry 1 }
32
33    wmanIfCmnHistogramIndex OBJECT-TYPE
34        SYNTAX      Unsigned32 (1 .. 4294967295)
35        MAX-ACCESS  read-only
36        STATUS      current
37        DESCRIPTION
38            "wmanIfCmnHistogramIndex identifies the histogram samples
39             in the table for each subscriber station."
40        ::= { wmanIfCmnSschMeasurementEntry 2 }
41
42    wmanIfCmnChannelNumber OBJECT-TYPE
43        SYNTAX      INTEGER
44        MAX-ACCESS  read-only
45        STATUS      current
46        DESCRIPTION
47            "Physical channel number to be reported on."
48        REFERENCE
49            "Section 8.5.1 in IEEE 802.16REVd/D3-2004"
50        ::= { wmanIfCmnSschMeasurementEntry 3 }
51
52    wmanIfCmnStartFrame OBJECT-TYPE
53        SYNTAX      INTEGER
54        MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "Frame number in which measurement for this channel
4          started."
5      REFERENCE
6          "Section 11.12 in IEEE 802.16REvD/D3-2004"
7      ::= { wmanIfCmnSsSchMeasurementEntry 4 }
8
9      wmanIfCmnDuration OBJECT-TYPE
10     SYNTAX      INTEGER
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "Cumulative measurement duration on the channel in
15         multiples of Ts. For any value exceeding 0xFFFFFFFF,
16         report 0xFFFFFFFF."
17     REFERENCE
18         "Section 11.12 in IEEE 802.16REvD/D3-2004"
19     ::= { wmanIfCmnSsSchMeasurementEntry 5 }
20
21     wmanIfCmnBasicReport OBJECT-TYPE
22     SYNTAX      BITS {wirelessHuman(0),
23                  unknownTransmission(1),
24                  primaryUser(2),
25                  channegNotMeasured(3)}
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "Bit #0: wirelessHUMAN detected on the channel
30         Bit #1: Unknown transmissions detected on the channel
31         Bit #2: Primary User detected on the channel
32         Bit #3: Unmeasured. Channel not measured"
33     REFERENCE
34         "Section 11.12 in IEEE 802.16REvD/D3-2004"
35     ::= { wmanIfCmnSsSchMeasurementEntry 6 }
36
37     wmanIfCmnMeanCinrReport OBJECT-TYPE
38     SYNTAX      INTEGER
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "Mean CINR report."
43     REFERENCE
44         "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
45         802.16REvD/D3-2004"
46     ::= { wmanIfCmnSsSchMeasurementEntry 7 }
47
48     wmanIfCmnStdDeviationCinrReport OBJECT-TYPE
49     SYNTAX      INTEGER
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "Standard deviation CINR report."
54     REFERENCE

```

```

1           "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
2           802.16REVd/D3-2004"
3           ::= { wmanIfCmnSsSchMeasurementEntry 8 }
4
5 wmanIfCmnMeanRssiReport OBJECT-TYPE
6     SYNTAX      INTEGER
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10            "Mean RSSI report."
11     REFERENCE
12            "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
13            802.16REVd/D3-2004"
14            ::= { wmanIfCmnSsSchMeasurementEntry 9 }
15
16 wmanIfCmnStdDeviationRssiReport OBJECT-TYPE
17     SYNTAX      INTEGER
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21            "Standard deviation RSSI report."
22     REFERENCE
23            "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
24            802.16REVd/D3-2004"
25            ::= { wmanIfCmnSsSchMeasurementEntry 10 }
26
27 -- Common PKM group
28 -- wmanIfCmnPkmObjects contain the Privacy Sublayer objects that are
29 -- common to both Base Station and Subscriber Station
30 wmanIfCmnPkmObjects OBJECT IDENTIFIER ::= { wmanIfCommonObjects 3 }
31
32 --
33 -- Table wmanIfCmnCryptoSuiteTable
34 --
35 wmanIfCmnCryptoSuiteTable OBJECT-TYPE
36     SYNTAX      SEQUENCE OF wmanIfCmnCryptoSuiteEntry
37     MAX-ACCESS  not-accessible
38     STATUS      current
39     DESCRIPTION
40            "This table describes the PKM cryptographic suite
41            capabilities for each SS or BS wireless interface."
42            ::= { wmanIfCmnPkmObjects 1 }
43
44 wmanIfCmnCryptoSuiteEntry OBJECT-TYPE
45     SYNTAX      wmanIfCmnCryptoSuiteEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49            "Each entry contains the cryptographic suite pair that SS
50            or BS supports."
51     INDEX      { ifIndex, wmanIfSsCryptoSuiteIndex }
52     ::= { wmanIfCmnCryptoSuiteTable 1 }
53
54 wmanIfCmnCryptoSuiteEntry ::= SEQUENCE {

```

