

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	2004-07-09	
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]
	Patrice Desmoulin France Telecom	[mailto:patrice.desmoulin@francetelecom.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	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."	
	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 (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.	<i>Introduction</i>	5
1.1	Scope.....	5
1.2	References.....	5
2.	<i>BWA Network Management Reference Model</i>	6
3.	<i>Relationship with Interface MIB</i>	7
3.1	MIB-2 Integration	7
3.2	Usage of MIB-II Tables	7
3.3	Events and Traps.....	8
4.	<i>802.16 MIB Structure</i>	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), managed nodes, and Service Flow Database. Bs and SS managed nodes collect and store the managed objects in the format of 802.16 MIB that are made available to NMSs using SNMP (Simple Network Management Protocol). Service Flow Database contains the service flow and the associated QoS information that have to be populated to BS and SS when the service is provisioned or a mobile SS roams into BS coverage. SSs can be managed directly from NMS, or indirectly through BS, acting as the SNMP proxy.

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.

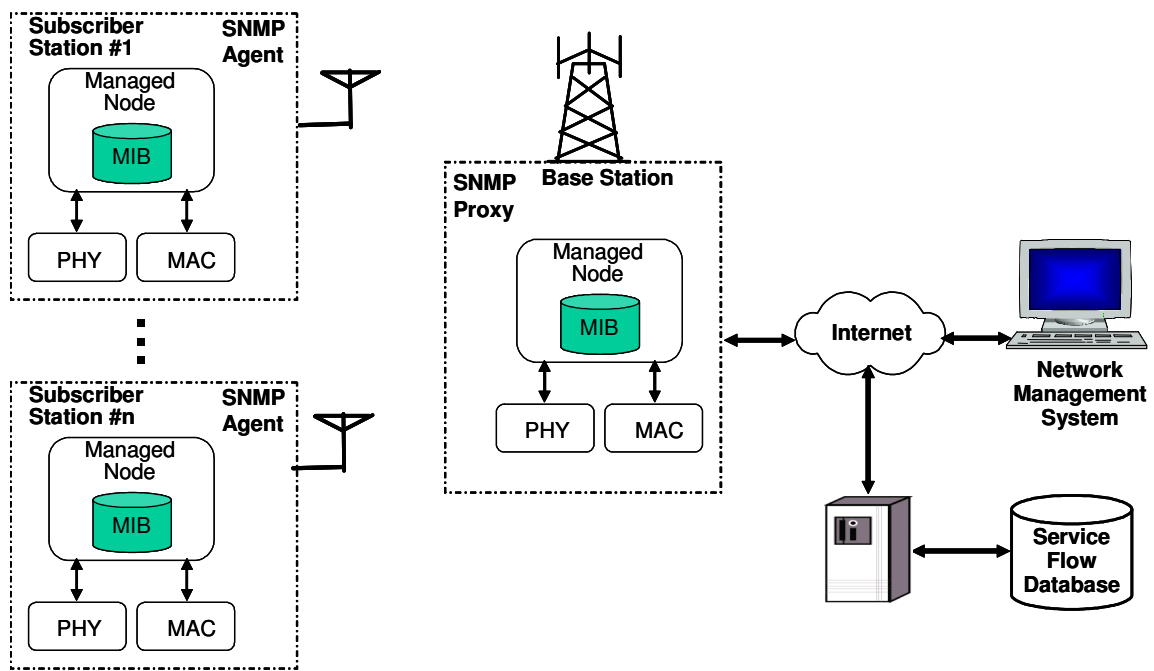


Figure 1 - BWA Network Management Reference Model

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

<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)	propBWAp2Mp	Null	MAC address of BS sector	Administration Status	Operational Status
BS Sector 2	An ifEntry per BS sector (2)	propBWAp2Mp	Null	MAC address of BS sector	Administration Status	Operational Status
BS Sector 3	An ifEntry per BS sector (3)	propBWAp2Mp	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	propBWAp2Mp	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

3.3 Events and Traps

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

```

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

```


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].

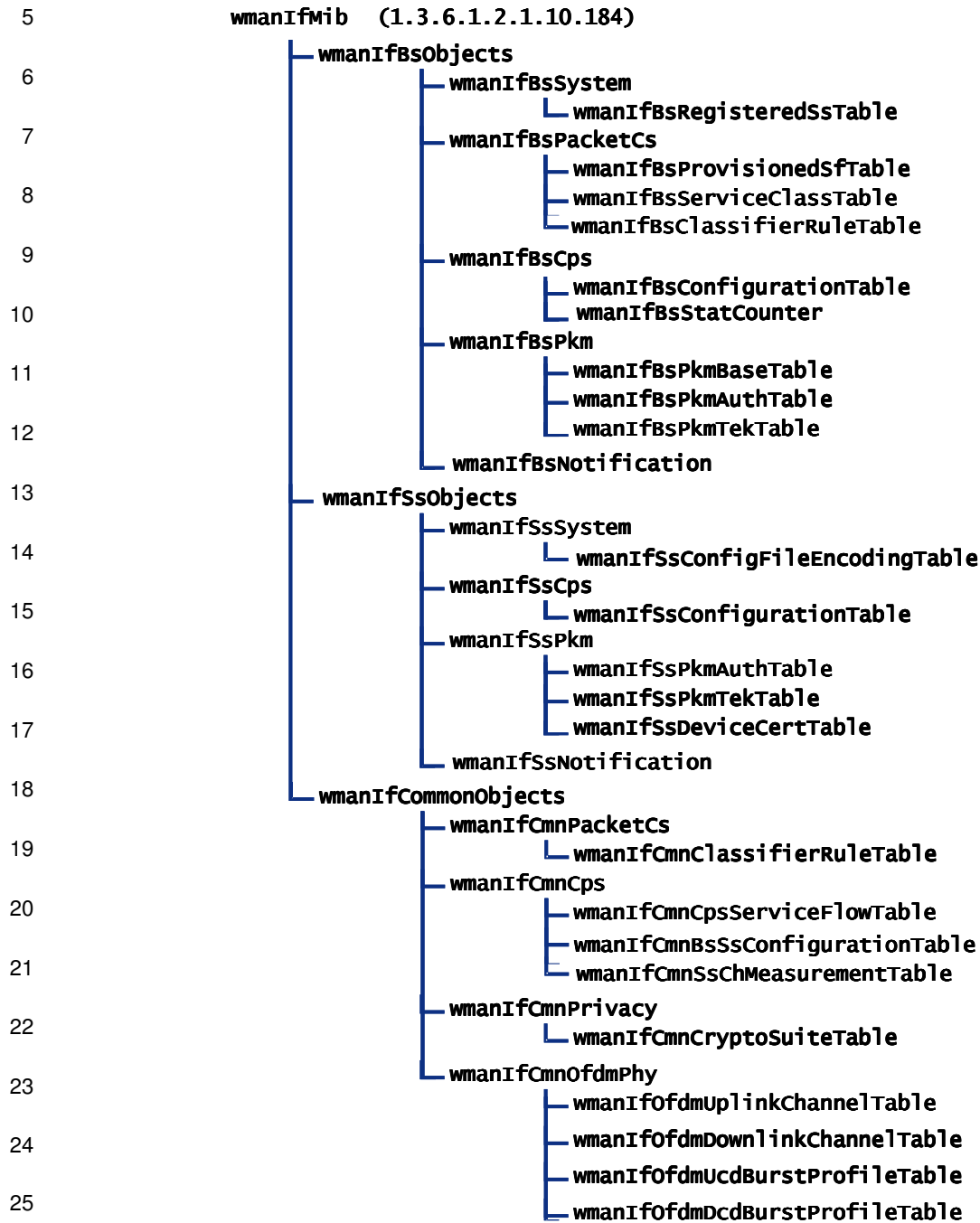


Figure 2 – wmanIfMib Structure

- 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.

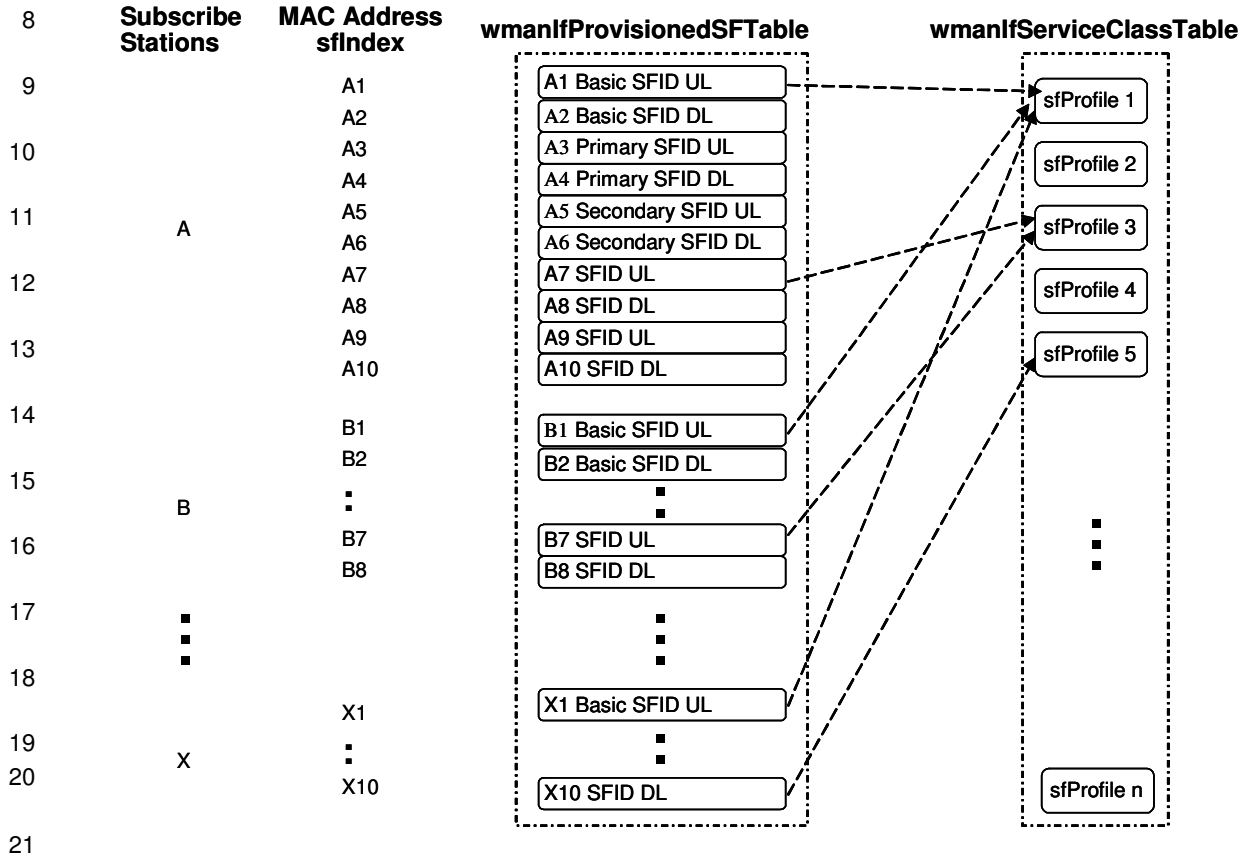


Figure 3 – Service Classes – Service Flows Mapping

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 power status, RSSI threshold crossing.
- 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 power status, RSSI threshold crossing.

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      "0407090000Z" -- July 09, 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 propBWA2Mp. 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      wmanIfBsSsDsxFlowControl    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     wmanIfBsPowerStatus          INTEGER,
18     wmanIfBsFanStatus            INTEGER,
19     wmanIfBsTemperatureStatus    INTEGER,
20     wmanIfBsPowerStatusInfo      OCTET STRING,
21     wmanIfBsFanStatusInfo        OCTET STRING,
22     wmanIfBsTemperatureStatusInfo OCTET STRING
23     }
24
25     wmanIfBsSsIdIndex OBJECT-TYPE
26         SYNTAX      Unsigned32 (1 .. 4294967295)
27         MAX-ACCESS  read-only
28         STATUS      current
29         DESCRIPTION
30             "wmanIfBsSsIdIndex identifies the SS that is registered."
31         REFERENCE
32             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
33         ::= { wmanIfBsRegisteredSsEntry 1 }
34
35     wmanIfBsSsMacAddress OBJECT-TYPE
36         SYNTAX      MacAddress
37         MAX-ACCESS  read-only
38         STATUS      current
39         DESCRIPTION
40             "The MAC address of SS is received from the RNG-REQ
41             message. This MAC address can be used as the
42             index to find out the BS and its associated Ss."
43         REFERENCE
44             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
45         ::= { wmanIfBsRegisteredSsEntry 2 }
46
47     wmanIfBsSsBasicCid OBJECT-TYPE
48         SYNTAX      INTEGER
49         MAX-ACCESS  read-only
50         STATUS      current
51         DESCRIPTION
52             "The value of this object indicates the SS's basic CID
53             that was sent in the RNG-RSP message."
54         REFERENCE

```

```

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

```

```

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

```

```

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

```

```

1         "Section 11.7.5.1 in IEEE 802.16REvd/D3-2004"
2         ::= { wmanIfBsRegisteredSsEntry 16 }
3
4     wmanIfBsSSMaxNumOfClassifier OBJECT-TYPE
5         SYNTAX      INTEGER
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "This object indicates the maximum number of admitted
10            Classifiers that the SS is allowed to have."
11        REFERENCE
12            "Section 11.7.5.2 in IEEE 802.16REvd/D3-2004"
13        ::= { wmanIfBsRegisteredSsEntry 17 }
14
15    wmanIfBsSSPhsSupport OBJECT-TYPE
16        SYNTAX      INTEGER {noPhsSupport(0),
17                            atmPhsSupport(1),
18                            packetPhsSupport(2)}
19        MAX-ACCESS  read-only
20        STATUS      current
21        DESCRIPTION
22            "This object indicates indicates the level of PHS support."
23        REFERENCE
24            "Section 11.7.5.3 in IEEE 802.16REvd/D3-2004"
25        ::= { wmanIfBsRegisteredSsEntry 18 }
26
27    wmanIfBsPowerStatus OBJECT-TYPE
28        SYNTAX      INTEGER {priOnSecStandby(0),
29                            secOnPriStandby(1),
30                            priOnSecFailed(2),
31                            secOnPriFailed(3)
32                            }
33        MAX-ACCESS  read-only
34        STATUS      current
35        DESCRIPTION
36            "Describes the status of the power supply in BS."
37        ::= { wmanIfBsRegisteredSsEntry 19 }
38
39    wmanIfBsFanStatus OBJECT-TYPE
40        SYNTAX      INTEGER {fanFail(1),
41                            fanSucc(2)
42                            }
43        MAX-ACCESS  read-only
44        STATUS      current
45        DESCRIPTION
46            "Describes the status of the fan in BS."
47        ::= { wmanIfBsRegisteredSsEntry 20 }
48
49    wmanIfBsTemperatureStatus OBJECT-TYPE
50        SYNTAX      INTEGER {lowTempReached(1),
51                            highTempReached(2),
52                            temperatureNormal(3)
53                            }
54        MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "Describes the status of the temperature in BS."
4      ::= { wmanIfBsRegisteredSsEntry 21 }
5
6      wmanIfBsPowerStatusInfo OBJECT-TYPE
7          SYNTAX      OCTET STRING
8          MAX-ACCESS  read-write
9          STATUS      current
10         DESCRIPTION
11             "Display the power supply ststus in text form."
12         ::= { wmanIfBsRegisteredSsEntry 22 }
13
14         wmanIfBsFanStatusInfo OBJECT-TYPE
15             SYNTAX      OCTET STRING
16             MAX-ACCESS  read-write
17             STATUS      current
18             DESCRIPTION
19                 "Display the fan ststus in text form."
20             ::= { wmanIfBsRegisteredSsEntry 23 }
21
22         wmanIfBsTemperatureStatusInfo OBJECT-TYPE
23             SYNTAX      OCTET STRING
24             MAX-ACCESS  read-write
25             STATUS      current
26             DESCRIPTION
27                 "Display the temperature ststus in text form."
28             ::= { wmanIfBsRegisteredSsEntry 24 }
29
30         --
31         -- wmanIfBsPacketCs contain the Base Station Packet Convergence Sublayer
32         -- objects
33         wmanIfBsPacketCs OBJECT IDENTIFIER ::= { wmanIfBsObjects 2 }
34
35         wmanSfsSchedulingType ::= TEXTUAL-CONVENTION
36             STATUS      current
37             DESCRIPTION
38                 "The scheduling service provided by a SC for an
39                 upstream service flow. If the parameter is omitted
40                 from an upstream QoS Parameter Set, this object takes
41                 the value of bestEffort (2). This parameter must be
42                 reported as undefined (1) for downstream QoS Parameter
43                 Sets."
44             SYNTAX      INTEGER {undefined(1),
45                             bestEffort(2),
46                             nonRealTimePollingService(3),
47                             realTimePollingService(4),
48                             unsolicitedGrantService(6)}
49
50         wmanIfBsProvisionedSfTable OBJECT-TYPE
51             SYNTAX      SEQUENCE OF wmanIfBsProvisionedSfEntry
52             MAX-ACCESS  not-accessible
53             STATUS      current
54             DESCRIPTION

```

```

1           "This table is doubly indexed (SS MAC address, SF ID) and
2           contains pre-provisioned service flow profiles, Per SS.
3           These connection parameters shall be provisioned for the SS
4           using DSA messages. NMS shall pre-provisioning the service
5           class table - wmanIfBsServiceClasTable by using
6           wmanIfBsServiceClassIndex, and packet classifier rule table
7           - wmanIfBsClassifierRuleTable by using wmanIfBsSfId"
8   REFERENCE
9           "Section 6.4.13 in IEEE 802.16REVD/D3-2004"
10          ::= { wmanIfBsPacketCs 1 }
11
12   wmanIfBsProvisionedSfEntry OBJECT-TYPE
13       SYNTAX      WmanIfBsProvisionedSfEntry
14       MAX-ACCESS  not-accessible
15       STATUS      current
16       DESCRIPTION
17           "This table provides one row for each service flow been
18           pre-provisioned by NMS."
19       INDEX { wmanIfBsSsProvMacAddress, wmanIfBsSfId}
20       ::= { wmanIfBsProvisionedSfTable 1 }
21
22   WmanIfBsProvisionedSfEntry ::= SEQUENCE {
23       wmanIfBsSfId                Unsigned32,
24       wmanIfBsSsProvMacAddress    MacAddress,
25       wmanIfBsSfDirection        INTEGER,
26       wmanIfBsServiceClassIndex  INTEGER,
27       wmanIfBsSfState            INTEGER,
28       wmanIfBsSfProvisionedTime  TimeStamp,
29       wmanIfBsProvisionedSfRowStatus RowStatus
30   }
31
32   wmanIfBsSfId OBJECT-TYPE
33       SYNTAX      Unsigned32 (1 .. 4294967295)
34       MAX-ACCESS  not-accessible
35       STATUS      current
36       DESCRIPTION
37           "A 32 bit quantity that uniquely identifies a service flow
38           to both the subscriber station and base station (BS)."
39       ::= { wmanIfBsProvisionedSfEntry 1 }
40
41   wmanIfBsSsProvMacAddress OBJECT-TYPE
42       SYNTAX      MacAddress
43       MAX-ACCESS  not-accessible
44       STATUS      current
45       DESCRIPTION
46           "The MAC address of the SS, where the service flow resides.
47           It can be used as the index to associate service flows
48           with the SS."
49       ::= { wmanIfBsProvisionedSfEntry 2 }
50
51   wmanIfBsSfDirection OBJECT-TYPE
52       SYNTAX      INTEGER {downstream(1),
53                       upstream(2)}
54       MAX-ACCESS  read-create

```



```

1      STATUS      current
2      DESCRIPTION
3          "An attribute indicating the service flow is downstream or
4          upstream."
5      ::= { wmanIfBsProvisionedSfEntry 3 }
6
7      wmanIfBsServiceClassIndex OBJECT-TYPE
8          SYNTAX      INTEGER
9          MAX-ACCESS  read-create
10         STATUS      current
11         DESCRIPTION
12             "The index in wmanIfBsServiceClasTable describing the
13             service class or QoS parameters for such service flow.
14             If no associated entry in wmanIfBsServiceClasTable
15             exists, this object returns a value of zero."
16         ::= { wmanIfBsProvisionedSfEntry 4 }
17
18         wmanIfBsSfState OBJECT-TYPE
19             SYNTAX      INTEGER {provisionedState(1),
20                             admittedState(2),
21                             activeState(3)}
22             MAX-ACCESS  read-create
23             STATUS      current
24             DESCRIPTION
25                 "wmanIfBsSfState determines how the service flow will be
26                 transitioned to the Admitted or Active state.
27                 Admitted or Active state: The pre-provisioned service flow
28                 will be transitioned to the Admitted or Active state, as
29                 soon as the SS passes the network entry procedure, and
30                 connection admission control. An entry will be created
31                 in the SS and BS service flow tables.
32                 Provisioned state: After SS enters the network; the
33                 pre-provisioned service flow will remain in the Provisioned
34                 state until NMS set it different state. An entry will be
35                 created in the SS and BS service flow tables"
36             REFERENCE
37                 "Section 6.4.13.6, in IEEE 802.16REvD/D3-2004"
38             ::= { wmanIfBsProvisionedSfEntry 5 }
39
40         wmanIfBsSfProvisionedTime OBJECT-TYPE
41             SYNTAX      TimeStamp
42             MAX-ACCESS  read-create
43             STATUS      current
44             DESCRIPTION
45                 "Indicates the data and time when the service flow is
46                 provisioned."
47             ::= { wmanIfBsProvisionedSfEntry 6 }
48
49         wmanIfBsProvisionedSfRowStatus OBJECT-TYPE
50             SYNTAX      RowStatus
51             MAX-ACCESS  read-create
52             STATUS      current
53             DESCRIPTION
54                 "This object is used to create a new row or modify or

```

