

Project	IEEE 802.16 Broadband Wireless Access Working Group < http://ieee802.org/16 >	
Title	wmanIfMib for WirelessMAN and WirelessHUMAN BS and SS	
Date Submitted	2004-08-26	
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 Krzysztof Dudzinski Airspan	[mailto:seini@Airspan.com] [mailto:KDudzinski@Airspan.com]
	Radu Selea Bogdan Moldoveanu Redline Communications	[mailto:radu@redlinecommunications.com]
	Patrice Desmoulin France Telecom	[mailto:patrice.desmoulin@francetelecom.com]
Re:		
Abstract	This contribution defines wmanIfMib (Wireless MAN Interface MIB) in response to the Network Management Task Group calls for contributions 802.16f PAR – "Amendment to IEEE Standard for Local and Metropolitan Area Networks- Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Management Information Base".	
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/ipr/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.....	11
	4.1.4 wmanIfBsPkm	11
	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.....	12
	4.2.3.1 wmanIfSsPkmAuthTable.....	12
	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.....	13
4.3.2.3	wmanIfCmnSsChMeasurementTable.....	13
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.....	14
5.	<i>ASN.1 Definition of 802.16 MIB</i>	15

1. Introduction

This contribution defines wmanIfMib (Wireless MAN Interface MIB) in response to the Network Management Task Group calls for contributions 802.16f PAR – "Amendment to IEEE Standard for Local and Metropolitan Area Networks- Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Management Information Base".

It is proposed this contribution be adopted as the base line document for 802.16f – "Amendment to IEEE Standard for Local and Metropolitan Area Networks- Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Management Information Base"

1.1 Scope

The scope of this contribution is to define the wmanIfMib for IEEE 802.16REVd/D5 specification [3], covering both SS (Subscriber Station) and BS (Base Station). 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 P802.16-REVd/D5-2004, "Draft IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems", May 13, 2004
- [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.

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 wmanIfMib 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 a SS enters into a BS network.

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 from NMS, or via a SNMP proxy in the BS.

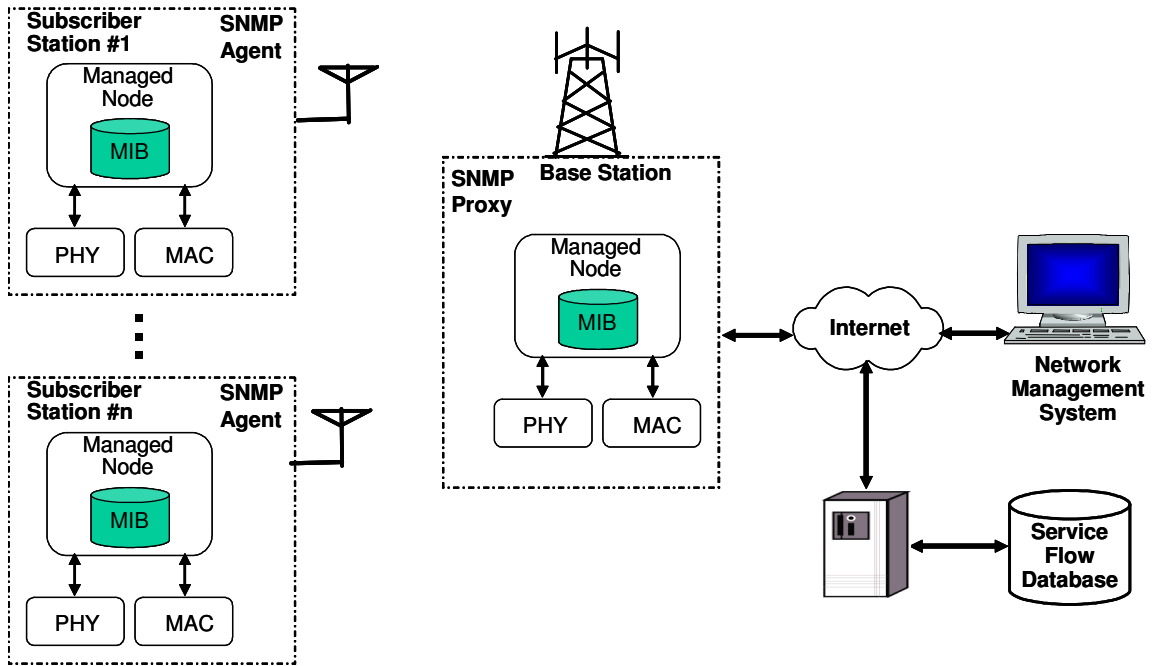


Figure 1 - BWA Network Management Reference Model

3. Relationship with Interface MIB

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

3.1 MIB-2 Integration

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

```

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

Therefore, upon `wmanIfMib` being approved by the IETF, this MIB can be accessed through

```

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

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

```

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

Before the approval of the IETF; however, `wmanIfMib` is temporary located under enterprise via