```

1      wmanIfSsCryptoSuiteIndex          Integer32,
2      wmanIfCmnCryptoSuiteDataEncryptAlg  INTEGER,
3      wmanIfCmnCryptoSuiteDataAuthentAlg  INTEGER,
4      wmanIfCmnCryptoSuiteTEKEncryptAlg   INTEGER
5      }
6
7  wmanIfSsCryptoSuiteIndex OBJECT-TYPE
8      SYNTAX      Integer32 (1 .. 1000)
9      MAX-ACCESS  not-accessible
10     STATUS      current
11     DESCRIPTION
12         "The index for a cryptographic suite row."
13     ::= { wmanIfCmnCryptoSuiteEntry 1 }
14
15  wmanIfCmnCryptoSuiteDataEncryptAlg OBJECT-TYPE
16     SYNTAX      INTEGER { none(0),
17                 des56CbcMode(1) }
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21         "The value of this object is the data encryption algorithm
22         for this cryptographic suite capability."
23     REFERENCE
24         "IEEE 802.16 standard; Table 301"
25     ::= { wmanIfCmnCryptoSuiteEntry 2 }
26
27  wmanIfCmnCryptoSuiteDataAuthentAlg OBJECT-TYPE
28     SYNTAX      INTEGER { none(0) }
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32         "The value of this object is the data authentication
33         algorithm for this cryptographic suite capability."
34     REFERENCE
35         "IEEE 802.16 standard; Table 302"
36     ::= { wmanIfCmnCryptoSuiteEntry 3 }
37
38  wmanIfCmnCryptoSuiteTEKEncryptAlg OBJECT-TYPE
39     SYNTAX      INTEGER { tripleDES(0),
40                 rsa1024(1) }
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44         "The value of this object is the TEK key encryption
45         algorithm for this cryptographic suite capability."
46     REFERENCE
47         "IEEE 802.16 standard; Table 303"
48     ::= { wmanIfCmnCryptoSuiteEntry 4 }
49
50  --
51  -- wmanIfCmnOfdmPhy contain the OFDM PHY objects that are common to both
52  -- Base Station and Subscriber Station. When the objects are implemented
53  -- in the BS, they should have the read-write access. When the objects
54  -- are implemented the SS, they should have the read-only access.

```