```

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

```

```

1
2  wmanIfBsQoSProfileIndex OBJECT-TYPE
3      SYNTAX      INTEGER (1 .. 1000)
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "The index value which uniquely identifies an entry
8           in the wmanIfBsServiceClassTable"
9      ::= { wmanIfBsServiceClassEntry 1 }
10
11 wmanIfBsQoSServiceClassName OBJECT-TYPE
12     SYNTAX      DisplayString
13     MAX-ACCESS  read-create
14     STATUS      current
15     DESCRIPTION "Refers to the Service Class Name"
16     REFERENCE
17         "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
18     ::= { wmanIfBsServiceClassEntry 2 }
19
20 wmanIfBsQoSTrafficPriority OBJECT-TYPE
21     SYNTAX      INTEGER (0..7)
22     MAX-ACCESS  read-create
23     STATUS      current
24     DESCRIPTION
25         "The value of this parameter specifies the priority
26          assigned to a service flow. For uplink service flows,
27          the BS should use this parameter when determining
28          precedence in request service and grant generation,
29          and the SS shall preferentially select contention
30          Request opportunities for Priority Request CIDs
31          based on this priority. Higher numbers indicate higher
32          priority"
33     REFERENCE
34         "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
35     ::= { wmanIfBsServiceClassEntry 3 }
36
37 wmanIfBsQoSMaxSustainedRate OBJECT-TYPE
38     SYNTAX      INTEGER
39     UNITS       "bps"
40     MAX-ACCESS  read-create
41     STATUS      current
42     DESCRIPTION
43         "This parameter defines the peak information rate
44          of the service. The rate is expressed in bits per
45          second and pertains to the SDUs at the input to
46          the system."
47     REFERENCE
48         "Section 11.13.8 in IEEE 802.16REVd/D3-2004"
49     ::= { wmanIfBsServiceClassEntry 4 }
50
51 wmanIfBsQoSMaxTrafficBurst OBJECT-TYPE
52     SYNTAX      INTEGER
53     MAX-ACCESS  read-create
54     STATUS      current

```

```

1      DESCRIPTION
2          "This parameter defines the maximum burst size that
3          must be accommodated for the service."
4      REFERENCE
5          "Section 11.13.9 in IEEE 802.16REVd/D3-2004"
6      ::= { wmanIfBsServiceClassEntry 5 }
7
8      wmanIfBsQoSMinReservedRate OBJECT-TYPE
9          SYNTAX      INTEGER
10         UNITS       "bps"
11         MAX-ACCESS  read-create
12         STATUS      current
13         DESCRIPTION
14             "This parameter specifies the minimum rate reserved
15             for this service flow."
16         REFERENCE
17             "Section 11.13.10 in IEEE 802.16REVd/D3-2004"
18         ::= { wmanIfBsServiceClassEntry 6 }
19
20     wmanIfBsQoSToLeratedJitter OBJECT-TYPE
21         SYNTAX      INTEGER
22         UNITS       "millisecond"
23         MAX-ACCESS  read-create
24         STATUS      current
25         DESCRIPTION
26             "This parameter defines the Maximum delay
27             variation (jitter) for the connection."
28         REFERENCE
29             "Section 11.13.15 in IEEE 802.16REVd/D3-2004"
30         ::= { wmanIfBsServiceClassEntry 7 }
31
32     wmanIfBsQoSMaxLatency OBJECT-TYPE
33         SYNTAX      INTEGER
34         UNITS       "millisecond"
35         MAX-ACCESS  read-create
36         STATUS      current
37         DESCRIPTION
38             "The value of this parameter specifies the maximum
39             latency between the reception of a packet by the BS
40             or SS on its network interface and the forwarding
41             of the packet to its RF Interface."
42         REFERENCE
43             "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
44         ::= { wmanIfBsServiceClassEntry 8 }
45
46     wmanIfBsQoSFixedVsVariableSduInd OBJECT-TYPE
47         SYNTAX      INTEGER (0..1)
48         MAX-ACCESS  read-create
49         STATUS      current
50         DESCRIPTION
51             "The value of this parameter specifies whether the SDUs
52             on the service flow are fixed-length (0) or
53             variable-length (1). The parameter is used only if
54             packing is on for the service flow. The default value

```

```

1         is 0, i.e.,variable-length SDUs."
2     REFERENCE
3         "Section 11.13.16 in IEEE 802.16REvd/D4-2004"
4     DEFVAL     { 0 }
5     ::= { wmanIfBsServiceClassEntry 9 }
6
7     wmanIfBsQoSsduSize OBJECT-TYPE
8         SYNTAX     INTEGER
9         MAX-ACCESS read-create
10        STATUS     current
11        DESCRIPTION
12            "The value of this parameter specifies the length of the
13            SDU for a fixed-length SDU service flow. This parameter
14            is used only if packing is on and the service flow is
15            indicated as carrying fixed-length SDUs. The default
16            value is 49 bytes, i.e., VC-switched ATM cells with PHS.
17            The parameter is relevant for both ATM and Packet
18            Convergence Sublayers."
19        REFERENCE
20            "Section 11.13.17 in IEEE 802.16REvd/D4-2004"
21        DEFVAL     { 49 }
22        ::= { wmanIfBsServiceClassEntry 10 }
23
24        wmanIfBsQoSschedulingType OBJECT-TYPE
25            SYNTAX     wmanSfSchedulingType
26            MAX-ACCESS read-create
27            STATUS     current
28            DESCRIPTION
29                "Specifies the upstream scheduling service used for
30                upstream service flow. If the referenced parameter
31                is not present in the corresponding 802.16 QoS
32                Parameter Set of an upstream service flow, the
33                default value of this object is bestEffort(2)."
34            REFERENCE
35                "Section 11.13.13 in IEEE 802.16REvd/D3-2004"
36            DEFVAL     {2}
37            ::= { wmanIfBsServiceClassEntry 11 }
38
39        wmanIfBsQoSsArqEnable OBJECT-TYPE
40            SYNTAX     TruthValue
41            MAX-ACCESS read-create
42            STATUS     current
43            DESCRIPTION
44                "True(1) ARQ enabling is requested for the connection."
45            REFERENCE
46                "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
47            ::= { wmanIfBsServiceClassEntry 12 }
48
49        wmanIfBsQoSsArqWindowSize OBJECT-TYPE
50            SYNTAX     INTEGER (1 .. 1024)
51            MAX-ACCESS read-create
52            STATUS     current
53            DESCRIPTION
54                "Indicates the maximum number of unacknowledged

```

```

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

```

```

1      STATUS      current
2      DESCRIPTION
3          "Indicates the time interval the ARQ window is advanced
4          after a fragment is received. A value of 0 means
5          Infinite."
6      REFERENCE
7          "Section 11.13.20 in IEEE 802.16REvd/D3-2004"
8      DEFVAL      {0}
9      ::= { wmanIfBsServiceClassEntry 17 }
10
11     wmanIfBsQoSscFragmentLen OBJECT-TYPE
12         SYNTAX      INTEGER (32 .. 2040)
13         MAX-ACCESS  read-create
14         STATUS      current
15         DESCRIPTION
16             "The maximum size fragment a transmitter shall form
17             or a receiver shall expect to receive."
18         ::= { wmanIfBsServiceClassEntry 18 }
19
20     wmanIfBsQoSSCminRsvdTolerableRate OBJECT-TYPE
21         SYNTAX      INTEGER
22         UNITS       "bps"
23         MAX-ACCESS  read-create
24         STATUS      current
25         DESCRIPTION
26             "Minimum Tolerable Traffic Rate = R (bits/sec) with
27             time base T(sec) means the following. Let S denote
28             additional demand accumulated at the MAC SAP of the
29             transmitter during an arbitrary time interval of the
30             length T. Then the amount of data forwarded at the
31             receiver to CS (in bits) during this interval should
32             be not less than min {S, R * T}."
33         REFERENCE  "Section 11.13.11 in IEEE 802.16REvd/D3-2004"
34         ::= { wmanIfBsServiceClassEntry 19 }
35
36     wmanIfBsQoSServiceClassRowStatus OBJECT-TYPE
37         SYNTAX      RowStatus
38         MAX-ACCESS  read-create
39         STATUS      current
40         DESCRIPTION
41             "This object is used to create a new row or modify or
42             delete an existing row in this table.
43
44             If the implementator of this MIB has chosen not
45             to implement 'dynamic assignment' of profiles, this
46             object is not useful and should return noSuchName
47             upon SNMP request."
48         ::= { wmanIfBsServiceClassEntry 20 }
49
50     wmanIfBsClassifierRuleTable OBJECT-TYPE
51         SYNTAX      SEQUENCE OF wmanIfBsClassifierRuleEntry
52         MAX-ACCESS  not-accessible
53         STATUS      current
54         DESCRIPTION

```

```

1           "This table contains packet classifier rules associated
2           with service flows."
3     REFERENCE
4           "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
5     ::= { wmanIfBsPacketCs 3 }
6
7     wmanIfBsClassifierRuleEntry OBJECT-TYPE
8       SYNTAX      wmanIfBsClassifierRuleEntry
9       MAX-ACCESS  not-accessible
10      STATUS      current
11      DESCRIPTION
12        "This table provides one row for each packet classifier
13        rule, and is indexed by wmanIfBsSfId and
14        wmanIfBsClassifierRuleIndex. wmanIfBsSfIndex
15        identifies the service flow, while
16        wmanIfBsClassifierRuleIndex identifies the packet
17        classifier rule."
18      INDEX { wmanIfBsSfIndex, wmanIfBsClassifierRuleIndex }
19      ::= { wmanIfBsClassifierRuleTable 1 }
20
21     wmanIfBsClassifierRuleEntry ::= SEQUENCE {
22       wmanIfBsSfIndex                Unsigned32,
23       wmanIfBsClassifierRuleIndex    Unsigned32,
24       wmanIfBsClassifierRulePriority  INTEGER,
25       wmanIfBsClassifierRuleIpTosLow  OCTET STRING,
26       wmanIfBsClassifierRuleIpTosHigh OCTET STRING,
27       wmanIfBsClassifierRuleIpTosMask OCTET STRING,
28       wmanIfBsClassifierRuleIpProtocol Integer32,
29       wmanIfBsClassifierRuleInetAddressType InetAddressType,
30       wmanIfBsClassifierRuleInetAddress InetAddress,
31       wmanIfBsClassifierRuleInetAddressMask InetAddress,
32       wmanIfBsClassifierRuleInetAddressDestAddr InetAddress,
33       wmanIfBsClassifierRuleInetAddressDestMask InetAddress,
34       wmanIfBsClassifierRuleSourcePortStart Integer32,
35       wmanIfBsClassifierRuleSourcePortEnd Integer32,
36       wmanIfBsClassifierRuleDestPortStart Integer32,
37       wmanIfBsClassifierRuleDestPortEnd Integer32,
38       wmanIfBsClassifierRuleDestMacAddr MacAddress,
39       wmanIfBsClassifierRuleDestMacMask MacAddress,
40       wmanIfBsClassifierRuleSourceMacAddr MacAddress,
41       wmanIfBsClassifierRuleSourceMacMask MacAddress,
42       wmanIfBsClassifierRuleEnetProtocolType INTEGER,
43       wmanIfBsClassifierRuleEnetProtocol Integer32,
44       wmanIfBsClassifierRuleUserPriLow Integer32,
45       wmanIfBsClassifierRuleUserPriHigh Integer32,
46       wmanIfBsClassifierRuleVlanId Integer32,
47       wmanIfBsClassifierRuleState INTEGER,
48       wmanIfBsClassifierRulePkts Counter64,
49       wmanIfBsClassifierRuleRowStatus RowStatus
50     }
51
52     wmanIfBsSfIndex OBJECT-TYPE
53       SYNTAX      Unsigned32 (1 .. 4294967295)
54       MAX-ACCESS  not-accessible

```



```

1      STATUS      current
2      DESCRIPTION
3          "A 32 bit quantity that uniquely identifies a service flow
4          to both the subscriber station and base station (BS)."
```

::= { wmanIfBsClassifierRuleEntry 1 }

```

6
7      wmanIfBsClassifierRuleIndex OBJECT-TYPE
8          SYNTAX      Unsigned32 (1..4294967295)
9          MAX-ACCESS  not-accessible
10         STATUS      current
11         DESCRIPTION
12             "An index is assigned to a classifier in BS classifiers
13             table"
```

::= { wmanIfBsClassifierRuleEntry 2 }

```

15
16     wmanIfBsClassifierRulePriority OBJECT-TYPE
17         SYNTAX      INTEGER
18         MAX-ACCESS  read-create
19         STATUS      current
20         DESCRIPTION
21             "The value specifies the priority for the Classifier, which
22             is used for determining the order of the Classifier. A
23             higher value indicates higher priority. Classifiers may
24             have
25             priorities in the range 0..255 with the default value = 0."
26         REFERENCE
27             "Section 11.13.21.3.4.3 in IEEE 802.16REvd/D4-2004"
```

DEFVAL { 0 }

::= { wmanIfBsClassifierRuleEntry 3 }

```

30
31     wmanIfBsClassifierRuleIpTosLow OBJECT-TYPE
32         SYNTAX      OCTET STRING (SIZE(1))
33         MAX-ACCESS  read-create
34         STATUS      current
35         DESCRIPTION
36             "The low value of a range of TOS byte values. If the
37             referenced parameter is not present in a classifier, this
38             object reports the value of 0."
39         REFERENCE
40             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
```

::= { wmanIfBsClassifierRuleEntry 4 }

```

42
43     wmanIfBsClassifierRuleIpTosHigh OBJECT-TYPE
44         SYNTAX      OCTET STRING (SIZE(1))
45         MAX-ACCESS  read-create
46         STATUS      current
47         DESCRIPTION
48             "The 8-bit high value of a range of TOS byte values.
49             If the referenced parameter is not present in a classifier,
50             this object reports the value of 0."
51         REFERENCE
52             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
```

::= { wmanIfBsClassifierRuleEntry 5 }

```

54
```

```

1  wmanIfBsClassifierRuleIpTosMask OBJECT-TYPE
2      SYNTAX      OCTET STRING (SIZE(1))
3      MAX-ACCESS  read-create
4      STATUS      current
5      DESCRIPTION
6          "The mask value is bitwise ANDed with TOS byte in an IP
7           packet and this value is used check range checking of
8           TosLow and TosHigh. If the referenced parameter is not
9           present in a classifier, this object reports the value
10          of 0."
11     REFERENCE
12         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
13     ::= { wmanIfBsClassifierRuleEntry 6 }
14
15  wmanIfBsClassifierRuleIpProtocol OBJECT-TYPE
16      SYNTAX      Integer32 (0..255)
17      MAX-ACCESS  read-create
18      STATUS      current
19      DESCRIPTION
20          "This object indicates the value of the IP Protocol field
21           required for IP packets to match this rule. If the
22           referenced parameter is not present in a classifier, this
23           object reports the value of 0."
24     REFERENCE
25         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
26     ::= { wmanIfBsClassifierRuleEntry 7 }
27
28  wmanIfBsClassifierRuleInetAddressType OBJECT-TYPE
29      SYNTAX      InetAddressType
30      MAX-ACCESS  read-create
31      STATUS      current
32      DESCRIPTION
33          "The type of the internet address for
34           wmanIfBsClassifierRuleInetAddress,
35           wmanIfBsClassifierRuleInetAddressMask,
36           wmanIfBsClassifierRuleInetAddress, and
37           wmanIfBsClassifierRuleInetAddressMask.
38           If the referenced parameter is not present in a classifier,
39           this object reports the value of ipv4(1)."
40     REFERENCE
41         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
42     ::= { wmanIfBsClassifierRuleEntry 8 }
43
44  wmanIfBsClassifierRuleInetAddress OBJECT-TYPE
45      SYNTAX      InetAddress
46      MAX-ACCESS  read-create
47      STATUS      current
48      DESCRIPTION
49          "This object specifies the value of the IP Source Address
50           required for packets to match this rule. An IP packet
51           matches the rule when the packet ip source address bitwise
52           ANDed with the wmanIfBsClassifierRuleInetAddressMask value
53           equals the wmanIfBsClassifierRuleInetAddress value.
54           If the referenced parameter is not present n a classifier,

```

```
1         this object reports the value of 0.0.0.0."
2     REFERENCE
3         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
4     ::= { wmanIfBsClassifierRuleEntry 9 }
5
6     wmanIfBsClassifierRuleInetSourceMask OBJECT-TYPE
7         SYNTAX      InetAddress
8         MAX-ACCESS  read-create
9         STATUS      current
10        DESCRIPTION
11            "This object specifies which bits of a packet's IP Source
12             Address that are compared to match this rule. An IP packet
13             matches the rule when the packet source address bitwise
14             ANDED with the
15             wmanIfBsClassifierRuleInetSourceMask value equals the
16             wmanIfBsClassifierRuleInetSourceAddr value.
17             If the referenced parameter is not present in a classifier,
18             this object reports the value of 0.0.0.0."
19        REFERENCE
20            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
21        ::= { wmanIfBsClassifierRuleEntry 10 }
22
23     wmanIfBsClassifierRuleInetDestAddr OBJECT-TYPE
24         SYNTAX      InetAddress
25         MAX-ACCESS  read-create
26         STATUS      current
27         DESCRIPTION
28            "This object specifies the value of the IP Destination
29             Address required for packets to match this rule. An IP
30             packet matches the rule when the packet IP destination
31             address bitwise ANDED with the
32             wmanIfBsClassifierRuleInetDestMask value equals the
33             wmanIfBsClassifierRuleInetDestAddr value.
34             If the referenced parameter is not present in a
35             classifier, this object reports the value of 0.0.0.0."
36        REFERENCE
37            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
38        ::= { wmanIfBsClassifierRuleEntry 11 }
39
40     wmanIfBsClassifierRuleInetDestMask OBJECT-TYPE
41         SYNTAX      InetAddress
42         MAX-ACCESS  read-create
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             wmanIfBsClassifierRuleInetDestMask value equals the
50             wmanIfBsClassifierRuleInetDestAddr 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
54            "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
```

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

```

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

```

```

1         MAC address bitwise ANDed with
2         wmanIfBsClassifierRuleSourceMacMask equals the value of
3         wmanIfBsClassifierRuleSourceMacAddr. If the referenced
4         parameter is not present in a classifier, this object
5         reports the value of '000000000000'H."
6     REFERENCE
7         "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
8     ::= { wmanIfBsClassifierRuleEntry 20 }
9
10    wmanIfBsClassifierRuleEnetProtocolType OBJECT-TYPE
11        SYNTAX          INTEGER {none(0),
12                        ethertype(1),
13                        dsap(2)}
14        MAX-ACCESS      read-create
15        STATUS           current
16        DESCRIPTION
17            "This object indicates the format of the layer 3 protocol
18            id in the Ethernet packet. A value of none(0) means that
19            the rule does not use the layer 3 protocol type as a
20            matching criteria. A value of ethertype(1) means that the
21            rule applies only to frames which contains an EtherType
22            value. Ethertype values are contained in packets using
23            the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
24            Sub-Network Access Protocol (SNAP) encapsulation formats.
25            A value of dsap(2) means that the rule applies only to
26            frames using the IEEE802.3 encapsulation format with a
27            Destination Service Access Point (DSAP) other than 0xAA
28            (which is reserved for SNAP). If the Ethernet frame
29            contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
30            this object applies to the embedded EtherType field within
31            the 802.1P/Q header. If the referenced parameter is not
32            present in a classifier, this object reports the value of
33            0."
34        REFERENCE
35            "Section 11.13.22.3.4 in IEEE 802.16REVd/D3-2004"
36        ::= { wmanIfBsClassifierRuleEntry 21 }
37
38    wmanIfBsClassifierRuleEnetProtocol OBJECT-TYPE
39        SYNTAX          Integer32 (0..65535)
40        MAX-ACCESS      read-create
41        STATUS           current
42        DESCRIPTION
43            "If wmanIfBsClassifierRuleEnetProtocolType is none(0),
44            this object is ignored when considering whether a packet
45            matches the current rule.
46            If wmanIfBsClassifierRuleEnetProtocolType is ethertype(1),
47            this object gives the 16-bit value of the EtherType that
48            the packet must match in order to match the rule.
49            If wmanIfBsClassifierRuleEnetProtocolType is dsap(2), the
50            lower 8 bits of this object's value must match the DSAP
51            byte of the packet in order to match the rule.
52            If wmanIfBsClassifierRuleEnetProtocolType is mac(3), the
53            lower 8 bits of this object value represent a lower bound
54            (inclusive) of MAC management message type codes matched,

```