```

1  --
2  wmanIfCmnOfdmPhy OBJECT IDENTIFIER ::= { wmanIfCommonObjects 4 }
3
4  wmanIfCmnOfdmUplinkChannelTable OBJECT-TYPE
5      SYNTAX      SEQUENCE OF wmanIfCmnOfdmUplinkChannelEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "This table contains UCD channel attributes, defining the
10         transmission characteristics of uplink channels"
11     REFERENCE
12         "Section 11.3.1, table 276 and 279, in IEEE
13         802.16REVd/D3-2004"
14     ::= { wmanIfCmnOfdmPhy 1 }
15
16  wmanIfCmnOfdmUplinkChannelEntry OBJECT-TYPE
17      SYNTAX      wmanIfCmnOfdmUplinkChannelEntry
18      MAX-ACCESS  not-accessible
19      STATUS      current
20      DESCRIPTION
21          "This table provides one row for each uplink channel of
22         multi-sector BS, and is indexed by BS ifIndex. An entry
23         in this table exists for each ifEntry of BS with an
24         ifType of propBWAmp2Mp.
25         The objects in each entry will be implemented as
26         read-create in BS and read-only in SS."
27     INDEX { ifIndex }
28     ::= { wmanIfCmnOfdmUplinkChannelTable 1 }
29
30  wmanIfCmnOfdmUplinkChannelEntry ::= SEQUENCE {
31      wmanIfCmnOfdmCtBasedResvTimeout      INTEGER,
32      wmanIfCmnOfdmBwReqOppSize            INTEGER,
33      wmanIfCmnOfdmRangReqOppSize          INTEGER,
34      wmanIfCmnOfdmUplinkCenterFreq        INTEGER,
35      wmanIfCmnOfdmSubChReqRegionFull      INTEGER,
36      wmanIfCmnOfdmSubChFocusCtCode        INTEGER,
37      wmanIfCmnOfdmUplinkChannelRowStatus  RowStatus
38  }
39
40  wmanIfCmnOfdmCtBasedResvTimeout OBJECT-TYPE
41      SYNTAX      INTEGER (1..255)
42      MAX-ACCESS  read-write
43      STATUS      current
44      DESCRIPTION
45          "The number of UL-MAPS to receive before contention-based
46         reservation is attempted again for the same connection."
47     REFERENCE
48         "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
49     ::= { wmanIfCmnOfdmUplinkChannelEntry 1 }
50
51  wmanIfCmnOfdmBwReqOppSize OBJECT-TYPE
52      SYNTAX      INTEGER (1..65535)
53      MAX-ACCESS  read-write
54      STATUS      current

```



```

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      REFERENCE
8          "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
9      ::= { wmanIfCmnOfdmUplinkChannelEntry 2 }
10
11  wmanIfCmnOfdmRangReqOppSize OBJECT-TYPE
12      SYNTAX      INTEGER (1..65535)
13      UNITS       "PS"
14      MAX-ACCESS  read-write
15      STATUS      current
16      DESCRIPTION
17          " Size (in units of PS) of PHY payload that SS may use to
18          format and transmit a RNG-REQ message in a contention
19          request opportunity. The value includes all PHY overhead
20          as well as allowance for the MAC data the message may
21          hold and the maximum SS/BS roundtrip propagation delay."
22      REFERENCE
23          "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
24      ::= { wmanIfCmnOfdmUplinkChannelEntry 3 }
25
26  wmanIfCmnOfdmUplinkCenterFreq OBJECT-TYPE
27      SYNTAX      INTEGER
28      UNITS       "KHZ"
29      MAX-ACCESS  read-write
30      STATUS      current
31      DESCRIPTION
32          " Uplink center frequency (KHz)"
33      REFERENCE
34          "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
35      ::= { wmanIfCmnOfdmUplinkChannelEntry 4 }
36
37  wmanIfCmnOfdmSubChReqRegionFull OBJECT-TYPE
38      SYNTAX      INTEGER {oneSubchannel(0),
39                        twoSubchannels(1),
40                        fourSubchannels(2),
41                        eightSubchannels(3),
42                        sixteenSubchannels(4)}
43      MAX-ACCESS  read-write
44      STATUS      current
45      DESCRIPTION
46          "Bits 0 - 2 Number of subchannels used by each transmit
47          opportunity when REQ Region-Full is allocated in
48          subchannelization region, per the following enumeration:
49          0: 1 Subchannel.
50          1: 2 Subchannels.
51          2: 4 Subchannels.
52          3: 8 Subchannels.
53          4: 16 Subchannels.
54          5-7: Shall not be used.

```

```

1           Bits 3 - 7: Number of OFDM symbols used by each transmit
2           opportunity when REQ Region-Full is allocated in
3           subchannelization region.
4   REFERENCE
5           Section 11.3.1, table 279, in IEEE 802.16REVd/D3-2004"
6   ::= { wmanIfCmnOfdmUplinkChannelEntry 5 }
7
8   wmanIfCmnOfdmSubChFocusCtCode OBJECT-TYPE
9       SYNTAX      INTEGER
10      MAX-ACCESS  read-write
11      STATUS      current
12      DESCRIPTION
13          "Number of contention codes (CSE) that shall only be used to
14          request a subchannelized allocation. Default value 0.
15          Allowed values 0-48."
16      REFERENCE
17          "Section 11.3.1, table 279, in IEEE 802.16REVd/D3-2004"
18      ::= { wmanIfCmnOfdmUplinkChannelEntry 6 }
19
20  wmanIfCmnOfdmUplinkChannelRowStatus OBJECT-TYPE
21      SYNTAX      RowStatus
22      MAX-ACCESS  read-create
23      STATUS      current
24      DESCRIPTION
25          "This object is used to create a new row or modify or
26          delete an existing row in this table.
27
28          If the implementator of this MIB has chosen not
29          to implement 'dynamic assignment' of profiles, this
30          object is not useful and should return noSuchName
31          upon SNMP request."
32      ::= { wmanIfCmnOfdmUplinkChannelEntry 7 }
33
34  wmanIfCmnOfdmDownlinkChannelTable OBJECT-TYPE
35      SYNTAX      SEQUENCE OF wmanIfCmnOfdmDownlinkChannelEntry
36      MAX-ACCESS  not-accessible
37      STATUS      current
38      DESCRIPTION
39          "This table contains DCD channel attributes, defining the
40          transmission characteristics of downlink channels"
41      REFERENCE
42          "Section 11.4.1, Table 286, in IEEE 802.16REVd/D3-2004"
43      ::= { wmanIfCmnOfdmPhy 2 }
44
45  wmanIfCmnOfdmDownlinkChannelEntry OBJECT-TYPE
46      SYNTAX      wmanIfCmnOfdmDownlinkChannelEntry
47      MAX-ACCESS  not-accessible
48      STATUS      current
49      DESCRIPTION
50          "This table provides one row for each downlink channel of
51          multi-sector BS, and is indexed by BS ifIndex. An entry
52          in this table exists for each ifEntry of BS with an
53          ifType of propBWA2Mp.
54          The objects in each entry will be implemented as

```