```

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

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object applies only to Ethernet frames using the
4          802.1P/Q tag header.
5          If this object's value is nonzero, tagged packets must
6          have a VLAN Identifier that matches the value in order
7          to match the rule.
8          Only the least significant 12 bits of this object's
9          value are valid.
10         If the referenced parameter is not present in the
11         classifier, the value of this object is reported as 0."
12     REFERENCE
13         "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
14     ::= { wmanIfBsClassifierRuleEntry 25 }
15
16     wmanIfBsClassifierRuleState OBJECT-TYPE
17         SYNTAX      INTEGER {active(1),
18                     inactive(2)}
19         MAX-ACCESS  read-create
20         STATUS      current
21         DESCRIPTION
22             "This object indicates whether or not the classifier is
23             enabled to classify packets to a Service Flow.
24             If the referenced parameter is not present in the
25             classifier, the value of this object is reported
26             as active(1)."
27         REFERENCE
28             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
29         ::= { wmanIfBsClassifierRuleEntry 26 }
30
31     wmanIfBsClassifierRulePkts OBJECT-TYPE
32         SYNTAX      Counter64
33         MAX-ACCESS  read-create
34         STATUS      current
35         DESCRIPTION
36             "This object counts the number of packets that have
37             been classified using this entry."
38         REFERENCE
39             "Section 11.13.22.3.4 in IEEE 802.16REvd/D3-2004"
40         ::= { wmanIfBsClassifierRuleEntry 27 }
41
42     wmanIfBsClassifierRuleRowStatus OBJECT-TYPE
43         SYNTAX      RowStatus
44         MAX-ACCESS  read-create
45         STATUS      current
46         DESCRIPTION
47             "This object is used to create a new row or modify or
48             delete an existing row in this table.
49
50             If the implementator of this MIB has choosen not
51             to implement 'dynamic assignment' of profiles, this
52             object is not useful and should return noSuchName
53             upon SNMP request."
54         ::= { wmanIfBsClassifierRuleEntry 28 }

```



```

1
2  --
3  -- wmanIfBsCps contain the Base Station Common Part Sublayer objects
4  wmanIfBsCps OBJECT IDENTIFIER ::= { wmanIfBsObjects 3 }
5
6  --
7  -- wmanIfBsConfigurationTable contains global parameters common in BS
8  --
9  wmanIfBsConfigurationTable OBJECT-TYPE
10     SYNTAX      SEQUENCE OF WmanIfBsConfigurationEntry
11     MAX-ACCESS  not-accessible
12     STATUS      current
13     DESCRIPTION
14         "This table provides one row for each BS sector that
15         contains the BS system parameters as defined in section
16         10.1 of [3]."

```

```

1  wmanIfBsDcdInterval OBJECT-TYPE
2      SYNTAX      INTEGER(0..10000)
3      UNITS       "milliseconds"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "Time between transmission of DCD messages in ms."
8      ::= { wmanIfBsConfigurationEntry 1 }
9
10 wmanIfBsUcdInterval OBJECT-TYPE
11     SYNTAX      INTEGER(0..10000)
12     UNITS       "milliseconds"
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Time between transmission of UCD messages in ms."
17     ::= { wmanIfBsConfigurationEntry 2 }
18
19 wmanIfBsUcdTransition OBJECT-TYPE
20     SYNTAX      INTEGER
21     UNITS       "Number of MAC Frames"
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         "The time the BS shall wait after repeating a UCD message
26         with an incremented Configuration Change Count before
27         issuing a UL-MAP message referring to
28         Downlink_Burst_Profiles defined in that UCD message."
29     ::= { wmanIfBsConfigurationEntry 3 }
30
31 wmanIfBsDcdTransition OBJECT-TYPE
32     SYNTAX      INTEGER
33     UNITS       "Number of MAC Frames"
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37         "The time the BS shall wait after repeating a DCD message
38         with an incremented Configuration Change Count before
39         issuing a DL-MAP message referring to Uplink_Burst_Profiles
40         defined in that DCD message."
41     ::= { wmanIfBsConfigurationEntry 4 }
42
43 wmanIfBsMaxMAPPending OBJECT-TYPE
44     SYNTAX      INTEGER
45     MAX-ACCESS  read-write
46     STATUS      current
47     DESCRIPTION
48         "Maximum validity of map."
49     ::= { wmanIfBsConfigurationEntry 5 }
50
51 wmanIfBsInitialRangingInterval OBJECT-TYPE
52     SYNTAX      INTEGER(0..2000)
53     UNITS       "milliseconds"
54     MAX-ACCESS  read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          "Time between Initial Ranging regions assigned by the BS
4          in ms."
5      ::= { wmanIfBsConfigurationEntry 6 }
6
7      wmanIfBsClkCmpInterval OBJECT-TYPE
8          SYNTAX      INTEGER(50..50)
9          UNITS        "milliseconds"
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "Time between the clock compare measurements used for the
14             generation of CLK-CMP messages."
15         ::= { wmanIfBsConfigurationEntry 7 }
16
17         wmanIfBsSsULMapProcTime OBJECT-TYPE
18             SYNTAX      Unsigned32 (200 .. 4294967295)
19             UNITS        "micro seconds"
20             MAX-ACCESS  read-write
21             STATUS      current
22             DESCRIPTION
23                 "Time provided between arrival of the last bit of a UL-MAP
24                 at an SS and effectiveness of that map in us."
25             ::= { wmanIfBsConfigurationEntry 8 }
26
27         wmanIfBsSsRangRespProcTime OBJECT-TYPE
28             SYNTAX      Unsigned32 (10000 .. 4294967295)
29             UNITS        "micro seconds"
30             MAX-ACCESS  read-write
31             STATUS      current
32             DESCRIPTION
33                 "Time allowed for an SS following receipt of a ranging
34                 response before it is expected to reply to an invited
35                 ranging request in us."
36             ::= { wmanIfBsConfigurationEntry 9 }
37
38         wmanIfBsT5Timeout OBJECT-TYPE
39             SYNTAX      INTEGER(0 .. 2000)
40             UNITS        "milliseconds"
41             MAX-ACCESS  read-write
42             STATUS      current
43             DESCRIPTION
44                 "Wait for Uplink Channel Change Response in ms."
45             ::= { wmanIfBsConfigurationEntry 10 }
46
47         wmanIfBsT9Timeout OBJECT-TYPE
48             SYNTAX      INTEGER(300 .. 65535)
49             UNITS        "milliseconds"
50             MAX-ACCESS  read-write
51             STATUS      current
52             DESCRIPTION
53                 "Registration Timeout, the time allowed between the BS
54                 sending a RNG-RSP (success) to an SS, and receiving a

```

```

1         SBC-REQ from that same SS in ms."
2         ::= { wmanIfBsConfigurationEntry 11 }
3
4     wmanIfBsT13Timeout OBJECT-TYPE
5         SYNTAX      INTEGER(15 .. 65535)
6         UNITS       "minutes"
7         MAX-ACCESS  read-write
8         STATUS      current
9         DESCRIPTION
10          "The time allowed for an SS, following receipt of a
11           REG-RSP message to send a TFTP-CPLT message to the BS
12           in min."
13         ::= { wmanIfBsConfigurationEntry 12 }
14
15     wmanIfBsT15Timeout OBJECT-TYPE
16         SYNTAX      INTEGER(20 .. 65535)
17         UNITS       "milliseconds"
18         MAX-ACCESS  read-write
19         STATUS      current
20         DESCRIPTION
21          "Wait for MCA-RSP in ms."
22         ::= { wmanIfBsConfigurationEntry 13 }
23
24     wmanIfBsT17Timeout OBJECT-TYPE
25         SYNTAX      INTEGER(5 .. 65535)
26         UNITS       "minutes"
27         MAX-ACCESS  read-write
28         STATUS      current
29         DESCRIPTION
30          "Time allowed for SS to complete SS Authorization and
31           Key Exchange in min ."
32         ::= { wmanIfBsConfigurationEntry 14 }
33
34     wmanIfBsT27IdleTimer OBJECT-TYPE
35         SYNTAX      INTEGER
36         UNITS       "milliseconds"
37         MAX-ACCESS  read-write
38         STATUS      current
39         DESCRIPTION
40          "Maximum time between unicast grants to SS when BS believes
41           SS uplink transmission quality is good enough."
42         ::= { wmanIfBsConfigurationEntry 15 }
43
44     wmanIfBsT27ActiveTimer OBJECT-TYPE
45         SYNTAX      INTEGER
46         UNITS       "milliseconds"
47         MAX-ACCESS  read-write
48         STATUS      current
49         DESCRIPTION
50          "Maximum time between unicast grants to SS when BS believes
51           SS uplink transmission quality is not good enough."
52         ::= { wmanIfBsConfigurationEntry 16 }
53
54     wmanIfBsRssiLowThreshold OBJECT-TYPE

```

```

1      SYNTAX      INTEGER
2      UNITS       "dBm"
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "Low threshold for generating the RSSI alarm trap.
7           The detection of RSSI alarm will be disabled until the
8           RSSI goes above wmanIfBsRssiHighThreshold"
9      ::= { wmanIfBsConfigurationEntry 17 }
10
11     wmanIfBsRssiHighThreshold OBJECT-TYPE
12         SYNTAX      INTEGER
13         UNITS       "dBm"
14         MAX-ACCESS  read-write
15         STATUS      current
16         DESCRIPTION
17             "High threshold for generating a trap indicating
18              the the RSSI alarm is restored."
19         ::= { wmanIfBsConfigurationEntry 18 }
20
21     wmanIfBsTempLowAlarmThreshold OBJECT-TYPE
22         SYNTAX      INTEGER
23         UNITS       "degreeF"
24         MAX-ACCESS  read-write
25         STATUS      current
26         DESCRIPTION
27             "Low threshold for generating the temperature low alarm
28              trap. The detection of temperature low alarm will be
29              disabled until the temperature goes above
30              wmanIfBsTempLowAlarmRestoredThreshold"
31         ::= { wmanIfBsConfigurationEntry 19 }
32
33     wmanIfBsTempLowAlarmRestoredThreshold OBJECT-TYPE
34         SYNTAX      INTEGER
35         UNITS       "degreeF"
36         MAX-ACCESS  read-write
37         STATUS      current
38         DESCRIPTION
39             "Low threshold for generating a trap indicating
40              the temperature alarm is restored."
41         ::= { wmanIfBsConfigurationEntry 20 }
42
43     wmanIfBsTempHighAlarmThreshold OBJECT-TYPE
44         SYNTAX      INTEGER
45         UNITS       "degreeF"
46         MAX-ACCESS  read-write
47         STATUS      current
48         DESCRIPTION
49             "Low threshold for generating the temperature low alarm
50              trap. The detection of temperature low alarm will be
51              disabled until the temperature goes above
52              wmanIfBsTempLowAlarmRestoredThreshold"
53         ::= { wmanIfBsConfigurationEntry 21 }
54

```

```

1  wmanIfBsTempHighAlarmRestoredThreshold OBJECT-TYPE
2      SYNTAX      INTEGER
3      UNITS      "degreeF"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "High threshold for generating a trap indicating
8           the temperature alarm is restored."
9      ::= { wmanIfBsConfigurationEntry 22 }
10
11  wmanIfBsConfigurationRowStatus OBJECT-TYPE
12      SYNTAX      RowStatus
13      MAX-ACCESS  read-create
14      STATUS      current
15      DESCRIPTION
16          "This object is used to create a new row or modify or
17           delete an existing row in this table.
18
19           If the implementator of this MIB has chosen not
20           to implement 'dynamic assignment' of profiles, this
21           object is not useful and should return noSuchName
22           upon SNMP request."
23      ::= { wmanIfBsConfigurationEntry 23 }
24
25  --
26  -- Base Station statistics counters
27  --
28  wmanIfBsStatisticCounter OBJECT IDENTIFIER ::= { wmanIfBsCps 2 }
29
30  wmanIfBsChMeasurementTable OBJECT-TYPE
31      SYNTAX      SEQUENCE OF wmanIfBsChMeasurementEntry
32      MAX-ACCESS  not-accessible
33      STATUS      current
34      DESCRIPTION
35          "This table contains channel measurement information
36           on the uplink signal received from SS. The table shall
37           be maintained as FIFO to store measurement samples that
38           can be used to create RSSI and CINR histogram report.
39           When the measurement entry for a SS reaches the limit,
40           the oldest entry shall be deleted as the new entry is
41           added to the table."
42      ::= { wmanIfBsStatisticCounter 1 }
43
44  wmanIfBsChMeasurementEntry OBJECT-TYPE
45      SYNTAX      wmanIfBsChMeasurementEntry
46      MAX-ACCESS  not-accessible
47      STATUS      current
48      DESCRIPTION
49          "Each entry in the table contains RSSI and CINR
50           signal quality measurement on signal received from the SS.
51           The primary index is the ifIndex with ifType of propBWA2Mp
52           identifying the BS sector. wmanIfChSsIdIndex identifies
53           the SS from which the signal was received.
54           wmanIfBsHistogramIndex is the index to histogram samples.

```

```

1           Since there is no time stamp in the table,
2           wmanIfBsHistogramIndex should be increased monotonically,
3           and warps around when it reaches the limit. "
4   INDEX      { ifIndex, wmanIfBsChSsIdIndex,
5               wmanIfBsHistogramIndex }
6   ::= { wmanIfBsChMeasurementTable 1 }
7
8   wmanIfBsChMeasurementEntry ::= SEQUENCE {
9       wmanIfBsChSsIdIndex          Unsigned32,
10      wmanIfBsHistogramIndex       Unsigned32,
11      wmanIfBsChannelNumber        INTEGER,
12      wmanIfBsStartFrame           INTEGER,
13      wmanIfBsDuration             INTEGER,
14      wmanIfBsBasicReport          BITS,
15      wmanIfBsMeanCinrReport       INTEGER,
16      wmanIfBsMeanRssiReport       INTEGER
17  }
18
19  wmanIfBsChSsIdIndex OBJECT-TYPE
20      SYNTAX      Unsigned32 (1 .. 4294967295)
21      MAX-ACCESS  read-only
22      STATUS      current
23      DESCRIPTION
24          "wmanIfBsChIdIndex identifies the SS providing the
25           channel measurement."
26      REFERENCE
27          "Section 6.4.2.3.5 in IEEE 802.16REVd/D3-2004"
28      ::= { wmanIfBsChMeasurementEntry 1 }
29
30  wmanIfBsHistogramIndex OBJECT-TYPE
31      SYNTAX      Unsigned32 (1 .. 4294967295)
32      MAX-ACCESS  read-only
33      STATUS      current
34      DESCRIPTION
35          "wmanIfBsHistogramIndex identifies the histogram samples
36           in the table for each subscriber station."
37      ::= { wmanIfBsChMeasurementEntry 2 }
38
39  wmanIfBsChannelNumber OBJECT-TYPE
40      SYNTAX      INTEGER
41      MAX-ACCESS  read-only
42      STATUS      current
43      DESCRIPTION
44          "Physical channel number to be reported on is only
45           applicable to licence exempt band. For licenced band,
46           this parameter should be null."
47      REFERENCE
48          "Section 8.5.1 in IEEE 802.16REVd/D3-2004"
49      ::= { wmanIfBsChMeasurementEntry 3 }
50
51  wmanIfBsStartFrame OBJECT-TYPE
52      SYNTAX      INTEGER
53      MAX-ACCESS  read-only
54      STATUS      current

```

```

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

```



```

1           802.16REvD/D3-2004"
2           ::= { wmanIfBsChMeasurementEntry 8 }
3
4           --
5           -- Base station PKM group
6           -- wmanIfBsPkmObjects contain the Base Station Privacy Sublayer objects
7           wmanIfBsPkmObjects OBJECT IDENTIFIER ::= { wmanIfBsObjects 4 }
8
9           --
10          -- Table wmanIfBsPkmBaseTable
11          --
12          wmanIfBsPkmBaseTable OBJECT-TYPE
13              SYNTAX      SEQUENCE OF      WmanIfBsPkmBaseEntry
14              MAX-ACCESS  not-accessible
15              STATUS      current
16              DESCRIPTION
17                  "This table describes the basic PKM attributes of each Base
18                  Station wireless interface."
19              ::= { wmanIfBsPkmObjects 1 }
20
21          wmanIfBsPkmBaseEntry OBJECT-TYPE
22              SYNTAX      WmanIfBsPkmBaseEntry
23              MAX-ACCESS  not-accessible
24              STATUS      current
25              DESCRIPTION
26                  "Each entry contains objects describing attributes of one
27                  BS wireless interface."
28              INDEX      { ifIndex }
29              ::= { wmanIfBsPkmBaseTable 1 }
30
31          WmanIfBsPkmBaseEntry ::= SEQUENCE {
32              wmanIfBsPkmDefaultAuthLifetime      Integer32,
33              wmanIfBsPkmDefaultTEKLifetime      Integer32,
34              wmanIfBsPkmDefaultSelfSigManufCertTrust INTEGER,
35              wmanIfBsPkmCheckCertValidityPeriods TruthValue,
36              wmanIfBsPkmAuthentInfos            Counter32,
37              wmanIfBsPkmAuthRequests            Counter32,
38              wmanIfBsPkmAuthReplies            Counter32,
39              wmanIfBsPkmAuthRejects            Counter32,
40              wmanIfBsPkmAuthInvalids           Counter32
41          }
42
43          wmanIfBsPkmDefaultAuthLifetime OBJECT-TYPE
44              SYNTAX      Integer32 (86400..6048000)
45              UNITS      "seconds"
46              MAX-ACCESS  read-write
47              STATUS      current
48              DESCRIPTION
49                  "The value of this object is the default lifetime, in
50                  seconds, the BS assigns to a new authorization key."
51              REFERENCE
52                  "IEEE 802.16 standard; Table 270"
53              DEFVAL      { 604800 }
54              ::= { wmanIfBsPkmBaseEntry 1 }

```

```

1
2  wmanIfBsPkmDefaultTEKLifetime OBJECT-TYPE
3      SYNTAX      Integer32 (1800..604800)
4      UNITS       "seconds"
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8          "The value of this object is the default lifetime, in
9           seconds, the BS assigns to a new Traffic Encryption
10          Key(TEK)."

```

```

1      STATUS      current
2      DESCRIPTION
3          "The value of this object is the count of times the BS has
4          received an Authorization Request message from any SS"
5      ::= { wmanIfBsPkmBaseEntry 6 }
6
7      wmanIfBsPkmAuthReplies 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 BS has
13             transmitted an Authorization Reply message to any SS."
14         ::= { wmanIfBsPkmBaseEntry 7 }
15
16         wmanIfBsPkmAuthRejects 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                 transmitted an Authorization Reject message to any SS."
23             ::= { wmanIfBsPkmBaseEntry 8 }
24
25         wmanIfBsPkmAuthInvalids 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 BS has
31                 transmitted an Authorization Invalid message to any SS."
32             ::= { wmanIfBsPkmBaseEntry 9 }
33
34         --
35         -- Table wmanIfBsPkmAuthTable
36         --
37         wmanIfBsPkmAuthTable OBJECT-TYPE
38             SYNTAX      SEQUENCE OF      wmanIfBsPkmAuthEntry
39             MAX-ACCESS  not-accessible
40             STATUS      current
41             DESCRIPTION
42                 "This table describes the attributes of each SS
43                 authorization association. The BS maintains one
44                 authorization association with each Baseline
45                 Privacy-enabled SS on each BS wireless interface."
46             ::= { wmanIfBsPkmObjects 2 }
47
48         wmanIfBsPkmAuthEntry OBJECT-TYPE
49             SYNTAX      wmanIfBsPkmAuthEntry
50             MAX-ACCESS  not-accessible
51             STATUS      current
52             DESCRIPTION
53                 "Each entry contains objects describing attributes of one
54                 authorization association. The BS MUST create one entry per

```

```

1         SS per wireless interface, based on the receipt of an
2         Authorization Request message, and MUST not delete the
3         entry before the SS authorization permanently expires."
4     INDEX        { ifIndex, wmanIfBSPkmAuthSsMacAddress }
5     ::= { wmanIfBSPkmAuthTable 1 }
6
7     wmanIfBSPkmAuthEntry ::= SEQUENCE {
8         wmanIfBSPkmAuthSsMacAddress           MacAddress,
9         wmanIfBSPkmAuthSsPublicKey           OCTET STRING,
10        wmanIfBSPkmAuthSsKeySequenceNumber   Integer32,
11        wmanIfBSPkmAuthSsExpiresOld         DateAndTime,
12        wmanIfBSPkmAuthSsExpiresNew        DateAndTime,
13        wmanIfBSPkmAuthSsLifetime          Integer32,
14        wmanIfBSPkmAuthSsGraceTime         Integer32,
15        wmanIfBSPkmAuthSsReset             INTEGER,
16        wmanIfBSPkmAuthSsInfos             Counter64,
17        wmanIfBSPkmAuthSsRequests          Counter64,
18        wmanIfBSPkmAuthSsReplies           Counter64,
19        wmanIfBSPkmAuthSsRejects          Counter64,
20        wmanIfBSPkmAuthSsInvalids          Counter64,
21        wmanIfBSPkmAuthRejectErrorCode     INTEGER,
22        wmanIfBSPkmAuthRejectErrorString   SnmpAdminString,
23        wmanIfBSPkmAuthInvalidErrorCode    INTEGER,
24        wmanIfBSPkmAuthInvalidErrorString  SnmpAdminString,
25        wmanIfBSPkmAuthPrimarySAId        Integer32,
26        wmanIfBSPkmAuthBpkmSsCertValid    INTEGER,
27        wmanIfBSPkmAuthBpkmSsCert         OCTET STRING
28    }
29
30    wmanIfBSPkmAuthSsMacAddress OBJECT-TYPE
31        SYNTAX      MacAddress
32        MAX-ACCESS  not-accessible
33        STATUS      current
34        DESCRIPTION
35            "The value of this object is the physical address of the SS
36             to which the authorization association applies."
37        ::= { wmanIfBSPkmAuthEntry 1 }
38
39    wmanIfBSPkmAuthSsPublicKey OBJECT-TYPE
40        SYNTAX      OCTET STRING (SIZE (140))
41        MAX-ACCESS  read-only
42        STATUS      current
43        DESCRIPTION
44            "The value of this object is a DER-encoded RSAPublicKey
45             ASN.1 type string, as defined in the RSA Encryption
46             Standard (PKCS #1) [10], corresponding to the public key of
47             the SS. The 74, 106, 140, 204, and 270 byte key encoding
48             lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
49             and 2048 public moduli respectively. This is a zero-length
50             string if the BS does not retain the public key."
51        ::= { wmanIfBSPkmAuthEntry 2 }
52
53    wmanIfBSPkmAuthSsKeySequenceNumber OBJECT-TYPE
54        SYNTAX      Integer32 (0..15)