```

1         read-create in BS and read-only in SS."
2     INDEX { ifIndex }
3     ::= { wmanIfCmnOfdmDownlinkChannelTable 1 }
4
5     wmanIfCmnOfdmDownlinkChannelEntry ::= SEQUENCE {
6         wmanIfCmnOfdmBSEIRP                INTEGER,
7         wmanIfCmnOfdmChannelNumber          INTEGER,
8         wmanIfCmnOfdmTTG                    INTEGER,
9         wmanIfCmnOfdmRTG                    INTEGER,
10        wmanIfCmnOfdmInitRngMaxRSS          INTEGER,
11        wmanIfCmnOfdmChSwitchFrameNmr      INTEGER,
12        wmanIfCmnOfdmDownlinkCenterFreq    INTEGER,
13        wmanIfCmnOfdmBsId                   INTEGER,
14        wmanIfCmnOfdmMacVersion             INTEGER,
15        wmanIfCmnOfdmFrameDurationCode     INTEGER,
16        wmanIfCmnOfdmFrameNumber           INTEGER,
17        wmanIfCmnOfdmDownlinkChannelRowStatus RowStatus
18    }
19
20    wmanIfCmnOfdmBSEIRP OBJECT-TYPE
21        SYNTAX      INTEGER
22        UNITS       "dbM"
23        MAX-ACCESS  read-write
24        STATUS      current
25        DESCRIPTION
26            " Signed in units of 1 dBm."
27        REFERENCE
28            "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
29        ::= { wmanIfCmnOfdmDownlinkChannelEntry 1 }
30
31    wmanIfCmnOfdmChannelNumber OBJECT-TYPE
32        SYNTAX      INTEGER
33        MAX-ACCESS  read-write
34        STATUS      current
35        DESCRIPTION
36            " Downlink channel number as defined in 8.5.
37            Used for license-exempt operation only."
38        REFERENCE
39            "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
40        ::= { wmanIfCmnOfdmDownlinkChannelEntry 2 }
41
42    wmanIfCmnOfdmTTG OBJECT-TYPE
43        SYNTAX      INTEGER
44        MAX-ACCESS  read-write
45        STATUS      current
46        DESCRIPTION
47            " Transmit / Receive Transition Gap."
48        REFERENCE
49            "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
50        ::= { wmanIfCmnOfdmDownlinkChannelEntry 3 }
51
52    wmanIfCmnOfdmRTG OBJECT-TYPE
53        SYNTAX      INTEGER
54        MAX-ACCESS  read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          " Receive / Transmit Transition Gap."
4      REFERENCE
5          "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
6          ::= { wmanIfCmnOfdmDownlinkChannelEntry 4 }
7
8      wmanIfCmnOfdmInitRngMaxRSS OBJECT-TYPE
9          SYNTAX      INTEGER
10         UNITS        "dbm"
11         MAX-ACCESS  read-write
12         STATUS      current
13         DESCRIPTION
14             " Initial Ranging Max. Received Signal Strength at BS
15             Signed in units of 1 dBm."
16         REFERENCE
17             "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
18             ::= { wmanIfCmnOfdmDownlinkChannelEntry 5 }
19
20     wmanIfCmnOfdmChSwitchFrameNmr OBJECT-TYPE
21         SYNTAX      INTEGER
22         MAX-ACCESS  read-write
23         STATUS      current
24         DESCRIPTION
25             " Channel switch frame number as defined in 6.4.14.7,
26             Used for license-exempt operation only."
27         REFERENCE
28             "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
29             ::= { wmanIfCmnOfdmDownlinkChannelEntry 6 }
30
31     wmanIfCmnOfdmDownlinkCenterFreq OBJECT-TYPE
32         SYNTAX      INTEGER
33         UNITS        "KHz"
34         MAX-ACCESS  read-write
35         STATUS      current
36         DESCRIPTION
37             " Downlink center frequency (kHz)."
38         REFERENCE
39             "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
40             ::= { wmanIfCmnOfdmDownlinkChannelEntry 7 }
41
42     wmanIfCmnOfdmBsId OBJECT-TYPE
43         SYNTAX      INTEGER
44         MAX-ACCESS  read-write
45         STATUS      current
46         DESCRIPTION
47             " Base station ID."
48         REFERENCE
49             "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
50             ::= { wmanIfCmnOfdmDownlinkChannelEntry 8 }
51
52     wmanIfCmnOfdmMacVersion OBJECT-TYPE
53         SYNTAX      INTEGER
54         MAX-ACCESS  read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          " This parameter specifies the version of 802.16 to which
4          the message originator conforms."
5      REFERENCE
6          "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
7      ::= { wmanIfCmnOfdmDownlinkChannelEntry 9 }
8
9      wmanIfCmnOfdmFrameDurationCode OBJECT-TYPE
10     SYNTAX      INTEGER
11     MAX-ACCESS  read-write
12     STATUS      current
13     DESCRIPTION
14         " The duration of the frame. The frame duration code
15         values are specified in Table 211."
16     REFERENCE
17         "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
18     ::= { wmanIfCmnOfdmDownlinkChannelEntry 10 }
19
20     wmanIfCmnOfdmFrameNumber OBJECT-TYPE
21     SYNTAX      INTEGER
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         " The number of frame containing the DCD message."
26     REFERENCE
27         "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
28     ::= { wmanIfCmnOfdmDownlinkChannelEntry 11 }
29
30     wmanIfCmnOfdmDownlinkChannelRowStatus OBJECT-TYPE
31     SYNTAX      RowStatus
32     MAX-ACCESS  read-create
33     STATUS      current
34     DESCRIPTION
35         "This object is used to create a new row or modify or
36         delete an existing row in this table.
37
38         If the implementator of this MIB has chosen not
39         to implement 'dynamic assignment' of profiles, this
40         object is not useful and should return noSuchName
41         upon SNMP request."
42     ::= { wmanIfCmnOfdmDownlinkChannelEntry 12 }
43
44     wmanIfCmnOfdmUcdBurstProfileTable OBJECT-TYPE
45     SYNTAX      SEQUENCE OF wmanIfCmnOfdmUcdBurstProfileEntry
46     MAX-ACCESS  not-accessible
47     STATUS      current
48     DESCRIPTION
49         "This table contains UCD burst profiles for each uplink
50         channel"
51     REFERENCE
52         "Section 11.3.1.1, table 281 and 284, in IEEE
53         802.16REVd/D3-2004"
54     ::= { wmanIfCmnOfdmPhy 3 }

```

```

1
2  wmanIfCmnOfdmUcdBurstProfileEntry OBJECT-TYPE
3      SYNTAX      wmanIfCmnOfdmUcdBurstProfileEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "This table provides one row for each UCD burst profile.
8          This table is double indexed. The primary index is an
9          ifIndex with an ifType of propBWAp2Mp. The secondary index
10         is wmanIfCmnOfdmOfdmUcdBurstProfIndex.
11         The objects in each entry will be implemented as
12         read-create in BS and read-only in SS."
13     INDEX { ifIndex, wmanIfCmnOfdmOfdmUcdBurstProfIndex }
14     ::= { wmanIfCmnOfdmUcdBurstProfileTable 1 }
15
16  wmanIfCmnOfdmUcdBurstProfileEntry ::= SEQUENCE {
17      wmanIfCmnOfdmOfdmUcdBurstProfIndex      INTEGER,
18      wmanIfCmnOfdmUplinkFrequency           INTEGER,
19      wmanIfCmnOfdmUcdFecCodeType           INTEGER,
20      wmanIfCmnOfdmFocusCtPowerBoost        INTEGER,
21      wmanIfCmnOfdmUcdBurstProfileRowStatus RowStatus
22  }
23
24  wmanIfCmnOfdmOfdmUcdBurstProfIndex OBJECT-TYPE
25      SYNTAX      INTEGER (1 .. 100)
26      MAX-ACCESS  not-accessible
27      STATUS      current
28      DESCRIPTION
29          "ifIndex and wmanIfCmnOfdmOfdmUcdBurstProfIndex uniquely
30          identify an entry in the wmanIfCmnOfdmUcdBurstProfileTable."
31      ::= { wmanIfCmnOfdmUcdBurstProfileEntry 1 }
32
33  wmanIfCmnOfdmUplinkFrequency OBJECT-TYPE
34      SYNTAX      INTEGER
35      UNITS       "KHZ"
36      MAX-ACCESS  read-write
37      STATUS      current
38      DESCRIPTION
39          "Uplink Frequency (kHz)."

```