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the most recent authorization
5           key sequence number for this SS."
6      ::= { wmanIfBsPkmAuthEntry 3 }
7
8      wmanIfBsPkmAuthSsExpiresOld OBJECT-TYPE
9          SYNTAX      DateAndTime
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "The value of this object is the actual clock time for
14              expiration of the immediate predecessor of the most recent
15              authorization key for this FSM. If this FSM has only one
16              authorization key, then the value is the time of activation
17              of this FSM."
18         ::= { wmanIfBsPkmAuthEntry 4 }
19
20         wmanIfBsPkmAuthSsExpiresNew OBJECT-TYPE
21             SYNTAX      DateAndTime
22             MAX-ACCESS  read-only
23             STATUS      current
24             DESCRIPTION
25                 "The value of this object is the actual clock time for
26                  expiration of the most recent authorization key for this
27                  FSM"
28             ::= { wmanIfBsPkmAuthEntry 5 }
29
30         wmanIfBsPkmAuthSsLifetime OBJECT-TYPE
31             SYNTAX      Integer32 (86400..6048000)
32             UNITS        "seconds"
33             MAX-ACCESS  read-write
34             STATUS      current
35             DESCRIPTION
36                 "The vaue of this object is the lifetime, in seconds, the
37                  BS assigns to an authorization key for this SS."
38             REFERENCE
39                 "IEEE 802.16 standard; Table 270"
40             DEFVAL      { 604800 }
41             ::= { wmanIfBsPkmAuthEntry 6 }
42
43         wmanIfBsPkmAuthSsGraceTime OBJECT-TYPE
44             SYNTAX      Integer32 (300..3024000)
45             UNITS        "seconds"
46             MAX-ACCESS  read-only
47             STATUS      current
48             DESCRIPTION
49                 "The value of this object is the grace time for the
50                  authorization key in seconds. The SS is expected to start
51                  trying to get a new authorization key beginning
52                  AuthGraceTime seconds before the authorization key actually
53                  expires."
54             REFERENCE

```

```

1           "IEEE 802.16 standard; Table 270"
2     DEFVAL      { 600 }
3     ::= { wmanIfBsPkmAuthEntry 7 }
4
5     wmanIfBsPkmAuthSsReset OBJECT-TYPE
6         SYNTAX      INTEGER { noResetRequested(1),
7                     invalidateAuth(2),
8                     sendAuthInvalid(3),
9                     invalidateTeks(4) }
10        MAX-ACCESS  read-write
11        STATUS      current
12        DESCRIPTION
13            "Setting this object to invalidateAuth(2) causes the BS to
14            invalidate the current SS authorization key(s), but not to
15            transmit an Authorization Invalid message nor to invalidate
16            unicast TEKS. Setting this object to sendAuthInvalid(3)
17            causes the BS to invalidate the current SS authorization
18            key(s), and to transmit an Authorization Invalid message to
19            the SS, but not to invalidate unicast TEKS. Setting this
20            object to invalidateTeks(4) causes the BS to invalidate the
21            current SS authorization key(s), to transmit an
22            Authorization Invalid message to the SS, and to
23            invalidate all unicast TEKS associated with this SS
24            authorization. Reading this object returns the
25            most-recently-set value of this object, or returns
26            noResetRequested(1) if the object has not been set since
27            the last BS reboot."
28        ::= { wmanIfBsPkmAuthEntry 8 }
29
30     wmanIfBsPkmAuthSsInfos OBJECT-TYPE
31         SYNTAX      Counter64
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             received an Authentication Information message from this
37             SS."
38         ::= { wmanIfBsPkmAuthEntry 9 }
39
40     wmanIfBsPkmAuthSsRequests OBJECT-TYPE
41         SYNTAX      Counter64
42         MAX-ACCESS  read-only
43         STATUS      current
44         DESCRIPTION
45             "The value of this object is the count of times the BS has
46             received an Authorization Request message from this SS."
47         ::= { wmanIfBsPkmAuthEntry 10 }
48
49     wmanIfBsPkmAuthSsReplies OBJECT-TYPE
50         SYNTAX      Counter64
51         MAX-ACCESS  read-only
52         STATUS      current
53         DESCRIPTION
54             "The value of this object is the count of times the BS has

```

```

1         transmitted an Authorization Reply message to this SS."
2     ::= { wmanIfBsPkmAuthEntry 11 }
3
4     wmanIfBsPkmAuthSsRejects 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            transmitted an Authorization Reject message to this SS."
11     ::= { wmanIfBsPkmAuthEntry 12 }
12
13     wmanIfBsPkmAuthSsInvalids OBJECT-TYPE
14         SYNTAX      Counter64
15         MAX-ACCESS  read-only
16         STATUS      current
17         DESCRIPTION
18             "The value of this object is the count of times the BS has
19            transmitted an Authorization Invalid message to this SS."
20     ::= { wmanIfBsPkmAuthEntry 13 }
21
22     wmanIfBsPkmAuthRejectErrorCode OBJECT-TYPE
23         SYNTAX      INTEGER {none(1),
24                        unknown(2),
25                        unauthorizedSs(3),
26                        unauthorizedSaid(4),
27                        permanentAuthorizationFailure(8),
28                        timeOfDayNotAcquired(11)}
29         MAX-ACCESS  read-only
30         STATUS      current
31         DESCRIPTION
32             "The value of this object is the enumerated description of
33            the Error-Code in most recent Authorization Reject message
34            transmitted to the SS. This has value unknown(2) if the
35            last Error-Code value was 0, and none(1) if no
36            Authorization Reject message has been transmitted to the
37            SS."
38     ::= { wmanIfBsPkmAuthEntry 14 }
39
40     wmanIfBsPkmAuthRejectErrorString OBJECT-TYPE
41         SYNTAX      SnmpAdminString (SIZE (0..128))
42         MAX-ACCESS  read-only
43         STATUS      current
44         DESCRIPTION
45             "The value of this object is the Display-String in most
46            recent Authorization Reject message transmitted to the SS.
47            This is a zero length string if no Authorization Reject
48            message has been transmitted to the SS."
49     ::= { wmanIfBsPkmAuthEntry 15 }
50
51     wmanIfBsPkmAuthInvalidErrorCode OBJECT-TYPE
52         SYNTAX      INTEGER {none(1),
53                        unknown(2),
54                        unauthorizedSs(3),

```

```

1             unsolicited(5),
2             invalidKeySequence(6),
3             keyRequestAuthenticationFailure(7)}
4     MAX-ACCESS    read-only
5     STATUS        current
6     DESCRIPTION
7         "The value of this object is the enumerated description of
8         the Error-Code in most recent Authorization Invalid message
9         transmitted to the SS. This has value unknown(2) if the
10        last Error-Code value was 0, and none(1) if no
11        Authorization Invalid message has been transmitted to the
12        SS."
13    ::= { wmanIfBsPkmAuthEntry 16 }
14
15    wmanIfBsPkmAuthInvalidErrorString OBJECT-TYPE
16        SYNTAX      SnmpAdminString (SIZE (0..128))
17        MAX-ACCESS  read-only
18        STATUS      current
19        DESCRIPTION
20            "The value of this object is the Display-String in most
21            recent Authorization Invalid message transmitted to the SS.
22            This is a zero length string if no Authorization Invalid
23            message has been transmitted to the SS."
24        ::= { wmanIfBsPkmAuthEntry 17 }
25
26    wmanIfBsPkmAuthPrimarySAId OBJECT-TYPE
27        SYNTAX      Integer32 (0..16383)
28        MAX-ACCESS  read-only
29        STATUS      current
30        DESCRIPTION
31            "The value of this object is the Primary Security
32            Association identifier."
33        ::= { wmanIfBsPkmAuthEntry 18 }
34
35    wmanIfBsPkmAuthBpkmSsCertValid OBJECT-TYPE
36        SYNTAX      INTEGER {unknown (0),
37                        validSsChained (1),
38                        validSsTrusted (2),
39                        invalidSsUntrusted (3),
40                        invalidCAUntrusted (4),
41                        invalidSsOther (5),
42                        invalidCAOther (6) }
43        MAX-ACCESS  read-only
44        STATUS      current
45        DESCRIPTION
46            "Contains the reason why a SS's certificate is deemed valid
47            or invalid. Return unknown if the SS is running PKM mode.
48            ValidSsChained means the certificate is valid because it
49            chains to a valid certificate. ValidSsTrusted means the
50            certificate is valid because it has been provisioned to be
51            trusted. InvalidSsUntrusted means the certificate is
52            invalid because it has been provisioned to be untrusted.
53            InvalidCAUntrusted means the certificate is invalid
54            because it chains to an untrusted certificate.

```



```

1         InvalidSsOther and InvalidCAOther refer to errors in
2         parsing, validity periods, etc, which are attributable to
3         the SS certificate or its chain respectively."
4     ::= { wmanIfBsPkmAuthEntry 19 }
5
6     wmanIfBsPkmAuthBpkmSsCert OBJECT-TYPE
7         SYNTAX      OCTET STRING
8         MAX-ACCESS  read-only
9         STATUS      current
10        DESCRIPTION
11            "The X509 SS Certificate sent as part of a PKM
12             Authorization Request."
13        ::= { wmanIfBsPkmAuthEntry 20 }
14
15    --
16    -- Table wmanIfBsPkmTEKTable
17    --
18    wmanIfBsPkmTEKTable OBJECT-TYPE
19        SYNTAX      SEQUENCE OF WmanIfBsPkmTEKEntry
20        MAX-ACCESS  not-accessible
21        STATUS      current
22        DESCRIPTION
23            "This table describes the attributes of each Traffic
24             Encryption Key (TEK) association. The BS maintains one TEK
25             association per SAID on each BS wireless interface."
26        ::= { wmanIfBsPkmObjects 3 }
27
28    wmanIfBsPkmTEKEntry OBJECT-TYPE
29        SYNTAX      WmanIfBsPkmTEKEntry
30        MAX-ACCESS  not-accessible
31        STATUS      current
32        DESCRIPTION
33            "Each entry contains objects describing attributes of one
34             TEK association on a particular BS wireless interface. The
35             BS MUST create one entry per SAID per wireless interface,
36             based on the receipt of a Key Request message, and MUST not
37             delete the entry before the SS authorization for the SAID
38             permanently expires."
39        INDEX      { ifIndex, wmanIfBsPkmTEKSAID }
40        ::= { wmanIfBsPkmTEKTable 1 }
41
42    wmanIfBsPkmTEKEntry ::= SEQUENCE {
43        wmanIfBsPkmTEKSAID          Integer32,
44        wmanIfBsPkmTEKSAType        INTEGER,
45        wmanIfBsPkmTEKDataEncryptAlg INTEGER,
46        wmanIfBsPkmTEKDataAuthentAlg INTEGER,
47        wmanIfBsPkmTEKEncryptAlg    INTEGER,
48        wmanIfBsPkmTEKLifetime      Integer32,
49        wmanIfBsPkmTEKGraceTime     Integer32,
50        wmanIfBsPkmTEKKeySequenceNumber Integer32,
51        wmanIfBsPkmTEKExpiresOld    DateAndTime,
52        wmanIfBsPkmTEKExpiresNew    DateAndTime,
53        wmanIfBsPkmTEKReset         TruthValue,
54        wmanIfBsPkmKeyRequests       Counter32,

```

```

1      wmanIfBSPkmKeyReplies          Counter32,
2      wmanIfBSPkmKeyRejects         Counter32,
3      wmanIfBSPkmTEKInvalids        Counter32,
4      wmanIfBSPkmKeyRejectErrorCode INTEGER,
5      wmanIfBSPkmKeyRejectErrorString SnmpAdminString,
6      wmanIfBSPkmTEKInvalidErrorCode INTEGER,
7      wmanIfBSPkmTEKInvalidErrorString SnmpAdminString
8      }
9
10     wmanIfBSPkmTEKSAID OBJECT-TYPE
11         SYNTAX      Integer32 (0..16383)
12         MAX-ACCESS  not-accessible
13         STATUS      current
14         DESCRIPTION
15             "The value of this object is the WiMAX Security Association
16              ID (SAID)."
```

```

17         ::= { wmanIfBSPkmTEKEntry 1 }
18
19     wmanIfBSPkmTEKSAType OBJECT-TYPE
20         SYNTAX      INTEGER {none(0),
21                             primary(1),
22                             static(2),
23                             dynamic(3) }
24         MAX-ACCESS  read-only
25         STATUS      current
26         DESCRIPTION
27             "The value of this object is the type of security
28              association. Dynamic does not apply to SSSs running in PKM
29              mode."
30         ::= { wmanIfBSPkmTEKEntry 2 }
31
32     wmanIfBSPkmTEKDataEncryptAlg OBJECT-TYPE
33         SYNTAX      INTEGER { none(0),
34                             des56CbcMode(1) }
35         MAX-ACCESS  read-only
36         STATUS      current
37         DESCRIPTION
38             "The value of this object is the data encryption algorithm
39              being utilized."
40         REFERENCE
41             "IEEE 802.16 standard; Table 301"
42         ::= { wmanIfBSPkmTEKEntry 3 }
43
44     wmanIfBSPkmTEKDataAuthentAlg OBJECT-TYPE
45         SYNTAX      INTEGER { none(0) }
46         MAX-ACCESS  read-only
47         STATUS      current
48         DESCRIPTION
49             "The value of this object is the data authentication
50              algorithm being utilized."
51         REFERENCE
52             "IEEE 802.16 standard; Table 302"
53         ::= { wmanIfBSPkmTEKEntry 4 }
54

```

```

1  wmanIfBsPkmTEKEncryptAlg OBJECT-TYPE
2      SYNTAX      INTEGER { tripleDES(0),
3                  rsa1024(1) }
4      MAX-ACCESS  read-only
5      STATUS      current
6      DESCRIPTION
7          "The value of this object is the TEK key encryption
8          algorithm being utilized."
9      REFERENCE
10         "IEEE 802.16 standard; Table 303"
11         ::= { wmanIfBsPkmTEKEntry 5 }
12
13  wmanIfBsPkmTEKLifetime OBJECT-TYPE
14      SYNTAX      Integer32 (1800..604800)
15      UNITS       "seconds"
16      MAX-ACCESS  read-write
17      STATUS      current
18      DESCRIPTION
19          "The value of this object is the lifetime, in seconds, the
20          BS assigns to keys for this TEK association."
21      REFERENCE
22         "IEEE 802.16 standard; Table 270"
23      DEFVAL      { 43200 }
24      ::= { wmanIfBsPkmTEKEntry 6 }
25
26  wmanIfBsPkmTEKGraceTime OBJECT-TYPE
27      SYNTAX      Integer32 (300..302399)
28      UNITS       "seconds"
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "The value of this object is the grace time for the TEK in
33          seconds. The SS is expected to start trying to acquire a
34          new TEK beginning TEK GraceTime seconds before the TEK
35          actually expires."
36      REFERENCE
37         "IEEE 802.16 standard; Table 270"
38      DEFVAL      { 3600 }
39      ::= { wmanIfBsPkmTEKEntry 7 }
40
41  wmanIfBsPkmTEKKeySequenceNumber OBJECT-TYPE
42      SYNTAX      Integer32 (0..15)
43      MAX-ACCESS  read-only
44      STATUS      current
45      DESCRIPTION
46          "The value of this object is the most recent TEK key
47          sequence number for this SAID."
48      ::= { wmanIfBsPkmTEKEntry 8 }
49
50  wmanIfBsPkmTEKExpiresOld OBJECT-TYPE
51      SYNTAX      DateAndTime
52      MAX-ACCESS  read-only
53      STATUS      current
54      DESCRIPTION

```

```

1           "The value of this object is the actual clock time for
2           expiration of the immediate predecessor of the most recent
3           TEK for this FSM. If this FSM has only one TEK, then the
4           value is the time of activation of this FSM."
5           ::= { wmanIfBsPkmTEKEntry 9 }
6
7 wmanIfBsPkmTEKExpiresNew OBJECT-TYPE
8     SYNTAX      DateAndTime
9     MAX-ACCESS  read-only
10    STATUS      current
11    DESCRIPTION
12      "The value of this object is the actual clock time for
13      expiration of the most recent TEK for this FSM."
14      ::= { wmanIfBsPkmTEKEntry 10 }
15
16 wmanIfBsPkmTEKReset OBJECT-TYPE
17     SYNTAX      TruthValue
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21       "Setting this object to TRUE causes the BS to invalidate
22       the current active TEK(s) (plural due to key transition
23       periods), and to generate a new TEK for the associated
24       SAID; the BS MAY also generate an unsolicited TEK Invalid
25       message, to optimize the TEK synchronization between the BS
26       and the SS. Reading this object always returns FALSE."
27       ::= { wmanIfBsPkmTEKEntry 11 }
28
29 wmanIfBsPkmKeyRequests OBJECT-TYPE
30     SYNTAX      Counter32
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34       "The value of this object is the count of times the BS has
35       received a Key Request message."
36       ::= { wmanIfBsPkmTEKEntry 12 }
37
38 wmanIfBsPkmKeyReplies OBJECT-TYPE
39     SYNTAX      Counter32
40     MAX-ACCESS  read-only
41     STATUS      current
42     DESCRIPTION
43       "The value of this object is the count of times the BS has
44       transmitted a Key Reply message."
45       ::= { wmanIfBsPkmTEKEntry 13 }
46
47 wmanIfBsPkmKeyRejects OBJECT-TYPE
48     SYNTAX      Counter32
49     MAX-ACCESS  read-only
50     STATUS      current
51     DESCRIPTION
52       "The value of this object is the count of times the BS has
53       transmitted a Key Reject message."
54       ::= { wmanIfBsPkmTEKEntry 14 }

```

```

1
2 wmanIfBsPkmTEKInvalids OBJECT-TYPE
3     SYNTAX      Counter32
4     MAX-ACCESS  read-only
5     STATUS      current
6     DESCRIPTION
7         "The value of this object is the count of times the BS has
8         transmitted a TEK Invalid message."
9     ::= { wmanIfBsPkmTEKEntry 15 }
10
11 wmanIfBsPkmKeyRejectErrorCode OBJECT-TYPE
12     SYNTAX      INTEGER {none(1),
13                    unknown(2),
14                    unauthorizedSaid(4)}
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is the enumerated; description of
19         the Error-Code in the most recent Key Reject message sent
20         in response to a Key Request for this SAID. This has value
21         unknown(2) if the last Error-Code value was 0, and none(1)
22         if no Key Reject message has been received since reboot."
23     ::= { wmanIfBsPkmTEKEntry 16 }
24
25 wmanIfBsPkmKeyRejectErrorString OBJECT-TYPE
26     SYNTAX      SnmpAdminString (SIZE (0..128))
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "The value of this object is the Display-String in the most
31         recent Key Reject message sent in response to a Key Request
32         for this SAID. This is a zero length string if no Key
33         Reject message has been received since reboot."
34     ::= { wmanIfBsPkmTEKEntry 17 }
35
36 wmanIfBsPkmTEKInvalidErrorCode OBJECT-TYPE
37     SYNTAX      INTEGER {none(1),
38                    unknown(2),
39                    invalidKeySequence(6)}
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 the most recent TEK Invalid message sent
45         in association with this SAID. This has value unknown(2)
46         if the last Error-Code value was 0, and none(1) if no TEK
47         Invalid message has been received since reboot."
48     ::= { wmanIfBsPkmTEKEntry 18 }
49
50 wmanIfBsPkmTEKInvalidErrorString 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 the most
2             recent TEK Invalid message sent in association with this
3             SAID. This is a zero length string if no TEK Invalid
4             message has been received since reboot."
5       ::= { wmanIfBsPkmTEKEntry 19 }
6
7       --
8       -- Base station Notification Group
9       -- wmanIfBsNotificationObjects contains the BS SNMP Trap objects
10      --
11      wmanIfBsNotification OBJECT IDENTIFIER ::= { wmanIfBsObjects 5 }
12      wmanIfBsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIfBsNotification 1 }
13      wmanIfBsTrapControl OBJECT IDENTIFIER ::= { wmanIfBsNotification 2 }
14
15      wmanIfBsTrapControlRegister OBJECT-TYPE
16          SYNTAX      BITS {wmanBsSsStatusNotification (0),
17                          wmanBsDynamicServiceFail (1),
18                          wmanBsPowerStatusChange (2),
19                          wmanBsFanStatusChange (3),
20                          wmanBsTemperatureChange (4),
21                          wmanBsRssiStatusChange (5)
22                          }
23          MAX-ACCESS read-write
24          STATUS      current
25          DESCRIPTION
26              "The object is used to enable Base Station traps. From left
27              to right, the set bit indicates the corresponding Base
28              Station trap is enabled."
29          ::= { wmanIfBsTrapControl 1 }
30
31      wmanIfBsSsNotificationObjectsTable OBJECT-TYPE
32          SYNTAX      SEQUENCE OF wmanIfBsSsNotificationObjectsEntry
33          MAX-ACCESS not-accessible
34          STATUS      current
35          DESCRIPTION
36              "This table contains SS notification objects that have been
37              reported by the trap."
38          ::= { wmanIfBsTrapDefinitions 1 }
39
40      wmanIfBsSsNotificationObjectsEntry OBJECT-TYPE
41          SYNTAX      wmanIfBsSsNotificationObjectsEntry
42          MAX-ACCESS not-accessible
43          STATUS      current
44          DESCRIPTION
45              "This table provides one row for each SS that has
46              generated traps, and is double indexed by
47              wmanIfBsTrapSsId and ifIndex for BS sector."
48          INDEX      { ifIndex, wmanIfBsTrapSsId }
49          ::= { wmanIfBsSsNotificationObjectsTable 1 }
50
51      wmanIfBsSsNotificationObjectsEntry ::= SEQUENCE {
52          wmanIfBsTrapSsId Unsigned32,
53          wmanIfBsSsMacAddress MacAddress,
54          wmanIfBsSsStatusValue INTEGER,

```