```

1          sixtyFourQamBtc2-3(10),
2          sixtyFourQamBtc5-6(11),
3          qpskCtc1-2(12),
4          qpskCtc2-3(13),
5          qpskCtc3-4(14),
6          sixteenQamCtc3-4(16),
7          sixteenQamCtc2-3(17),
8          sixtyFourQamCtc3-4(18)}
9  MAX-ACCESS  read-write
10 STATUS      current
11 DESCRIPTION
12     " 0= QPSK (RS+CC/CC) 1/2
13       1= QPSK (RS+CC/Cc) 3/4
14       2= 16-QAM (RS+CC/CC) 1/2
15       3= 16-QAM (RS+CC/CC) 3/4
16       4= 64-QAM (RS+CC/CC) 2/3
17       5= 64-QAM (RS+CC/CC) 3/4
18       6= QPSK (BTC) 1/2
19       7= QPSK (BTC) 3/4
20       8= 16-QAM (BTC) 3/5
21       9= 16-QAM (BTC) 4/5
22       10 = 64-QAM (BTC) 2/3
23       11 = 64-QAM (BTC) 5/6
24       12 = QPSK (CTC) 1/2
25       13 = QPSK (CTC) 2/3
26       14 = QPSK (CTC) 3/4
27       15 = 16-QAM (CTC) 1/2
28       16 = 16-QAM (CTC) 3/4
29       17 = 64-QAM (CTC) 2/3
30       18 = 64-QAM (CTC) 3/4
31       19 - 255 Reserved."
32 REFERENCE
33     "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
34 ::= { wmanIfCmnOfdmUcdBurstProfileEntry 3 }
35
36 wmanIfCmnOfdmFocusCtPowerBoost OBJECT-TYPE
37     SYNTAX      INTEGER
38     MAX-ACCESS  read-write
39     STATUS      current
40     DESCRIPTION
41         "The power boost in dB of focused contention carriers, as
42         described in 8.3.6.3.3."
43     REFERENCE
44         "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
45     ::= { wmanIfCmnOfdmUcdBurstProfileEntry 4 }
46
47 wmanIfCmnOfdmUcdBurstProfileRowStatus OBJECT-TYPE
48     SYNTAX      RowStatus
49     MAX-ACCESS  read-create
50     STATUS      current
51     DESCRIPTION
52         "This object is used to create a new row or modify or
53         delete an existing row in this table.
54

```

```

1         If the implementator of this MIB has chosen not
2         to implement 'dynamic assignment' of profiles, this
3         object is not useful and should return noSuchName
4         upon SNMP request."
5         ::= { wmanIfCmnOfdmUcdBurstProfileEntry 5 }
6
7     wmanIfCmnOfdmDcdBurstProfileTable OBJECT-TYPE
8         SYNTAX      SEQUENCE OF wmanIfOfdmDcdBurstProfileEntry
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 propBWAp2Mp. The secondary
15            index is wmanIfCmnOfdmOfdmDcdBurstProfIndex"
16        ::= { wmanIfCmnOfdmPhy 4 }
17
18
19    wmanIfCmnOfdmDcdBurstProfileEntry OBJECT-TYPE
20        SYNTAX      wmanIfOfdmDcdBurstProfileEntry
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 propBWAp2Mp. The secondary index
27            is wmanIfCmnOfdmDcdBurstProfIndex.
28            The objects in each entry will be implemented as
29            read-create in BS and read-only in SS."
30        INDEX { ifIndex, wmanIfCmnOfdmDcdBurstProfIndex }
31        ::= { wmanIfCmnOfdmDcdBurstProfileTable 1 }
32
33    wmanIfOfdmDcdBurstProfileEntry ::= SEQUENCE {
34        wmanIfCmnOfdmDcdBurstProfIndex      INTEGER,
35        wmanIfCmnOfdmDownlinkFrequency     INTEGER,
36        wmanIfCmnOfdmDcdFecCodeType       INTEGER,
37        wmanIfCmnOfdmDiucMandatoryExitThresh  INTEGER,
38        wmanIfCmnOfdmDiucMinEntryThresh     INTEGER,
39        wmanIfCmnOfdmDcdBurstProfileRowStatus  RowStatus
40    }
41
42    wmanIfCmnOfdmDcdBurstProfIndex OBJECT-TYPE
43        SYNTAX      INTEGER (1 .. 100)
44        MAX-ACCESS  not-accessible
45        STATUS      current
46        DESCRIPTION
47            "ifIndex and wmanIfCmnOfdmDcdBurstProfIndex uniquely
48            identify an entry in the wmanIfCmnOfdmDcdBurstProfileTable."
49        ::= { wmanIfCmnOfdmDcdBurstProfileEntry 1 }
50
51    wmanIfCmnOfdmDownlinkFrequency OBJECT-TYPE
52        SYNTAX      INTEGER
53        UNITS       "KHZ"
54        MAX-ACCESS  read-write

```