```

1      wmanIfBsSsStatusInfo          OCTET STRING,
2      wmanIfBsDynamicServiceType    INTEGER,
3      wmanIfBsDynamicServiceFailReason  OCTET STRING,
4      wmanIfBsSsRssiStatus          INTEGER,
5      wmanIfBsSsRssiStatusInfo      OCTET STRING
6      }
7
8      wmanIfBsTrapSsId OBJECT-TYPE
9          SYNTAX      Unsigned32 (1 .. 4294967295)
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "wmanIfBsTrapSsId identifies the entry in
14             wmanIfBsSsNotificationObjectsTable."
15         ::= { wmanIfBsSsNotificationObjectsEntry 1 }
16
17     wmanIfBsSsStatusValue OBJECT-TYPE
18         SYNTAX      INTEGER {ssRangingSucc(1),
19                             ssRangingFail(2),
20                             ssRegistered(3),
21                             ssRegistrationFail(4),
22                             ssDeregistered(5),
23                             ssBasicCapabilitySucc(6),
24                             ssBasicCapabilityFail(7),
25                             ssAuthorizationSucc(8),
26                             ssAuthorizationFail(9),
27                             tftpSucc(10),
28                             tftpFail(11),
29                             sfCreationSucc(12),
30                             sfCreationFail(13)
31         }
32     }
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36         "This object indicates the status of a SS, as it goes
37         through network entry and initialization procedure."
38     ::= { wmanIfBsSsNotificationObjectsEntry 2 }
39
40     wmanIfBsSsStatusInfo OBJECT-TYPE
41         SYNTAX      OCTET STRING
42         MAX-ACCESS  read-only
43         STATUS      current
44         DESCRIPTION
45             "This object indicates the reason of SS's status change."
46     ::= { wmanIfBsSsNotificationObjectsEntry 3 }
47
48     wmanIfBsDynamicServiceType OBJECT-TYPE
49         SYNTAX      INTEGER {bsSfCreationReq(1),
50                             bsSfCreationRsp(2),
51                             bsSfCreationAck(3)
52         }
53     }
54     MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "This object indicates the dynamic service flow
4          creation command type."
5      ::= { wmanIfBsSsNotificationObjectsEntry 4 }
6
7      wmanIfBsDynamicServiceFailReason OBJECT-TYPE
8          SYNTAX      OCTET STRING
9          MAX-ACCESS  read-only
10         STATUS      current
11         DESCRIPTION
12             "This object indicates the reason why the service flow
13             cration has failed."
14         ::= { wmanIfBsSsNotificationObjectsEntry 5 }
15
16         wmanIfBsSsRssiStatus OBJECT-TYPE
17             SYNTAX      INTEGER {bsRssiAlarm(1),
18                             bsRssiNoAlarm(2)
19
20                             }
21             MAX-ACCESS  read-only
22             STATUS      current
23             DESCRIPTION
24                 "A RSSI alarm is generated if the RSSI is lower than
25                 wmanIfBsLowRssiThreshold."
26             ::= { wmanIfBsSsNotificationObjectsEntry 6 }
27
28         wmanIfBsSsRssiStatusInfo OBJECT-TYPE
29             SYNTAX      OCTET STRING
30             MAX-ACCESS  read-only
31             STATUS      current
32             DESCRIPTION
33                 "This object indicates the reason why RSSI alarm is
34                 generated."
35             ::= { wmanIfBsSsNotificationObjectsEntry 7 }
36
37         --
38         -- Subscriber station Notification Trap Definitions
39         --
40         wmanBsSsStatusNotificationTrap NOTIFICATION-TYPE
41             OBJECTS      {ifIndex,
42                             wmanIfBsTrapSsId,
43                             wmanIfBsSsMacAddress,
44                             wmanIfBsSsStatusValue,
45                             wmanIfBsSsStatusInfo
46                             }
47             STATUS      current
48             DESCRIPTION
49                 "This trap reports the status of a SS. Based on this
50                 notification the NMS will issue an alarm with certain
51                 severity depending on the status and the reason received."
52             ::= { wmanIfBsTrapDefinitions 2 }
53
54         wmanBsSsDynamicServiceFailTrap NOTIFICATION-TYPE

```



```

1      OBJECTS      {ifIndex,
2                    wmanIfBsTrapSsId,
3                    wmanIfBsSsMacAddress,
4                    wmanIfBsDynamicServiceType,
5                    wmanIfBsDynamicServiceFailReason
6                    }
7      STATUS      current
8      DESCRIPTION
9          "An event to report the failure of a dynamic service
10         operation happened during the dynamic services process
11         and detected in the Bs side."
12      ::= { wmanIfBsTrapDefinitions 3 }
13
14  wmanBSsSRssiStatusChangeTrap NOTIFICATION-TYPE
15      OBJECTS      {ifIndex,
16                    wmanIfBsTrapSsId,
17                    wmanIfBsSsMacAddress,
18                    wmanIfBSsSRssiStatus,
19                    wmanIfBSsSRssiStatusInfo
20                    }
21      STATUS      current
22      DESCRIPTION
23          "An event to report that the uplink RSSI is below or above
24         (after alarm) wmanIfBsLowRssiThreshold."
25      ::= { wmanIfBsTrapDefinitions 4 }
26
27  --
28  -- Base station Notification Trap Definitions
29  --
30  wmanBSPowerStatusChangeTrap NOTIFICATION-TYPE
31      OBJECTS      {wmanIfBsPowerStatus,
32                    wmanIfBsPowerStatusInfo
33                    }
34      STATUS      current
35      DESCRIPTION
36          "An event to report a change in the status of the power
37         supply in BS. Typically it represents a failure."
38      ::= { wmanIfBsTrapDefinitions 5 }
39
40  wmanBSFanStatusTrap NOTIFICATION-TYPE
41      OBJECTS      {wmanIfBsFanStatus,
42                    wmanIfBsFanStatusInfo
43                    }
44      STATUS      current
45      DESCRIPTION
46          "An event to report the status of the fan inside the BS."
47      ::= { wmanIfBsTrapDefinitions 6 }
48
49  wmanBSTemperatureChangeTrap NOTIFICATION-TYPE
50      OBJECTS      {wmanIfBsTemperatureStatus,
51                    wmanIfBsTemperatureStatusInfo
52                    }
53      STATUS      current
54      DESCRIPTION

```

```

1         "An alarm event will be generated when the temperature goes
2         above wmanIfBsTempHighAlarmThreshold or below
3         wmanIfBsTempLowAlarmThreshold. An event reporting the alarm
4         has disappeared when the temperature goes below
5         wmanIfBsTempHighAlarmRestoredThreshold or above
6         wmanIfBsTempLowAlarmRestoredThreshold."
7     ::= { wmanIfBsTrapDefinitions 7 }
8
9     --
10    -- SS object group - containing tables and objects to be implemented in
11    -- the Subscriber station
12    --
13    -- wmanIfSsSystem contain the Subscriber Station System objects
14    wmanIfSsSystem OBJECT IDENTIFIER ::= { wmanIfSsObjects 1 }
15
16    wmanIfSsConfigFileEncodingTable OBJECT-TYPE
17        SYNTAX      SEQUENCE OF wmanIfSsConfigFileEncodingEntry
18        MAX-ACCESS  not-accessible
19        STATUS      current
20        DESCRIPTION
21            "This table contains configuration file encoding
22            information of the SS."
23        REFERENCE
24            "Section 11.2 in IEEE 802.16REvD/D3-2004"
25        ::= { wmanIfSsSystem 1 }
26
27    wmanIfSsConfigFileEncodingEntry OBJECT-TYPE
28        SYNTAX      wmanIfSsConfigFileEncodingEntry
29        MAX-ACCESS  not-accessible
30        STATUS      current
31        DESCRIPTION
32            "This table provides one row for each SS, and is indexed
33            by wmanIfSsIdIndex."
34        INDEX { wmanIfSsIdIndex }
35        ::= { wmanIfSsConfigFileEncodingTable 1 }
36
37    wmanIfSsConfigFileEncodingEntry ::= SEQUENCE {
38        wmanIfSsIdIndex          Unsigned32,
39        wmanIfSsMicConfigSetting OCTET STRING,
40        wmanIfSsVendorId        OCTET STRING,
41        wmanIfSsHwId            OCTET STRING,
42        wmanIfSsSwVersion       OCTET STRING,
43        wmanIfSsUpgradeFileName  OCTET STRING,
44        wmanIfSsSwUpgradeTftpServer InetAddress,
45        wmanIfSsTftpServerTimeStamp DateAndTime
46    }
47
48    wmanIfSsIdIndex OBJECT-TYPE
49        SYNTAX      Unsigned32 (1 .. 4294967295)
50        MAX-ACCESS  read-only
51        STATUS      current
52        DESCRIPTION
53            "wmanIfSsIdIndex identifies the SS that is registered."
54        REFERENCE

```

```

1         "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
2         ::= { wmanIfSsConfigFileEncodingEntry 1 }
3
4     wmanIfSsMicConfigSetting OBJECT-TYPE
5         SYNTAX      OCTET STRING (SIZE(16))
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "The value field contains the SS MIC code. This is used
10            to detect unauthorized modification or corruption of
11            the configuration file."
12         ::= { wmanIfSsConfigFileEncodingEntry 2 }
13
14     wmanIfSsVendorId OBJECT-TYPE
15         SYNTAX      OCTET STRING (SIZE(3))
16         MAX-ACCESS  read-only
17         STATUS      current
18         DESCRIPTION
19             "This value identifies the managed SS vendor to which the
20            software upgrade is to be applied."
21         ::= { wmanIfSsConfigFileEncodingEntry 3 }
22
23     wmanIfSsHwId OBJECT-TYPE
24         SYNTAX      OCTET STRING
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "This value identifies the hardware version to which the
29            software upgrade is to be applied."
30         ::= { wmanIfSsConfigFileEncodingEntry 4 }
31
32     wmanIfSsSwVersion OBJECT-TYPE
33         SYNTAX      OCTET STRING
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "The value field contains the SS MIC code. This is used
38            to detect unauthorized modification or corruption of
39            the configuration file."
40         ::= { wmanIfSsConfigFileEncodingEntry 5 }
41
42     wmanIfSsUpgradeFileName OBJECT-TYPE
43         SYNTAX      OCTET STRING
44         MAX-ACCESS  read-only
45         STATUS      current
46         DESCRIPTION
47             "The filename is a fully qualified directory path
48            name which is in a format appropriate to the server."
49         ::= { wmanIfSsConfigFileEncodingEntry 6 }
50
51     wmanIfSsSwUpgradeTftpServer OBJECT-TYPE
52         SYNTAX      InetAddress
53         MAX-ACCESS  read-only
54         STATUS      current

```

```

1      DESCRIPTION
2          "This object is the IP address of the TFTP server on
3          which the software upgrade file for the SS resides."
4      ::= { wmanIfSsConfigFileEncodingEntry 7 }
5
6      wmanIfSsTftpServerTimeStamp OBJECT-TYPE
7          SYNTAX      DateAndTime
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "This is the sending time of the configuration file in
12             seconds. The definition of time is as in IETF RFC 868."
13         ::= { wmanIfSsConfigFileEncodingEntry 8 }
14
15     --
16     -- wmanIfSsCps contain the Base Station Common Part Sublayer objects
17     wmanIfSsCps OBJECT IDENTIFIER ::= { wmanIfSsObjects 2 }
18
19     --
20     -- wmanIfSsConfigurationTable contains global parameters common
21     -- in BS and SS
22     wmanIfSsConfigurationTable OBJECT-TYPE
23         SYNTAX      SEQUENCE OF wmanIfSsConfigurationEntry
24         MAX-ACCESS  not-accessible
25         STATUS      current
26         DESCRIPTION
27             "This table provides one row for each SS that contains
28             the system parameters as defined in section 10.1 of [3]."
29         ::= { wmanIfSsCps 1 }
30
31     wmanIfSsConfigurationEntry OBJECT-TYPE
32         SYNTAX      wmanIfSsConfigurationEntry
33         MAX-ACCESS  not-accessible
34         STATUS      current
35         DESCRIPTION
36             "This table is indexed by wmanIfCmnSsIdIndex."
37         INDEX { ifIndex }
38         ::= { wmanIfSsConfigurationTable 1 }
39
40     wmanIfSsConfigurationEntry ::= SEQUENCE {
41         wmanIfSsId                               Unsigned32,
42         wmanIfSsLostDLMapInterval                INTEGER,
43         wmanIfSsLostULMapInterval                INTEGER,
44         wmanIfSsContentionRangRetries            INTEGER,
45         wmanIfSsRequestRetries                  INTEGER,
46         wmanIfSsRegRequestRetries                INTEGER,
47         wmanIfSsTftpBackoffStart                 INTEGER,
48         wmanIfSsTftpBackoffEnd                  INTEGER,
49         wmanIfSsTftpRequestRetries               INTEGER,
50         wmanIfSsTftpDownloadRetries              INTEGER,
51         wmanIfSsTftpWait                         INTEGER,
52         wmanIfSsToDRetries                       INTEGER,
53         wmanIfSsToDRetryPeriod                   INTEGER,
54         wmanIfSsT1Timeout                        INTEGER,

```

```

1      wmanIfSsT2Timeout          INTEGER,
2      wmanIfSsT3Timeout          INTEGER,
3      wmanIfSsT4Timeout          INTEGER,
4      wmanIfSsT6Timeout          INTEGER,
5      wmanIfSsT12Timeout         INTEGER,
6      wmanIfSsT14Timeout         INTEGER,
7      wmanIfSsT16Timeout         INTEGER,
8      wmanIfSsT18Timeout         INTEGER,
9      wmanIfSsT19Timeout         INTEGER,
10     wmanIfSsT20Timeout          INTEGER,
11     wmanIfSsT21Timeout          INTEGER,
12     wmanIfSsSBCRequestRetries  INTEGER,
13     wmanIfSsTftpCpltRetries    INTEGER,
14     wmanIfSsT26Timeout          INTEGER,
15     wmanIfSsDLManagProcTime    INTEGER,
16     wmanIfSsRssiLowThreshold    INTEGER,
17     wmanIfSsRssiHighThreshold  INTEGER,
18     wmanIfSsConfigurationRowStatus RowStatus
19     }
20
21     wmanIfSsId OBJECT-TYPE
22         SYNTAX      Unsigned32 (1 .. 4294967295)
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "wmanIfSsId is the index to
27             wmanIfSsConfigurationTable."
28         REFERENCE
29             "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
30         ::= { wmanIfSsConfigurationEntry 1 }
31
32     wmanIfSsLostDLMapInterval OBJECT-TYPE
33         SYNTAX      INTEGER(0..600)
34         UNITS       "milliseconds"
35         MAX-ACCESS  read-write
36         STATUS      current
37         DESCRIPTION
38             "Time since last received DL-MAP message before downlink
39             synchronization is considered lost in ms."
40         ::= { wmanIfSsConfigurationEntry 2 }
41
42     wmanIfSsLostULMapInterval OBJECT-TYPE
43         SYNTAX      INTEGER(0..600)
44         UNITS       "milliseconds"
45         MAX-ACCESS  read-write
46         STATUS      current
47         DESCRIPTION
48             "Time since last received UL-MAP message before downlink
49             synchronization is considered lost in ms."
50         ::= { wmanIfSsConfigurationEntry 3 }
51
52     wmanIfSsContentionRangRetries OBJECT-TYPE
53         SYNTAX      INTEGER(16..65535)
54         MAX-ACCESS  read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          "Number of retries on contention Ranging Requests."
4      ::= { wmanIfsConfigurationEntry 4 }
5
6      wmanIfsRequestRetries OBJECT-TYPE
7          SYNTAX      INTEGER(16..65535)
8          MAX-ACCESS  read-write
9          STATUS      current
10         DESCRIPTION
11             "Number of retries on bandwidth allocation requests."
12         ::= { wmanIfsConfigurationEntry 5 }
13
14         wmanIfsRegRequestRetries OBJECT-TYPE
15             SYNTAX      INTEGER(3..65535)
16             MAX-ACCESS  read-write
17             STATUS      current
18             DESCRIPTION
19                 "Number of retries on registration requests."
20             ::= { wmanIfsConfigurationEntry 6 }
21
22         wmanIfsTftpBackoffStart OBJECT-TYPE
23             SYNTAX      INTEGER(1..65535)
24             UNITS       "seconds"
25             MAX-ACCESS  read-write
26             STATUS      current
27             DESCRIPTION
28                 "Initial value for TFTP backoff in s."
29             ::= { wmanIfsConfigurationEntry 7 }
30
31         wmanIfsTftpBackoffEnd OBJECT-TYPE
32             SYNTAX      INTEGER(16..65535)
33             UNITS       "seconds"
34             MAX-ACCESS  read-write
35             STATUS      current
36             DESCRIPTION
37                 "Last value for TFTP backoff in s."
38             ::= { wmanIfsConfigurationEntry 8 }
39
40         wmanIfsTftpRequestRetries OBJECT-TYPE
41             SYNTAX      INTEGER(16..65535)
42             MAX-ACCESS  read-write
43             STATUS      current
44             DESCRIPTION
45                 "Number of retries on TFTP request."
46             ::= { wmanIfsConfigurationEntry 9 }
47
48         wmanIfsTftpDownloadRetries OBJECT-TYPE
49             SYNTAX      INTEGER(3..65535)
50             MAX-ACCESS  read-write
51             STATUS      current
52             DESCRIPTION
53                 "Number of retries on entire TFTP downloads."
54             ::= { wmanIfsConfigurationEntry 10 }

```

```
1
2 wmanIfsStftpwait OBJECT-TYPE
3     SYNTAX      INTEGER(2..65535)
4     UNITS       "minutes"
5     MAX-ACCESS  read-write
6     STATUS      current
7     DESCRIPTION
8         "The duration between two consecutive TFTP retries in min."
9     ::= { wmanIfsConfigurationEntry 11 }
10
11 wmanIfsStoDRetries OBJECT-TYPE
12     SYNTAX      INTEGER(3..65535)
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Number of Retries per Time of Day Retry Period."
17     ::= { wmanIfsConfigurationEntry 12 }
18
19 wmanIfsStoDRetryPeriod OBJECT-TYPE
20     SYNTAX      INTEGER(5..65535)
21     MAX-ACCESS  read-write
22     STATUS      current
23     DESCRIPTION
24         "Time of Day Retry Period."
25     ::= { wmanIfsConfigurationEntry 13 }
26
27 wmanIfsSt1Timeout OBJECT-TYPE
28     SYNTAX      INTEGER(0..65535)
29     UNITS       "milliseconds"
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "Wait for DCD timeout in ms."
34     ::= { wmanIfsConfigurationEntry 14 }
35
36 wmanIfsSt2Timeout OBJECT-TYPE
37     SYNTAX      INTEGER(0..65535)
38     UNITS       "milliseconds"
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "Wait for broadcast ranging timeout in ms."
43     ::= { wmanIfsConfigurationEntry 15 }
44
45 wmanIfsSt3Timeout OBJECT-TYPE
46     SYNTAX      INTEGER(0..200)
47     UNITS       "milliseconds"
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51         "Ranging Response reception timeout following the
52         transmission of a Ranging Request in ms."
53     ::= { wmanIfsConfigurationEntry 16 }
54
```

```

1  wmanIfsSt4Timeout OBJECT-TYPE
2      SYNTAX      INTEGER(30..35)
3      UNITS       "seconds"
4      MAX-ACCESS  read-write
5      STATUS      current
6      DESCRIPTION
7          "wait for unicast ranging opportunity. If the pending until
8          complete field was used earlier by this SS, then the value
9          of that field shall be added to this interval in s."
10     ::= { wmanIfsConfigurationEntry 17 }
11
12  wmanIfsSt6Timeout OBJECT-TYPE
13      SYNTAX      INTEGER(0..3000)
14      UNITS       "milliseconds"
15      MAX-ACCESS  read-write
16      STATUS      current
17      DESCRIPTION
18          "wait for registration response in ms."
19     ::= { wmanIfsConfigurationEntry 18 }
20
21  wmanIfsSt12Timeout OBJECT-TYPE
22      SYNTAX      INTEGER
23      UNITS       "milliseconds"
24      MAX-ACCESS  read-write
25      STATUS      current
26      DESCRIPTION
27          "wait for UCD descriptor in ms."
28     ::= { wmanIfsConfigurationEntry 19 }
29
30  wmanIfsSt14Timeout OBJECT-TYPE
31      SYNTAX      INTEGER(0..200)
32      UNITS       "milliseconds"
33      MAX-ACCESS  read-write
34      STATUS      current
35      DESCRIPTION
36          "wait for DSX-RVD Timeout in ms."
37     ::= { wmanIfsConfigurationEntry 20 }
38
39  wmanIfsSt16Timeout OBJECT-TYPE
40      SYNTAX      INTEGER(10..65535)
41      UNITS       "milliseconds"
42      MAX-ACCESS  read-write
43      STATUS      current
44      DESCRIPTION
45          "wait for bandwidth request grant in ms."
46     ::= { wmanIfsConfigurationEntry 21 }
47
48  wmanIfsSt18Timeout OBJECT-TYPE
49      SYNTAX      INTEGER(0..65535)
50      UNITS       "milliseconds"
51      MAX-ACCESS  read-write
52      STATUS      current
53      DESCRIPTION
54          "wait for SBC-RSP timeout in ms."

```



```
1      ::= { wmanIfsConfigurationEntry 22 }
2
3  wmanIfsSt19Timeout OBJECT-TYPE
4      SYNTAX      INTEGER(0..65535)
5      UNITS       "milliseconds"
6      MAX-ACCESS  read-write
7      STATUS      current
8      DESCRIPTION
9          "Time DL-channel remains unusable in ms."
10     ::= { wmanIfsConfigurationEntry 23 }
11
12  wmanIfsSt20Timeout OBJECT-TYPE
13      SYNTAX      INTEGER(0..65535)
14      UNITS       "milliseconds"
15      MAX-ACCESS  read-write
16      STATUS      current
17      DESCRIPTION
18          "Time SS searches for preambles on a given channel in ms."
19     ::= { wmanIfsConfigurationEntry 24 }
20
21  wmanIfsSt21Timeout OBJECT-TYPE
22      SYNTAX      INTEGER(0..10000)
23      UNITS       "milliseconds"
24      MAX-ACCESS  read-write
25      STATUS      current
26      DESCRIPTION
27          "Time SS searches for DL-MAP on a given channel in ms."
28     ::= { wmanIfsConfigurationEntry 25 }
29
30  wmanIfsSsSBCRequestRetries OBJECT-TYPE
31      SYNTAX      INTEGER(3..16)
32      MAX-ACCESS  read-write
33      STATUS      current
34      DESCRIPTION
35          "Number of retries on SBC Request."
36     ::= { wmanIfsConfigurationEntry 26 }
37
38  wmanIfsStftpCpltRetries OBJECT-TYPE
39      SYNTAX      INTEGER(3..16)
40      MAX-ACCESS  read-write
41      STATUS      current
42      DESCRIPTION
43          "Number of retries on TFTP-CPLT."
44     ::= { wmanIfsConfigurationEntry 27 }
45
46  wmanIfsSt26Timeout OBJECT-TYPE
47      SYNTAX      INTEGER(10..200)
48      UNITS       "milliseconds"
49      MAX-ACCESS  read-write
50      STATUS      current
51      DESCRIPTION
52          "Wait for TFTP-RSP in ms."
53     ::= { wmanIfsConfigurationEntry 28 }
54
```

```

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

```

```

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

```

```

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

```

```

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

```

```

1      ::= { wmanIfSsPkmBaseEntry 10 }
2
3  wmanIfSsPkmReauthWaitTimeout OBJECT-TYPE
4      SYNTAX      Integer32 (2..30)
5      UNITS       "seconds"
6      MAX-ACCESS  read-only
7      STATUS      current
8      DESCRIPTION
9          "The value of this object is the Reauthorize Wait Timeout
10         in seconds."
11     REFERENCE
12         "IEEE 802.16 standard; Table 270"
13     DEFVAL      { 10 }
14     ::= { wmanIfSsPkmBaseEntry 11 }
15
16  wmanIfSsPkmOpWaitTimeout OBJECT-TYPE
17     SYNTAX      Integer32 (1..10)
18     UNITS       "seconds"
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "The value of this object is the Operational Wait Timeout
23         in seconds."
24     REFERENCE
25         "IEEE 802.16 standard; Table 270"
26     DEFVAL      { 1 }
27     ::= { wmanIfSsPkmBaseEntry 12 }
28
29  wmanIfSsPkmRekeyWaitTimeout OBJECT-TYPE
30     SYNTAX      Integer32 (1..10)
31     UNITS       "seconds"
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "The value of this object is the Rekey Wait Timeout in
36         seconds."
37     REFERENCE
38         "IEEE 802.16 standard; Table 270"
39     DEFVAL      { 1 }
40     ::= { wmanIfSsPkmBaseEntry 13 }
41
42  wmanIfSsPkmAuthRejectWaitTimeout OBJECT-TYPE
43     SYNTAX      Integer32 (10..600)
44     UNITS       "seconds"
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "The value of this object is the Authorization Reject Wait
49         Timeout in seconds."
50     REFERENCE
51         "IEEE 802.16 standard; Table 270"
52     DEFVAL      { 60 }
53     ::= { wmanIfSsPkmBaseEntry 14 }
54

```

```

1  wmanIfSsPkmAuthentInfos OBJECT-TYPE
2      SYNTAX          Counter32
3      MAX-ACCESS     read-only
4      STATUS          current
5      DESCRIPTION
6          "The value of this object is the count of times the SS has
7              transmitted an Authentication Information message."
8      ::= { wmanIfSsPkmBaseEntry 15 }
9
10 wmanIfSsPkmAuthRequests OBJECT-TYPE
11     SYNTAX          Counter32
12     MAX-ACCESS     read-only
13     STATUS          current
14     DESCRIPTION
15         "The value of this object is the count of times the SS has
16             transmitted an Authorization Request message."
17     ::= { wmanIfSsPkmBaseEntry 16 }
18
19 wmanIfSsPkmAuthReplies OBJECT-TYPE
20     SYNTAX          Counter32
21     MAX-ACCESS     read-only
22     STATUS          current
23     DESCRIPTION
24         "The value of this object is the count of times the SS has
25             received an Authorization Reply message."
26     ::= { wmanIfSsPkmBaseEntry 17 }
27
28 wmanIfSsPkmAuthRejects OBJECT-TYPE
29     SYNTAX          Counter32
30     MAX-ACCESS     read-only
31     STATUS          current
32     DESCRIPTION
33         "The value of this object is the count of times the SS has
34             received an Authorization Reject message."
35     ::= { wmanIfSsPkmBaseEntry 18 }
36
37 wmanIfSsPkmAuthInvalids OBJECT-TYPE
38     SYNTAX          Counter32
39     MAX-ACCESS     read-only
40     STATUS          current
41     DESCRIPTION
42         "The value of this object is the count of times the SS has
43             received an Authorization Invalid message."
44     ::= { wmanIfSsPkmBaseEntry 19 }
45
46 wmanIfSsPkmAuthRejectErrorCode OBJECT-TYPE
47     SYNTAX          INTEGER {none(1),
48                             unknown(2),
49                             unauthorizedSs(3),
50                             unauthorizedSaid(4),
51                             permanentAuthorizationFailure(8),
52                             timeOfDayNotAcquired(11)}
53     MAX-ACCESS     read-only
54     STATUS          current

```

```

1      DESCRIPTION
2          "The value of this object is the enumerated description of
3          the Error-Code in most recent Authorization Reject message
4          received by the SS. This has value unknown(2)if the last
5          Error-Code value was 0, and none(1) if no Authorization
6          Reject message has been received since reboot."
7      ::= { wmanIfSsPkmBaseEntry 20 }
8
9      wmanIfSsPkmAuthRejectErrorString OBJECT-TYPE
10     SYNTAX      SnmpAdminString (SIZE (0..128))
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "The value of this object is the Display-String in most
15         recent Authorization Reject message received by the SS.
16         This is a zero length string if no Authorization Reject
17         message has been received since reboot."
18     ::= { wmanIfSsPkmBaseEntry 21 }
19
20     wmanIfSsPkmAuthInvalidErrorCode OBJECT-TYPE
21     SYNTAX      INTEGER {none(1),
22                 unknown(2),
23                 unauthorizedSs(3),
24                 unsolicited(5),
25                 invalidKeySequence(6),
26                 keyRequestAuthenticationFailure(7)}
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "The value of this object is the enumerated description of
31         the Error-Code in most recent Authorization Invald message
32         received by the SS. This has value unknown(2) if the last
33         Error-Code value was 0, and none(1) if no Authorization
34         Invalid message has been received since reboot."
35     ::= { wmanIfSsPkmBaseEntry 22 }
36
37     wmanIfSsPkmAuthInvalidErrorString OBJECT-TYPE
38     SYNTAX      SnmpAdminString (SIZE (0..128))
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42         "The value of this object is the Display-String in most
43         recent Authorization Invalid message received by the SS.
44         This is a zero length string if no Authorization Invalid
45         message has been received since reboot."
46     ::= { wmanIfSsPkmBaseEntry 23 }
47
48     --
49     -- Table wmanIfSsPkmTEKTable
50     --
51     wmanIfSsPkmTEKTable OBJECT-TYPE
52     SYNTAX      SEQUENCE OF      wmanIfSsPkmTEKEntry
53     MAX-ACCESS  not-accessible
54     STATUS      current

```



```

1      DESCRIPTION
2          "This table describes the attributes of each SS Traffic
3          Encryption Key(TEK) association. The SS maintains (no more
4          than) one TEK association per SAID per SS wireless
5          interface."
6      ::= { wmanIfSsPkmObjects 2 }
7
8      wmanIfSsPkmTEKEntry OBJECT-TYPE
9          SYNTAX      WmanIfSsPkmTEKEntry
10         MAX-ACCESS  not-accessible
11         STATUS      current
12         DESCRIPTION
13             "Each entry contains objects describing the TEK association
14             attributes of one SAID. The SS MUST create one entry per
15             SAID, regardless of whether the SAID was obtained from a
16             Registration Response message, from an Authorization Reply
17             message, or from any dynamic SAID establishment
18             mechanisms."
19         INDEX       { ifIndex, wmanIfSsPkmTEKSAID }
20         ::= { wmanIfSsPkmTEKTable 1 }
21
22     wmanIfSsPkmTEKEntry ::= SEQUENCE {
23         wmanIfSsPkmTEKSAID          Integer32,
24         wmanIfSsPkmTEKSAType        INTEGER,
25         wmanIfSsPkmTEKDataEncryptAlg INTEGER,
26         wmanIfSsPkmTEKDataAuthentAlg INTEGER,
27         wmanIfSsPkmTEKEncryptAlg    INTEGER,
28         wmanIfSsPkmTEKState          INTEGER,
29         wmanIfSsPkmTEKKeySequenceNumber Integer32,
30         wmanIfSsPkmTEKExpiresOld     DateAndTime,
31         wmanIfSsPkmTEKExpiresNew     DateAndTime,
32         wmanIfSsPkmTEKKeyRequests    Counter32,
33         wmanIfSsPkmTEKKeyReplies     Counter32,
34         wmanIfSsPkmTEKKeyRejects     Counter32,
35         wmanIfSsPkmTEKInvalids       Counter32,
36         wmanIfSsPkmTEKAuthPends      Counter32,
37         wmanIfSsPkmTEKKeyRejectErrorCode INTEGER,
38         wmanIfSsPkmTEKKeyRejectErrorString SnmpAdminString,
39         wmanIfSsPkmTEKInvalidErrorCode INTEGER,
40         wmanIfSsPkmTEKInvalidErrorString SnmpAdminString
41     }
42
43     wmanIfSsPkmTEKSAID OBJECT-TYPE
44         SYNTAX      Integer32 (1..16383)
45         MAX-ACCESS  not-accessible
46         STATUS      current
47         DESCRIPTION
48             "The value of this object is the WiMAX Security Association
49             ID (SAID)."
50         ::= { wmanIfSsPkmTEKEntry 1 }
51
52     wmanIfSsPkmTEKSAType OBJECT-TYPE
53         SYNTAX      INTEGER {none(0),
54                     primary(1),

```

```

1          static(2),
2          dynamic(3)}
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The value of this object is the type of security
7           association."
8      ::= { wmanIfSsPkmTEKEntry 2 }
9
10     wmanIfSsPkmTEKDataEncryptAlg OBJECT-TYPE
11         SYNTAX      INTEGER { none(0),
12                        des56CbcMode(1) }
13         MAX-ACCESS  read-only
14         STATUS      current
15         DESCRIPTION
16             "The value of this object is the data encryption algorithm
17              being utilized."
18         REFERENCE
19             "IEEE 802.16 standard; Table 301"
20         ::= { wmanIfSsPkmTEKEntry 3 }
21
22     wmanIfSsPkmTEKDataAuthentAlg OBJECT-TYPE
23         SYNTAX      INTEGER { none(0) }
24         MAX-ACCESS  read-only
25         STATUS      current
26         DESCRIPTION
27             "The value of this object is the data authentication
28              algorithm being utilized."
29         REFERENCE
30             "IEEE 802.16 standard; Table 302"
31         ::= { wmanIfSsPkmTEKEntry 4 }
32
33     wmanIfSsPkmTEKEncryptAlg OBJECT-TYPE
34         SYNTAX      INTEGER { tripleDES(0),
35                        rsa1024(1) }
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "The value of this object is the TEK key encryption
40              algorithm for this cryptographic suite capability."
41         REFERENCE
42             "IEEE 802.16 standard; Table 303"
43         ::= { wmanIfSsPkmTEKEntry 5 }
44
45     wmanIfSsPkmTEKState OBJECT-TYPE
46         SYNTAX      INTEGER { start(1),
47                        opWait(2),
48                        opReauthWait(3),
49                        operational(4),
50                        rekeyWait(5),
51                        rekeyReauthWait(6) }
52         MAX-ACCESS  read-only
53         STATUS      current
54         DESCRIPTION

```

```

1           "The value of this object is the state of the indicated TEK
2           FSM. The start(1) state indicates that FSM is in its
3           initial state."
4       ::= { wmanIfSsPkmTEKEntry 6 }
5
6   wmanIfSsPkmTEKKeySequenceNumber OBJECT-TYPE
7       SYNTAX      Integer32 (0..15)
8       MAX-ACCESS  read-only
9       STATUS      current
10      DESCRIPTION
11          "The value of this object is the most recent TEK key
12          sequence number for this TEK FSM."
13      ::= { wmanIfSsPkmTEKEntry 7 }
14
15   wmanIfSsPkmTEKExpiresOld OBJECT-TYPE
16       SYNTAX      DateAndTime
17       MAX-ACCESS  read-only
18       STATUS      current
19       DESCRIPTION
20          "The value of this object is the actual clock time for
21          expiration of the immediate predecessor of the most recent
22          TEK for this FSM. If this FSM has only one TEK, then the
23          value is the time of activation of this FSM."
24      ::= { wmanIfSsPkmTEKEntry 8 }
25
26   wmanIfSsPkmTEKExpiresNew OBJECT-TYPE
27       SYNTAX      DateAndTime
28       MAX-ACCESS  read-only
29       STATUS      current
30       DESCRIPTION
31          "The value of this object is the actual clock time for
32          expiration of the most recent TEK for this FSM."
33      ::= { wmanIfSsPkmTEKEntry 9 }
34
35   wmanIfSsPkmTEKKeyRequests 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 SS has
41          transmitted a Key Request message."
42      ::= { wmanIfSsPkmTEKEntry 10 }
43
44   wmanIfSsPkmTEKKeyReplies 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 SS has
50          received a Key Reply message, including a message whose
51          authentication failed."
52      ::= { wmanIfSsPkmTEKEntry 11 }
53
54   wmanIfSsPkmTEKKeyRejects OBJECT-TYPE

```

```

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

```

```

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

```

```

1      STATUS      current
2      DESCRIPTION
3          "The X509 DER-encoded subscriber station certificate."
4      ::= { wmanIfSsDeviceCertEntry 1 }
5
6      wmanIfSsDeviceManufCert OBJECT-TYPE
7          SYNTAX      OCTET STRING
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "The X509 DER-encoded manufacturer certificate which is
12             signed by the CA root authority certificate."
13         ::= { wmanIfSsDeviceCertEntry 2 }
14
15     --
16     -- Subscriber station Notification Group
17     -- wmanIfSsNotificationObjects contains the SS SNMP Trap objects
18     --
19     wmanIfSsNotification OBJECT IDENTIFIER ::= { wmanIfSsObjects 4 }
20     wmanIfSsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIfSsNotification 1 }
21     wmanIfSsTrapControl OBJECT IDENTIFIER ::= { wmanIfSsNotification 2 }
22
23     wmanIfSsTrapControlRegister OBJECT-TYPE
24         SYNTAX      BITS {wmanSsTLVUnknown(0),
25                         wmanSsDynamicServiceFail(1),
26                         wmanSsBPKMFail(2),
27                         wmanSsDHCPSuccess(3),
28                         wmanSsRssiStatusChange(4)
29                         }
30         MAX-ACCESS  read-write
31         STATUS      current
32         DESCRIPTION
33             "The object is used to enable Subscriber Station traps.
34             From left to right, the set bit indicates the corresponding
35             Subscriber Station trap is enabled."
36         ::= { wmanIfSsTrapControl 1 }
37
38     wmanBstLVUnknownTrap NOTIFICATION-TYPE
39         OBJECTS      {wmanIfSsMacAddress,
40                     wmanIfSsUnknownTlv
41                     }
42         STATUS      current
43         DESCRIPTION
44             "Event that notifies detection of unknown TLV during
45             the
46             TLV parsing process."
47         ::= { wmanIfSsTrapDefinitions 1 }
48
49     wmanSsDynamicServiceFailTrap NOTIFICATION-TYPE
50         OBJECTS      {wmanIfSsMacAddress,
51                     wmanIfSsDynamicServiceType,
52                     wmanIfSsDynamicServiceFailReason
53                     }
54         STATUS      current

```

```

1      DESCRIPTION
2          "An event to report the failure of a dynamic service
3          operation happened during the dynamic services process
4          and detected in the BS side."
5      ::= { wmanIfSsTrapDefinitions 2 }
6
7      wmanSsBPKMFailTrap NOTIFICATION-TYPE
8          STATUS      current
9          DESCRIPTION
10             "An event to report the failure of a BPKM operation."
11         ::= { wmanIfSsTrapDefinitions 3 }
12
13         wmanSsDHCPSuccessTrap NOTIFICATION-TYPE
14             STATUS      current
15             DESCRIPTION
16                 "An event to report a successful DHCP Handshake for
17                 the SS."
18             ::= { wmanIfSsTrapDefinitions 4 }
19
20         wmanSsRssiStatusChangeTrap NOTIFICATION-TYPE
21             OBJECTS      {wmanIfSsMacAddress,
22                         wmanIfSsRssiStatus,
23                         wmanIfSsRssiStatusInfo
24                         }
25             STATUS      current
26             DESCRIPTION
27                 "An event to report that the uplink RSSI is below or above
28                 (after alarm) wmanIfBsLowRssiThreshold."
29             ::= { wmanIfSsTrapDefinitions 5 }
30
31         wmanIfSsMacAddress OBJECT-TYPE
32             SYNTAX      MacAddress
33             MAX-ACCESS  read-only
34             STATUS      current
35             DESCRIPTION
36                 "The MAC address of the SS generating the trap."
37             ::= { wmanIfSsTrapDefinitions 6 }
38
39         wmanIfSsUnknownTlv OBJECT-TYPE
40             SYNTAX      INTEGER
41             MAX-ACCESS  read-only
42             STATUS      current
43             DESCRIPTION
44                 "The MAC address of the SS generating the trap."
45             ::= { wmanIfSsTrapDefinitions 7 }
46
47         wmanIfSsDynamicServiceType OBJECT-TYPE
48             SYNTAX      INTEGER {ssSfCreationReq(1),
49                               ssSfCreationRsp(2),
50                               ssSfCreationAck(3)
51                               }
52             MAX-ACCESS  read-only
53             STATUS      current

```