```

1      STATUS      current
2      DESCRIPTION
3          "Downlink Frequency (kHz)."
```

REFERENCE

```

5          "Section 11.4.1, table 287, in IEEE 802.16REVd/D3-2004"
6      ::= { wmanIfCmnOfdmDcdBurstProfileEntry 2 }
7
8      wmanIfCmnOfdmDcdFecCodeType OBJECT-TYPE
9          SYNTAX      INTEGER {qpskRsCc1-2(0),
10                      qpskRsCc3-4(1),
11                      sixteenQamRsCc1-2(2),
12                      sixteenQamRsCc3-4(3),
13                      sixtyFourQamRsCc2-3(4),
14                      sixtyFourQamRsCc3-4(5),
15                      qpskBtc1-2(6),
16                      qpskBtc3-4(7),
17                      sixteenQamBtc3-4(8),
18                      sixteenQamBtc4-5(9),
19                      sixtyFourQamBtc2-3or5-8(10),
20                      sixtyFourQamBtc5-6or4-5(11),
21                      qpskCtc1-2(12),
22                      qpskCtc2-3(13),
23                      qpskCtc3-4(14),
24                      sixteenQamCtc1-2(16),
25                      sixteenQamCtc3-4(17),
26                      sixtyFourQamCtc3-4(18)}
27
28      MAX-ACCESS    read-write
29      STATUS      current
30      DESCRIPTION
31          " 0= QPSK (RS+CC) 1/2
32          1= QPSK (RS+CC) 3/4
33          2= 16-QAM (RS+CC) 1/2
34          3= 16-QAM (RS+CC) 3/4
35          4= 64-QAM (RS+CC) 2/3
36          5= 64-QAM (RS+CC) 3/4
37          6= QPSK (BTC) 1/2
38          7= QPSK (BTC) 3/4
39          8= 16-QAM (BTC) 3/5
40          9= 16-QAM (BTC) 4/5
41          10 = 64-QAM (BTC) 2/3 or 5/8
42          11 = 64-QAM (BTC) 5/6 or 4/5
43          12 = QPSK (CTC) 1/2
44          13 = QPSK (CTC) 2/3
45          14 = QPSK (CTC) 3/4
46          15 = 16-QAM (CTC) 1/2
47          16 = 16-QAM (CTC) 3/4
48          17 = 64-QAM (CTC) 2/3
49          18 = 64-QAM (CTC) 3/4
50          19 - 255 Reserved."
51      REFERENCE
52          "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
53      ::= { wmanIfCmnOfdmDcdBurstProfileEntry 3 }
54
55      wmanIfCmnOfdmDiucMandatoryExitThresh OBJECT-TYPE
```

```

1      SYNTAX      INTEGER
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "DIUC mandatory exit threshold: 0 - 63.75 dB CINR at or
6          below where this DIUC can no longer be used and where this
7          change to a more robust DIUC is required, in 0.25 dB units."
8      REFERENCE
9          "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
10     ::= { wmanIfCmnOfdmDcdBurstProfileEntry 4 }
11
12     wmanIfCmnOfdmDiucMinEntryThresh OBJECT-TYPE
13         SYNTAX      INTEGER
14         MAX-ACCESS  read-write
15         STATUS      current
16         DESCRIPTION
17             "DIUC minimum entry threshold: 0 - 63.75 dB The minimum CINR
18             required to start using this DIUC when changing from a more
19             robust DIUC is required, in 0.25 dB units."
20         REFERENCE
21             "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
22         ::= { wmanIfCmnOfdmDcdBurstProfileEntry 5 }
23
24     wmanIfCmnOfdmDcdBurstProfileRowStatus OBJECT-TYPE
25         SYNTAX      RowStatus
26         MAX-ACCESS  read-create
27         STATUS      current
28         DESCRIPTION
29             "This object is used to create a new row or modify or
30             delete an existing row in this table.
31
32             If the implementator of this MIB has chosen not
33             to implement 'dynamic assignment' of profiles, this
34             object is not useful and should return noSuchName
35             upon SNMP request."
36         ::= { wmanIfCmnOfdmDcdBurstProfileEntry 6 }
37
38
39     END
40
41
42

```