```

1      DESCRIPTION
2          "This object indicates the dynamic service flow
3          creation command type."
4      ::= { wmanIfsTrapDefinitions 8 }
5
6      wmanIfsDynamicServiceFailReason OBJECT-TYPE
7          SYNTAX      OCTET STRING
8          MAX-ACCESS  read-only
9          STATUS      current
10         DESCRIPTION
11             "This object indicates the reason why the service flow
12             cration has failed."
13         ::= { wmanIfsTrapDefinitions 9 }
14
15         wmanIfsRssiStatus OBJECT-TYPE
16             SYNTAX      INTEGER {ssRssiAlarm(1),
17                             ssRssiNoAlarm(2)
18                             }
19             MAX-ACCESS  read-only
20             STATUS      current
21             DESCRIPTION
22                 "A RSSI alarm is generated if the RSSI is lower than
23                 wmanIfBsLowRssiThreshold."
24             ::= { wmanIfsTrapDefinitions 10 }
25
26
27         wmanIfsRssiStatusInfo OBJECT-TYPE
28             SYNTAX      OCTET STRING
29             MAX-ACCESS  read-only
30             STATUS      current
31             DESCRIPTION
32                 "This object indicates the reason why RSSI alarm is
33                 generated."
34             ::= { wmanIfsTrapDefinitions 11 }
35
36         --
37         -- Common object group - containing common tables and objects to be
38         -- implemented in both Base Station and Subscriber Station
39         --
40         -- wmanIfCmnPacketCs contain the Packet Convergence Sublayer objects
41         -- that are common to both Base Station and Subscriber Station
42         wmanIfCmnPacketCs OBJECT IDENTIFIER ::= { wmanIfCommonObjects 1 }
43
44         wmanIfCmnClassifierRuleTable OBJECT-TYPE
45             SYNTAX      SEQUENCE OF wmanIfCmnClassifierRuleEntry
46             MAX-ACCESS  not-accessible
47             STATUS      current
48             DESCRIPTION
49                 "This table contains packet classifier rules associated
50                 with service flows."
51             ::= { wmanIfCmnPacketCs 1 }
52
53         wmanIfCmnClassifierRuleEntry OBJECT-TYPE
54             SYNTAX      wmanIfCmnClassifierRuleEntry

```



```

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

```

```

1      SYNTAX      Unsigned32 (1 .. 4294967295)
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "A 32 bit quantity that uniquely identifies a service flow
6          to both the subscriber station and base station (BS)."
```

::= { wmanIfCmnClassifierRuleEntry 2 }

```

8
9      wmanIfCmnClassifierRuleServiceFlowId OBJECT-TYPE
10     SYNTAX      Unsigned32 (1..4294967295)
11     MAX-ACCESS  read-write
12     STATUS      current
13     DESCRIPTION
14         "An index assigned to a service flow by SC (SFID)."
```

REFERENCE "802.16 Chapter 11.4.8"

::= { wmanIfCmnClassifierRuleEntry 3 }

```

17
18     wmanIfCmnClassifierRulePriority OBJECT-TYPE
19     SYNTAX      INTEGER
20     MAX-ACCESS  read-write
21     STATUS      current
22     DESCRIPTION
23         "The value specifies the order of evaluation of the
24         classifiers. The higher the value the higher the
25         priority. The value of 0 is used as default in
26         provisioned service flows classifiers. The default
27         value of 64 is used for dynamic service flow classifiers.
28         If the referenced parameter is not present in a classifier,
29         this object reports the default value as defined above"
```

::= { wmanIfCmnClassifierRuleEntry 4 }

```

31
32     wmanIfCmnClassifierRuleIpTosLow OBJECT-TYPE
33     SYNTAX      OCTET STRING (SIZE(1))
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37         "The low value of a range of TOS byte values. If the
38         referenced parameter is not present in a classifier, this
39         object reports the value of 0."
```

REFERENCE "802.16 Chapter 11.4.9"

::= { wmanIfCmnClassifierRuleEntry 5 }

```

42
43     wmanIfCmnClassifierRuleIpTosHigh OBJECT-TYPE
44     SYNTAX      OCTET STRING (SIZE(1))
45     MAX-ACCESS  read-write
46     STATUS      current
47     DESCRIPTION
48         "The 8-bit high value of a range of TOS byte values.
49         If the referenced parameter is not present in a classifier,
50         this object reports the value of 0."
```

REFERENCE "802.16 Chapter 11.4.9"

::= { wmanIfCmnClassifierRuleEntry 6 }

```

53
54     wmanIfCmnClassifierRuleIpTosMask OBJECT-TYPE
```

```

1      SYNTAX      OCTET STRING (SIZE(1))
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "The mask value is bitwise ANDed with TOS byte in an IP
6          packet and this value is used check range checking of
7          TosLow and TosHigh. If the referenced parameter is not
8          present in a classifier, this object reports the value
9          of 0."
10     REFERENCE   "802.16 Chapter 11.4.9"
11     ::= { wmanIfCmnClassifierRuleEntry 7 }
12
13     wmanIfCmnClassifierRuleIpProtocol OBJECT-TYPE
14     SYNTAX      Integer32 (0..255)
15     MAX-ACCESS  read-write
16     STATUS      current
17     DESCRIPTION
18         "This object indicates the value of the IP Protocol field
19         required for IP packets to match this rule. If the
20         referenced parameter is not present in a classifier, this
21         object reports the value of 0."
22     REFERENCE   "802.16 Chapter 11.4.9"
23     ::= { wmanIfCmnClassifierRuleEntry 8 }
24
25     wmanIfCmnClassifierRuleInetAddressType OBJECT-TYPE
26     SYNTAX      InetAddressType
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "The type of the internet address for
31         wmanIfCmnClassifierRuleInetSourceAddr,
32         wmanIfCmnClassifierRuleInetSourceMask,
33         wmanIfCmnClassifierRuleInetDestAddr, and
34         wmanIfCmnClassifierRuleInetDestMask.
35         If the referenced parameter is not present in a classifier,
36         this object reports the value of ipv4(1)."
37     REFERENCE   ""
38     ::= { wmanIfCmnClassifierRuleEntry 9 }
39
40     wmanIfCmnClassifierRuleInetSourceAddr OBJECT-TYPE
41     SYNTAX      InetAddress
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "This object specifies the value of the IP Source Address
46         required for packets to match this rule. An IP packet
47         matches the rule when the packet ip source address bitwise
48         ANDed with the wmanIfCmnClassifierRuleInetSourceMask value
49         equals the wmanIfCmnClassifierRuleInetSourceAddr value.
50         If the referenced parameter is not present n a classifier,
51         this object reports the value of 0.0.0.0."
52     REFERENCE   "802.16 Chapter 11.4.9"
53     ::= { wmanIfCmnClassifierRuleEntry 10 }
54

```

```

1  wmanIfCmnClassifierRuleInetSourceMask OBJECT-TYPE
2      SYNTAX      InetAddress
3      MAX-ACCESS  read-write
4      STATUS      current
5      DESCRIPTION
6          "This object specifies which bits of a packet's IP Source
7          Address that are compared to match this rule. An IP packet
8          matches the rule when the packet source address bitwise
9          ANDed with the
10         wmanIfCmnClassifierRuleInetSourceMask value equals the
11         wmanIfCmnClassifierRuleInetSourceAddr value.
12         If the referenced parameter is not present in a classifier,
13         this object reports the value of 0.0.0.0."
14     REFERENCE      "802.16 Chapter 11.4.9"
15     ::= { wmanIfCmnClassifierRuleEntry 11 }
16
17  wmanIfCmnClassifierRuleInetDestAddr OBJECT-TYPE
18      SYNTAX      InetAddress
19      MAX-ACCESS  read-write
20      STATUS      current
21      DESCRIPTION
22          "This object specifies the value of the IP Destination
23          Address required for packets to match this rule. An IP
24          packet matches the rule when the packet IP destination
25          address bitwise ANDed with the
26          wmanIfCmnClassifierRuleInetDestMask value equals the
27          wmanIfCmnClassifierRuleInetDestAddr value.
28          If the referenced parameter is not present in a
29          classifier, this object reports the value of 0.0.0.0."
30     REFERENCE      "802.16 Chapter 11.4.9"
31     ::= { wmanIfCmnClassifierRuleEntry 12 }
32
33  wmanIfCmnClassifierRuleInetDestMask OBJECT-TYPE
34      SYNTAX      InetAddress
35      MAX-ACCESS  read-write
36      STATUS      current
37      DESCRIPTION
38          "This object specifies which bits of a packet's IP
39          Destination Address that are compared to match this rule.
40          An IP packet matches the rule when the packet destination
41          address bitwise ANDed with the
42          wmanIfCmnClassifierRuleInetDestMask value equals the
43          wmanIfCmnClassifierRuleInetDestAddr value.
44          If the referenced parameter is not present in a classifier
45          , this object reports the value of 0.0.0.0."
46     REFERENCE      "802.16 Chapter 11.4.9"
47     ::= { wmanIfCmnClassifierRuleEntry 13 }
48
49  wmanIfCmnClassifierRuleSourcePortStart OBJECT-TYPE
50      SYNTAX      Integer32 (0..65535)
51      MAX-ACCESS  read-write
52      STATUS      current
53      DESCRIPTION
54          "This object specifies the low end inclusive range of

```

```

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

```

```

1         parameter is not present in a classifier, this object
2         reports the value of '000000000000'H."
3     REFERENCE      "802.16 Chapter 11.4.9"
4     ::= { wmanIfCmnClassifierRuleEntry 18 }
5
6     wmanIfCmnClassifierRuleDestMacMask OBJECT-TYPE
7         SYNTAX      MacAddress
8         MAX-ACCESS  read-write
9         STATUS      current
10        DESCRIPTION
11            "An Ethernet packet matches an entry when its destination
12            MAC address bitwise ANDed with
13            wmanIfCmnClassifierRuleDestMacMask equals the value of
14            wmanIfCmnClassifierRuleDestMacAddr. If the referenced
15            parameter is not present in a classifier, this object
16            reports the value of '000000000000'H."
17        REFERENCE      "802.16 Chapter 11.4.9"
18        ::= { wmanIfCmnClassifierRuleEntry 19 }
19
20        wmanIfCmnClassifierRuleSourceMacAddr OBJECT-TYPE
21            SYNTAX      MacAddress
22            MAX-ACCESS  read-write
23            STATUS      current
24            DESCRIPTION
25                "An Ethernet packet matches this entry when its source
26                MAC address bitwise ANDed with
27                wmanIfCmnClassifierRuleSourceMacMask equals the value
28                of wmanIfCmnClassifierRuleSourceMacAddr. If the
29                referenced parameter is not present in a classifier,
30                this object reports the value of 'FFFFFFFFFFFF'H."
31            REFERENCE      "802.16 Chapter 11.4.9"
32            ::= { wmanIfCmnClassifierRuleEntry 20 }
33
34        wmanIfCmnClassifierRuleSourceMacMask OBJECT-TYPE
35            SYNTAX      MacAddress
36            MAX-ACCESS  read-write
37            STATUS      current
38            DESCRIPTION
39                "An Ethernet packet matches an entry when its destination
40                MAC address bitwise ANDed with
41                wmanIfCmnClassifierRuleSourceMacMask equals the value of
42                wmanIfCmnClassifierRuleSourceMacAddr. If the referenced
43                parameter is not present in a classifier, this object
44                reports the value of '000000000000'H."
45            REFERENCE      "802.16 Chapter 11.4.9"
46            ::= { wmanIfCmnClassifierRuleEntry 21 }
47
48        wmanIfCmnClassifierRuleEnetProtocolType OBJECT-TYPE
49            SYNTAX      INTEGER {none(0),
50                        ethertype(1),
51                        dsap(2)}
52            MAX-ACCESS  read-write
53            STATUS      current
54            DESCRIPTION

```

```

1       "This object indicates the format of the layer 3 protocol
2       id in the Ethernet packet. A value of none(0) means that
3       the rule does not use the layer 3 protocol type as a
4       matching criteria. A value of ethertype(1) means that the
5       rule applies only to frames which contains an EtherType
6       value. Ethertype values are contained in packets using
7       the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
8       Sub-Network Access Protocol (SNAP) encapsulation formats.
9       A value of dsap(2) means that the rule applies only to
10      frames using the IEEE802.3 encapsulation format with a
11      Destination Service Access Point (DSAP) other than 0xAA
12      (which is reserved for SNAP). If the Ethernet frame
13      contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
14      this object applies to the embedded EtherType field within
15      the 802.1P/Q header. If the referenced parameter is not
16      present in a classifier, this object reports the value of
17      0."
18      REFERENCE   "802.16 Chapter 11.4.9"
19      ::= { wmanIfCmnClassifierRuleEntry 22 }
20
21      wmanIfCmnClassifierRuleEnetProtocol OBJECT-TYPE
22          SYNTAX      Integer32 (0..65535)
23          MAX-ACCESS  read-write
24          STATUS      current
25          DESCRIPTION
26              "If wmanIfCmnClassifierRuleEnetProtocolType is none(0),
27              this object is ignored when considering whether a packet
28              matches the current rule.
29              If wmanIfCmnClassifierRuleEnetProtocolType is ethertype(1),
30              this object gives the 16-bit value of the EtherType that
31              the packet must match in order to match the rule.
32              If wmanIfCmnClassifierRuleEnetProtocolType is dsap(2), the
33              lower 8 bits of this object's value must match the DSAP
34              byte of the packet in order to match the rule.
35              If wmanIfCmnClassifierRuleEnetProtocolType is mac(3), the
36              lower 8 bits of this object value represent a lower bound
37              (inclusive) of MAC management message type codes matched,
38              and the upper 8 bits of this object value represent the
39              upper bound (inclusive) of matched MAC message type codes.
40              Certain message type codes are excluded from matching, as
41              specified in the reference.
42              If the Ethernet frame contains an 802.1P/Q Tag header
43              (i.e. EtherType 0x8100), this object applies to the
44              embedded EtherType field within the 802.1P/Q header.
45              If the referenced parameter is not present in the
46              classifier, the value of this object is reported as 0."
47      REFERENCE   "802.16 Chapter 11.4.9"
48      ::= { wmanIfCmnClassifierRuleEntry 23 }
49
50      wmanIfCmnClassifierRuleUserPriLow OBJECT-TYPE
51          SYNTAX      Integer32 (0..7)
52          MAX-ACCESS  read-write
53          STATUS      current
54          DESCRIPTION

```

```

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

```



```

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

```

```

1      ::= { wmanIfCmnCpsServiceFlowTable 1 }
2
3  wmanIfCmnCpsServiceFlowEntry ::= SEQUENCE {
4      wmanIfCmnCpsSfId                Unsigned32,
5      wmanIfCmnCpsSfCid                INTEGER,
6      wmanIfCmnCpsSfDirection          INTEGER,
7      wmanIfCmnCpsServiceClassIndex   INTEGER,
8      wmanIfCmnCpsSfState              INTEGER,
9      wmanIfCmnCpsServiceClassName     DisplayString,
10     wmanIfCmnCpsTrafficPriority       INTEGER,
11     wmanIfCmnCpsMaxSustainedRate     INTEGER,
12     wmanIfCmnCpsMaxTrafficBurst      INTEGER,
13     wmanIfCmnCpsMinReservedRate     INTEGER,
14     wmanIfCmnCpsToleratedJitter     INTEGER,
15     wmanIfCmnCpsMaxLatency           INTEGER,
16     wmanIfCmnCpsScSchedulingType     wmanSfsSchedulingType,
17     wmanIfCmnCpsScArqEnable          TruthValue,
18     wmanIfCmnCpsScArqWindowSize      INTEGER,
19     wmanIfCmnCpsScArqFragmentLifetime INTEGER,
20     wmanIfCmnCpsScArqSyncLossTimeout INTEGER,
21     wmanIfCmnCpsScArqDeliverInOrder  TruthValue,
22     wmanIfCmnCpsScArqRxPurgeTimeout  INTEGER,
23     wmanIfCmnCpsScFragmentLen        INTEGER,
24     wmanIfCmnCpsSCMinRsvdTolerableRate INTEGER
25     }
26
27  wmanIfCmnCpsSfId OBJECT-TYPE
28      SYNTAX      Unsigned32 ( 1 .. 4294967295)
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "A 32 bit quantity that uniquely identifies a service flow
33           to both the subscriber station and base station (BS)."
```

```

34      ::= { wmanIfCmnCpsServiceFlowEntry 1 }
35
36  wmanIfCmnCpsSfCid OBJECT-TYPE
37      SYNTAX      INTEGER
38      MAX-ACCESS  read-only
39      STATUS      current
40      DESCRIPTION
41          "A 16 bit channel identifier to identify the connection
42           being created by DSA."
43      ::= { wmanIfCmnCpsServiceFlowEntry 2 }
44
45  wmanIfCmnCpsSfDirection OBJECT-TYPE
46      SYNTAX      INTEGER {downstream(1),
47                      upstream(2)}
48      MAX-ACCESS  read-only
49      STATUS      current
50      DESCRIPTION
51          "An attribute indicating the service flow is downstream or
52           upstream."
53      ::= { wmanIfCmnCpsServiceFlowEntry 3 }
54

```

```

1  wmanIfCmnCpsServiceClassIndex OBJECT-TYPE
2      SYNTAX      INTEGER
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "The wmanIfCmnCpsServiceClassIndex associates this
7          service flow with QoS parameters.
8          If no associated entry in wmanIfCmnCpsServiceFlowTable
9          exists, this object returns a value of zero."
10     ::= { wmanIfCmnCpsServiceFlowEntry 4 }
11
12  wmanIfCmnCpsSfState OBJECT-TYPE
13      SYNTAX      INTEGER {provisionedState(1),
14                          admittedState(2),
15                          activeState(3)}
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19          "wmanIfCmnCpsSfState indicates the service flow state:
20          Provisioned, AdmittedState(2), and Active service flow
21          state."
22      REFERENCE
23          "Section 6.4.13.6, in IEEE 802.16REVd/D3-2004"
24     ::= { wmanIfCmnCpsServiceFlowEntry 5 }
25
26  wmanIfCmnCpsServiceClassName OBJECT-TYPE
27      SYNTAX      DisplayString
28      MAX-ACCESS  read-only
29      STATUS      current
30      DESCRIPTION
31          "Refers to the Service Class Name"
32      REFERENCE   "802.16 Chapter 11.4.8"
33     ::= { wmanIfCmnCpsServiceFlowEntry 6 }
34
35  wmanIfCmnCpsTrafficPriority OBJECT-TYPE
36      SYNTAX      INTEGER
37      MAX-ACCESS  read-only
38      STATUS      current
39      DESCRIPTION
40          "The value of this parameter specifies the priority
41          assigned to a service flow. For uplink service flows,
42          the BS should use this parameter when determining
43          precedence in request service and grant generation,
44          and the SS shall preferentially select contention
45          Request opportunities for Priority Request CIDs
46          based on this priority"
47      REFERENCE
48          "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
49     ::= { wmanIfCmnCpsServiceFlowEntry 7 }
50
51  wmanIfCmnCpsMaxSustainedRate OBJECT-TYPE
52      SYNTAX      INTEGER
53      MAX-ACCESS  read-only
54      STATUS      current

```

```

1      DESCRIPTION
2          "This parameter defines the peak information rate
3          of the service. The rate is expressed in bits per
4          second and pertains to the SDUs at the input to
5          the system."
6      REFERENCE
7          "Section 11.13.8 in IEEE 802.16REVd/D3-2004"
8      ::= { wmanIfCmnCpsServiceFlowEntry 8 }
9
10     wmanIfCmnCpsMaxTrafficBurst OBJECT-TYPE
11         SYNTAX      INTEGER
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "This parameter defines the maximum burst size that
16             must be accommodated for the service."
17         REFERENCE
18             "Section 11.13.9 in IEEE 802.16REVd/D3-2004"
19         ::= { wmanIfCmnCpsServiceFlowEntry 9 }
20
21     wmanIfCmnCpsMinReservedRate OBJECT-TYPE
22         SYNTAX      INTEGER
23         MAX-ACCESS  read-only
24         STATUS      current
25         DESCRIPTION
26             "This parameter specifies the minimum rate reserved
27             for this service flow."
28         REFERENCE
29             "Section 11.13.10 in IEEE 802.16REVd/D3-2004"
30         ::= { wmanIfCmnCpsServiceFlowEntry 10 }
31
32     wmanIfCmnCpsToleratedJitter OBJECT-TYPE
33         SYNTAX      INTEGER
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "This parameter defines the Maximum delay
38             variation (jitter) for the connection."
39         REFERENCE
40             "Section 11.13.15 in IEEE 802.16REVd/D3-2004"
41         ::= { wmanIfCmnCpsServiceFlowEntry 11 }
42
43     wmanIfCmnCpsMaxLatency OBJECT-TYPE
44         SYNTAX      INTEGER
45         MAX-ACCESS  read-only
46         STATUS      current
47         DESCRIPTION
48             "The value of this parameter specifies the maximum
49             latency between the reception of a packet by the BS
50             or SS on its network interface and the forwarding
51             of the packet to its RF Interface."
52         REFERENCE
53             "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
54         ::= { wmanIfCmnCpsServiceFlowEntry 12 }

```

```

1
2 wmanIfCmnCpsScsSchedulingType OBJECT-TYPE
3     SYNTAX      wmanSfsSchedulingType
4     MAX-ACCESS  read-only
5     STATUS      current
6     DESCRIPTION
7         "Specifies the upstream scheduling service used for
8         upstream service flow. If the referenced parameter
9         is not present in the corresponding 802.16 QoS
10        Parameter Set of an upstream service flow, the
11        default value of this object is bestEffort(2)."

```

```

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

```

```

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

```

```

1         of 2 (in units of PS)."
```

::= { wmanIfCmnBsSsConfigurationEntry 3 }

```

3
4 wmanIfCmnDSxReqRetries OBJECT-TYPE
5     SYNTAX      INTEGER
6     MAX-ACCESS  read-write
7     STATUS      current
8     DESCRIPTION
9         "Number of Timeout Retries on DSA/DSC/DSD Requests."
10    ::= { wmanIfCmnBsSsConfigurationEntry 4 }
11
12 wmanIfCmnDSxRespRetries OBJECT-TYPE
13     SYNTAX      INTEGER
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "Number of Timeout Retries on DSA/DSC/DSD Responses."
18    ::= { wmanIfCmnBsSsConfigurationEntry 5 }
19
20 wmanIfCmnT7Timeout OBJECT-TYPE
21     SYNTAX      INTEGER(0 .. 1000)
22     UNITS       "milliseconds"
23     MAX-ACCESS  read-write
24     STATUS      current
25     DESCRIPTION
26         "Wait for DSA/DSC/DSD Response Timeout in ms."
27    ::= { wmanIfCmnBsSsConfigurationEntry 6 }
28
29 wmanIfCmnT8Timeout OBJECT-TYPE
30     SYNTAX      INTEGER(0 .. 300)
31     UNITS       "milliseconds"
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "Wait for DSA/DSC/DSD Acknowledge Timeout in ms."
36    ::= { wmanIfCmnBsSsConfigurationEntry 7 }
37
38 wmanIfCmnT10Timeout OBJECT-TYPE
39     SYNTAX      INTEGER(0 .. 3000)
40     UNITS       "milliseconds"
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "Wait for Transaction End timeout in ms."
45    ::= { wmanIfCmnBsSsConfigurationEntry 8 }
46
47 wmanIfCmnT22Timeout OBJECT-TYPE
48     SYNTAX      INTEGER(0 .. 500)
49     UNITS       "milliseconds"
50     MAX-ACCESS  read-write
51     STATUS      current
52     DESCRIPTION
53         "Wait for ARQ Reset in ms."
54    ::= { wmanIfCmnBsSsConfigurationEntry 9 }
```



```

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

```

```

1
2  wmanIfCmnSsSchMeasurementEntry ::= SEQUENCE {
3      wmanIfCmnSsIdIndex                Unsigned32,
4      wmanIfCmnHistogramIndex           Unsigned32,
5      wmanIfCmnChannelNumber            INTEGER,
6      wmanIfCmnStartFrame                INTEGER,
7      wmanIfCmnDuration                  INTEGER,
8      wmanIfCmnBasicReport                BITS,
9      wmanIfCmnMeanCinrReport            INTEGER,
10     wmanIfCmnStdDeviationCinrReport     INTEGER,
11     wmanIfCmnMeanRssiReport             INTEGER,
12     wmanIfCmnStdDeviationRssiReport     INTEGER
13     }
14
15  wmanIfCmnSsIdIndex OBJECT-TYPE
16      SYNTAX      Unsigned32 (1 .. 4294967295)
17      MAX-ACCESS  read-only
18      STATUS      current
19      DESCRIPTION
20          "wmanIfCmnSsIdIndex identifies the SS providing the
21          channel measurement."
22      REFERENCE
23          "Section 6.4.2.3.5 in IEEE 802.16REvD/D3-2004"
24      ::= { wmanIfCmnSsSchMeasurementEntry 1 }
25
26  wmanIfCmnHistogramIndex OBJECT-TYPE
27      SYNTAX      Unsigned32 (1 .. 4294967295)
28      MAX-ACCESS  read-only
29      STATUS      current
30      DESCRIPTION
31          "wmanIfBsHistogramIndex identifies the histogram samples
32          in the table for each subscriber station."
33      ::= { wmanIfCmnSsSchMeasurementEntry 2 }
34
35  wmanIfCmnChannelNumber OBJECT-TYPE
36      SYNTAX      INTEGER
37      MAX-ACCESS  read-only
38      STATUS      current
39      DESCRIPTION
40          "Physical channel number to be reported on."
41      REFERENCE
42          "Section 8.5.1 in IEEE 802.16REvD/D3-2004"
43      ::= { wmanIfCmnSsSchMeasurementEntry 3 }
44
45  wmanIfCmnStartFrame OBJECT-TYPE
46      SYNTAX      INTEGER
47      MAX-ACCESS  read-only
48      STATUS      current
49      DESCRIPTION
50          "Frame number in which measurement for this channel
51          started."
52      REFERENCE
53          "Section 11.12 in IEEE 802.16REvD/D3-2004"
54      ::= { wmanIfCmnSsSchMeasurementEntry 4 }

```

```

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

```

```

1      STATUS      current
2      DESCRIPTION
3          "Mean RSSI report."
4      REFERENCE
5          "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
6          802.16REVd/D3-2004"
7      ::= { wmanIfCmnSsSchMeasurementEntry 9 }
8
9      wmanIfCmnStdDeviationRssiReport OBJECT-TYPE
10     SYNTAX      INTEGER
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "Standard deviation RSSI report."
15     REFERENCE
16         "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
17         802.16REVd/D3-2004"
18     ::= { wmanIfCmnSsSchMeasurementEntry 10 }
19
20     -- Common PKM group
21     -- wmanIfCmnPkmObjects contain the Privacy Sublayer objects that are
22     -- common to both Base Station and Subscriber Station
23     wmanIfCmnPkmObjects OBJECT IDENTIFIER ::= { wmanIfCommonObjects 3 }
24
25     --
26     -- Table wmanIfCmnCryptoSuiteTable
27     --
28     wmanIfCmnCryptoSuiteTable OBJECT-TYPE
29     SYNTAX      SEQUENCE OF wmanIfCmnCryptoSuiteEntry
30     MAX-ACCESS  not-accessible
31     STATUS      current
32     DESCRIPTION
33         "This table describes the PKM cryptographic suite
34         capabilities for each SS or BS wireless interface."
35     ::= { wmanIfCmnPkmObjects 1 }
36
37     wmanIfCmnCryptoSuiteEntry OBJECT-TYPE
38     SYNTAX      wmanIfCmnCryptoSuiteEntry
39     MAX-ACCESS  not-accessible
40     STATUS      current
41     DESCRIPTION
42         "Each entry contains the cryptographic suite pair that SS
43         or BS supports."
44     INDEX      { ifIndex, wmanIfSsCryptoSuiteIndex }
45     ::= { wmanIfCmnCryptoSuiteTable 1 }
46
47     wmanIfCmnCryptoSuiteEntry ::= SEQUENCE {
48         wmanIfSsCryptoSuiteIndex      Integer32,
49         wmanIfCmnCryptoSuiteDataEncryptAlg  INTEGER,
50         wmanIfCmnCryptoSuiteDataAuthentAlg  INTEGER,
51         wmanIfCmnCryptoSuiteTEKEncryptAlg   INTEGER
52     }
53
54     wmanIfSsCryptoSuiteIndex OBJECT-TYPE

```

```

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

```

```

1      DESCRIPTION
2          "This table contains UCD channel attributes, defining the
3          transmission characteristics of uplink channels"
4      REFERENCE
5          "Section 11.3.1, table 276 and 279, in IEEE
6          802.16REVd/D3-2004"
7      ::= { wmanIfCmnOfdmPhy 1 }
8
9      wmanIfCmnOfdmUplinkChannelEntry OBJECT-TYPE
10         SYNTAX      wmanIfCmnOfdmUplinkChannelEntry
11         MAX-ACCESS  not-accessible
12         STATUS      current
13         DESCRIPTION
14             "This table provides one row for each uplink channel of
15             multi-sector BS, and is indexed by BS ifIndex. An entry
16             in this table exists for each ifEntry of BS with an
17             ifType of propBWA2Mp.
18             The objects in each entry will be implemented as
19             read-create in BS and read-only in SS."
20         INDEX { ifIndex }
21         ::= { wmanIfCmnOfdmUplinkChannelTable 1 }
22
23     wmanIfCmnOfdmUplinkChannelEntry ::= SEQUENCE {
24         wmanIfCmnOfdmCtBasedResvTimeout    INTEGER,
25         wmanIfCmnOfdmBwReqOppSize         INTEGER,
26         wmanIfCmnOfdmRangReqOppSize       INTEGER,
27         wmanIfCmnOfdmUplinkCenterFreq     INTEGER,
28         wmanIfCmnOfdmSubChReqRegionFull   INTEGER,
29         wmanIfCmnOfdmSubChFocusCtCode     INTEGER,
30         wmanIfCmnOfdmUplinkChannelRowStatus RowStatus
31     }
32
33     wmanIfCmnOfdmCtBasedResvTimeout OBJECT-TYPE
34         SYNTAX      INTEGER (1..255)
35         MAX-ACCESS  read-write
36         STATUS      current
37         DESCRIPTION
38             "The number of UL-MAPS to receive before contention-based
39             reservation is attempted again for the same connection."
40         REFERENCE
41             "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
42         ::= { wmanIfCmnOfdmUplinkChannelEntry 1 }
43
44     wmanIfCmnOfdmBwReqOppSize OBJECT-TYPE
45         SYNTAX      INTEGER (1..65535)
46         MAX-ACCESS  read-write
47         STATUS      current
48         DESCRIPTION
49             " Size (in units of PS) of PHY payload that SS may use to
50             format and transmit a bandwidth request message in a
51             contention request opportunity. The value includes all
52             PHY overhead as well as allowance for the MAC data the
53             message may hold."
54         REFERENCE

```

```

1         "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
2         ::= { wmanIfCmnOfdmUplinkChannelEntry 2 }
3
4     wmanIfCmnOfdmRangReqOppSize OBJECT-TYPE
5         SYNTAX      INTEGER (1..65535)
6         UNITS       "PS"
7         MAX-ACCESS  read-write
8         STATUS      current
9         DESCRIPTION
10            " Size (in units of PS) of PHY payload that SS may use to
11            format and transmit a RNG-REQ message in a contention
12            request opportunity. The value includes all PHY overhead
13            as well as allowance for the MAC data the message may
14            hold and the maximum SS/BS roundtrip propagation delay."
15         REFERENCE
16            "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
17         ::= { wmanIfCmnOfdmUplinkChannelEntry 3 }
18
19     wmanIfCmnOfdmUplinkCenterFreq OBJECT-TYPE
20         SYNTAX      INTEGER
21         UNITS       "KHZ"
22         MAX-ACCESS  read-write
23         STATUS      current
24         DESCRIPTION
25            " Uplink center frequency (KHz)"
26         REFERENCE
27            "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
28         ::= { wmanIfCmnOfdmUplinkChannelEntry 4 }
29
30     wmanIfCmnOfdmSubChReqRegionFull OBJECT-TYPE
31         SYNTAX      INTEGER {oneSubchannel(0),
32                             twoSubchannels(1),
33                             fourSubchannels(2),
34                             eightSubchannels(3),
35                             sixteenSubchannels(4)}
36         MAX-ACCESS  read-write
37         STATUS      current
38         DESCRIPTION
39            "Bits 0 - 2 Number of subchannels used by each transmit
40            opportunity when REQ Region-Full is allocated in
41            subchannelization region, per the following enumeration:
42            0: 1 Subchannel.
43            1: 2 Subchannels.
44            2: 4 Subchannels.
45            3: 8 Subchannels.
46            4: 16 Subchannels.
47            5-7: Shall not be used.
48            Bits 3 - 7: Number of OFDM symbols used by each transmit
49            opportunity when REQ Region-Full is allocated in
50            subchannelization region.
51         REFERENCE
52            Section 11.3.1, table 279, in IEEE 802.16REVd/D3-2004"
53         ::= { wmanIfCmnOfdmUplinkChannelEntry 5 }
54

```

```

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

```



```

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

```

```

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

```

```

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

```

```

1           This table is double indexed. The primary index is an
2           ifIndex with an ifType of propBWApm2Mp. The secondary index
3           is wmanIfCmnOfdmOfdmUcdBurstProfIndex.
4           The objects in each entry will be implemented as
5           read-create in BS and read-only in SS."
6     INDEX { ifIndex, wmanIfCmnOfdmOfdmUcdBurstProfIndex }
7     ::= { wmanIfCmnOfdmUcdBurstProfileTable 1 }
8
9     wmanIfCmnOfdmUcdBurstProfileEntry ::= SEQUENCE {
10        wmanIfCmnOfdmOfdmUcdBurstProfIndex      INTEGER,
11        wmanIfCmnOfdmUplinkFrequency           INTEGER,
12        wmanIfCmnOfdmUcdFecCodeType           INTEGER,
13        wmanIfCmnOfdmFocusCtPowerBoost        INTEGER,
14        wmanIfCmnOfdmUcdBurstProfileRowStatus  RowStatus
15    }
16
17     wmanIfCmnOfdmOfdmUcdBurstProfIndex OBJECT-TYPE
18     SYNTAX      INTEGER (1 .. 100)
19     MAX-ACCESS  not-accessible
20     STATUS      current
21     DESCRIPTION
22         "ifIndex and wmanIfCmnOfdmOfdmUcdBurstProfIndex uniquely
23         identify an entry in the wmanIfCmnOfdmUcdBurstProfileTable."
24     ::= { wmanIfCmnOfdmUcdBurstProfileEntry 1 }
25
26     wmanIfCmnOfdmUplinkFrequency OBJECT-TYPE
27     SYNTAX      INTEGER
28     UNITS       "KHZ"
29     MAX-ACCESS  read-write
30     STATUS      current
31     DESCRIPTION
32         "Uplink Frequency (kHz)."

```

```

1                                     sixtyFourQamCtc3-4(18)}
2     MAX-ACCESS    read-write
3     STATUS       current
4     DESCRIPTION
5         " 0= QPSK (RS+CC/CC) 1/2
6           1= QPSK (RS+CC/CC) 3/4
7           2= 16-QAM (RS+CC/CC) 1/2
8           3= 16-QAM (RS+CC/CC) 3/4
9           4= 64-QAM (RS+CC/CC) 2/3
10          5= 64-QAM (RS+CC/CC) 3/4
11          6= QPSK (BTC) 1/2
12          7= QPSK (BTC) 3/4
13          8= 16-QAM (BTC) 3/5
14          9= 16-QAM (BTC) 4/5
15          10 = 64-QAM (BTC) 2/3
16          11 = 64-QAM (BTC) 5/6
17          12 = QPSK (CTC) 1/2
18          13 = QPSK (CTC) 2/3
19          14 = QPSK (CTC) 3/4
20          15 = 16-QAM (CTC) 1/2
21          16 = 16-QAM (CTC) 3/4
22          17 = 64-QAM (CTC) 2/3
23          18 = 64-QAM (CTC) 3/4
24          19 - 255 Reserved."
25     REFERENCE
26         "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
27     ::= { wmanIfCmnOfdmUcdBurstProfileEntry 3 }
28
29     wmanIfCmnOfdmFocusCtPowerBoost OBJECT-TYPE
30         SYNTAX      INTEGER
31         MAX-ACCESS  read-write
32         STATUS      current
33         DESCRIPTION
34             "The power boost in dB of focused contention carriers, as
35             described in 8.3.6.3.3."
36         REFERENCE
37             "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
38         ::= { wmanIfCmnOfdmUcdBurstProfileEntry 4 }
39
40     wmanIfCmnOfdmUcdBurstProfileRowStatus 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         ::= { wmanIfCmnOfdmUcdBurstProfileEntry 5 }
53
54     wmanIfCmnOfdmDcdBurstProfileTable OBJECT-TYPE

```

```

1      SYNTAX      SEQUENCE OF WmanIfOfdmDcdBurstProfileEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table provides one row for each DCD burst profile.
6          This table is double indexed. The primary index is an
7          ifIndex with an ifType of propBWA2Mp. The secondary
8          index is wmanIfCmnOfdmOfdmDcdBurstProfIndex"
9      ::= { wmanIfCmnOfdmPhy 4 }
10
11
12     wmanIfCmnOfdmDcdBurstProfileEntry OBJECT-TYPE
13         SYNTAX      WmanIfOfdmDcdBurstProfileEntry
14         MAX-ACCESS  not-accessible
15         STATUS      current
16         DESCRIPTION
17             "This table provides one row for each DCD burst profile.
18             This table is double indexed. The primary index is an
19             ifIndex with an ifType of propBWA2Mp. The secondary index
20             is wmanIfCmnOfdmDcdBurstProfIndex.
21             The objects in each entry will be implemented as
22             read-create in BS and read-only in SS."
23         INDEX { ifIndex, wmanIfCmnOfdmDcdBurstProfIndex }
24         ::= { wmanIfCmnOfdmDcdBurstProfileTable 1 }
25
26     WmanIfOfdmDcdBurstProfileEntry ::= SEQUENCE {
27         wmanIfCmnOfdmDcdBurstProfIndex      INTEGER,
28         wmanIfCmnOfdmDownlinkFrequency     INTEGER,
29         wmanIfCmnOfdmDcdFecCodeType       INTEGER,
30         wmanIfCmnOfdmDiucMandatoryExitThresh  INTEGER,
31         wmanIfCmnOfdmDiucMinEntryThresh    INTEGER,
32         wmanIfCmnOfdmDcdBurstProfileRowStatus  RowStatus
33     }
34
35     wmanIfCmnOfdmDcdBurstProfIndex OBJECT-TYPE
36         SYNTAX      INTEGER (1 .. 100)
37         MAX-ACCESS  not-accessible
38         STATUS      current
39         DESCRIPTION
40             "ifIndex and wmanIfCmnOfdmDcdBurstProfIndex uniquely
41             identify an entry in the wmanIfCmnOfdmDcdBurstProfileTable."
42         ::= { wmanIfCmnOfdmDcdBurstProfileEntry 1 }
43
44     wmanIfCmnOfdmDownlinkFrequency OBJECT-TYPE
45         SYNTAX      INTEGER
46         UNITS       "KHZ"
47         MAX-ACCESS  read-write
48         STATUS      current
49         DESCRIPTION
50             "Downlink Frequency (kHz)."
51         REFERENCE
52             "Section 11.4.1, table 287, in IEEE 802.16REVd/D3-2004"
53         ::= { wmanIfCmnOfdmDcdBurstProfileEntry 2 }
54

```

```

1  wmanIfCmnOfdmDcdFecCodeType OBJECT-TYPE
2      SYNTAX      INTEGER {qpskRsCc1-2(0),
3                    qpskRsCc3-4(1),
4                    sixteenQamRsCc1-2(2),
5                    sixteenQamRsCc3-4(3),
6                    sixtyFourQamRsCc2-3(4),
7                    sixtyFourQamRsCc3-4(5),
8                    qpskBtc1-2(6),
9                    qpskBtc3-4(7),
10                   sixteenQamBtc3-4(8),
11                   sixteenQamBtc4-5(9),
12                   sixtyFourQamBtc2-3or5-8(10),
13                   sixtyFourQamBtc5-6or4-5(11),
14                   qpskCtc1-2(12),
15                   qpskCtc2-3(13),
16                   qpskCtc3-4(14),
17                   sixteenQamCtc1-2(16),
18                   sixteenQamCtc3-4(17),
19                   sixtyFourQamCtc3-4(18)}
20  MAX-ACCESS    read-write
21  STATUS        current
22  DESCRIPTION
23      " 0= QPSK (RS+CC) 1/2
24      1= QPSK (RS+CC) 3/4
25      2= 16-QAM (RS+CC) 1/2
26      3= 16-QAM (RS+CC) 3/4
27      4= 64-QAM (RS+CC) 2/3
28      5= 64-QAM (RS+CC) 3/4
29      6= QPSK (BTC) 1/2
30      7= QPSK (BTC) 3/4
31      8= 16-QAM (BTC) 3/5
32      9= 16-QAM (BTC) 4/5
33      10 = 64-QAM (BTC) 2/3 or 5/8
34      11 = 64-QAM (BTC) 5/6 or 4/5
35      12 = QPSK (CTC) 1/2
36      13 = QPSK (CTC) 2/3
37      14 = QPSK (CTC) 3/4
38      15 = 16-QAM (CTC) 1/2
39      16 = 16-QAM (CTC) 3/4
40      17 = 64-QAM (CTC) 2/3
41      18 = 64-QAM (CTC) 3/4
42      19 - 255 Reserved."
43  REFERENCE
44      "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
45      ::= { wmanIfCmnOfdmDcdBurstProfileEntry 3 }
46
47  wmanIfCmnOfdmDiucMandatoryExitThresh OBJECT-TYPE
48      SYNTAX      INTEGER
49      MAX-ACCESS  read-write
50      STATUS      current
51      DESCRIPTION
52          "DIUC mandatory exit threshold: 0 - 63.75 dB CINR at or
53          below where this DIUC can no longer be used and where this
54          change to a more robust DIUC is required, in 0.25 dB units."

```

```
1      REFERENCE
2      "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
3      ::= { wmanIfCmnOfdmDcdBurstProfileEntry 4 }
4
5      wmanIfCmnOfdmDiucMinEntryThresh OBJECT-TYPE
6          SYNTAX      INTEGER
7          MAX-ACCESS  read-write
8          STATUS      current
9          DESCRIPTION
10             "DIUC minimum entry threshold: 0 - 63.75 dB The minimum CINR
11             required to start using this DIUC when changing from a more
12             robust DIUC is required, in 0.25 dB units."
13          REFERENCE
14             "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
15             ::= { wmanIfCmnOfdmDcdBurstProfileEntry 5 }
16
17      wmanIfCmnOfdmDcdBurstProfileRowStatus OBJECT-TYPE
18          SYNTAX      RowStatus
19          MAX-ACCESS  read-create
20          STATUS      current
21          DESCRIPTION
22             "This object is used to create a new row or modify or
23             delete an existing row in this table.
24
25             If the implementator of this MIB has choosen not
26             to implement 'dynamic assignment' of profiles, this
27             object is not useful and should return noSuchName
28             upon SNMP request."
29             ::= { wmanIfCmnOfdmDcdBurstProfileEntry 6 }
30
31
32      END
33
34
```