```

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

Or

```

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

3.2 Usage of MIB-II Tables

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

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

34
35
36
37
38

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

<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/D5 standard [3].

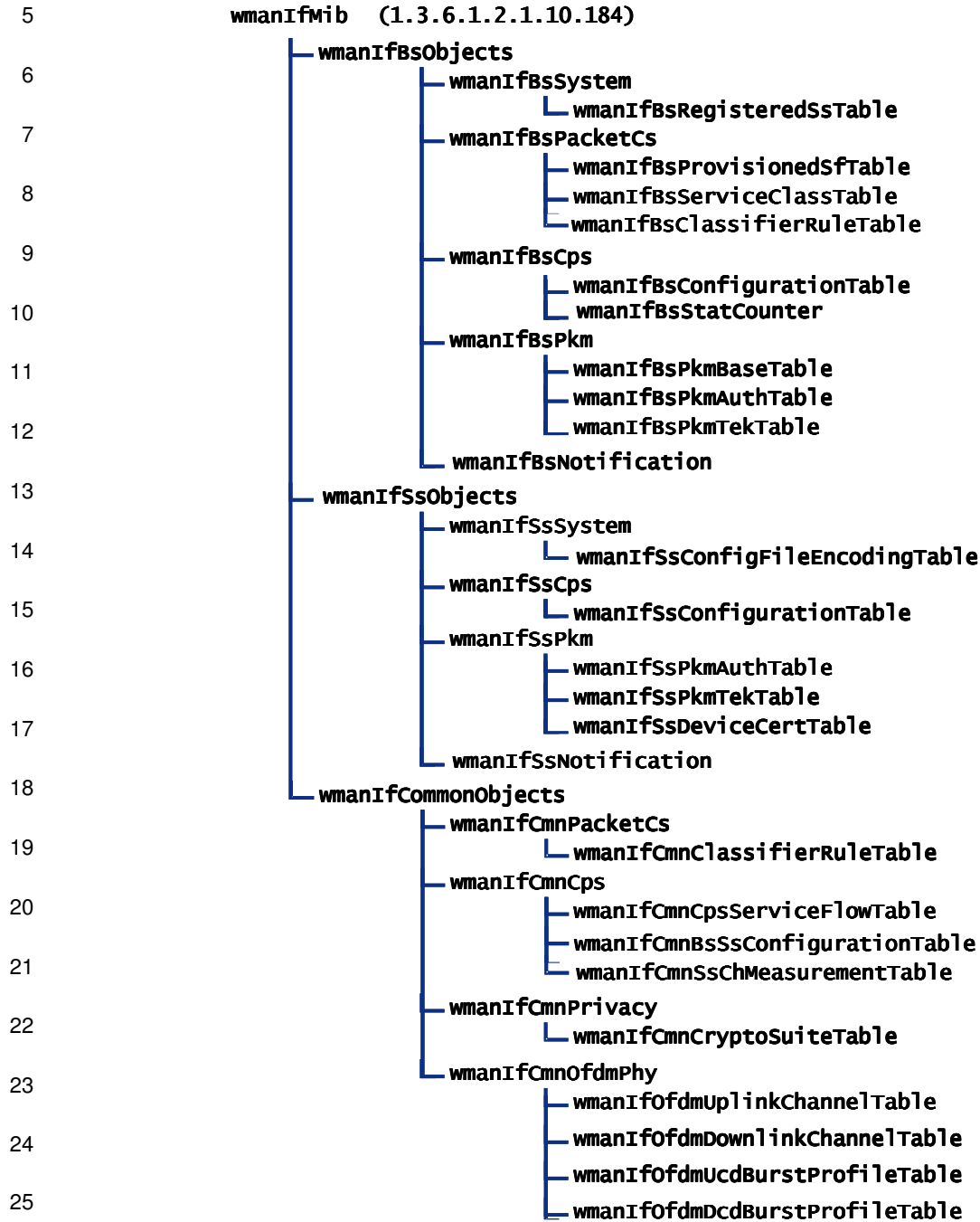


Figure 2 – wmanIfMib Structure

- 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, Each entry contains the
13 information of SS that has been registered through REG-REQ message as defined in
14 section 6.3.2.3.7 in [3].

15 4.1.2 **wmanIfBsPacketCs**

16 wmanIfBsPacketCs group contains BS managed objects relating to the Packet CS
17 management entity layer in Figure 1 of [3].

18 4.1.2.1 wmanIfBsProvisionedSfTable

19 This table is doubly indexed by SS MAC address and Service Flow ID and contains
20 provisioned service flow profiles, Per SS. It contains the service flow attributes that
21 have been pre-provisioned by NMS.

22 4.1.2.2 wmanIfBsServiceClassTable

23 This table is provisioned and is indexed by QoS profile index. Each entry of the table
24 contains QoS parameter set, as defined in section 6.3.14 and 11.13 in [3].

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

32

33

34

35

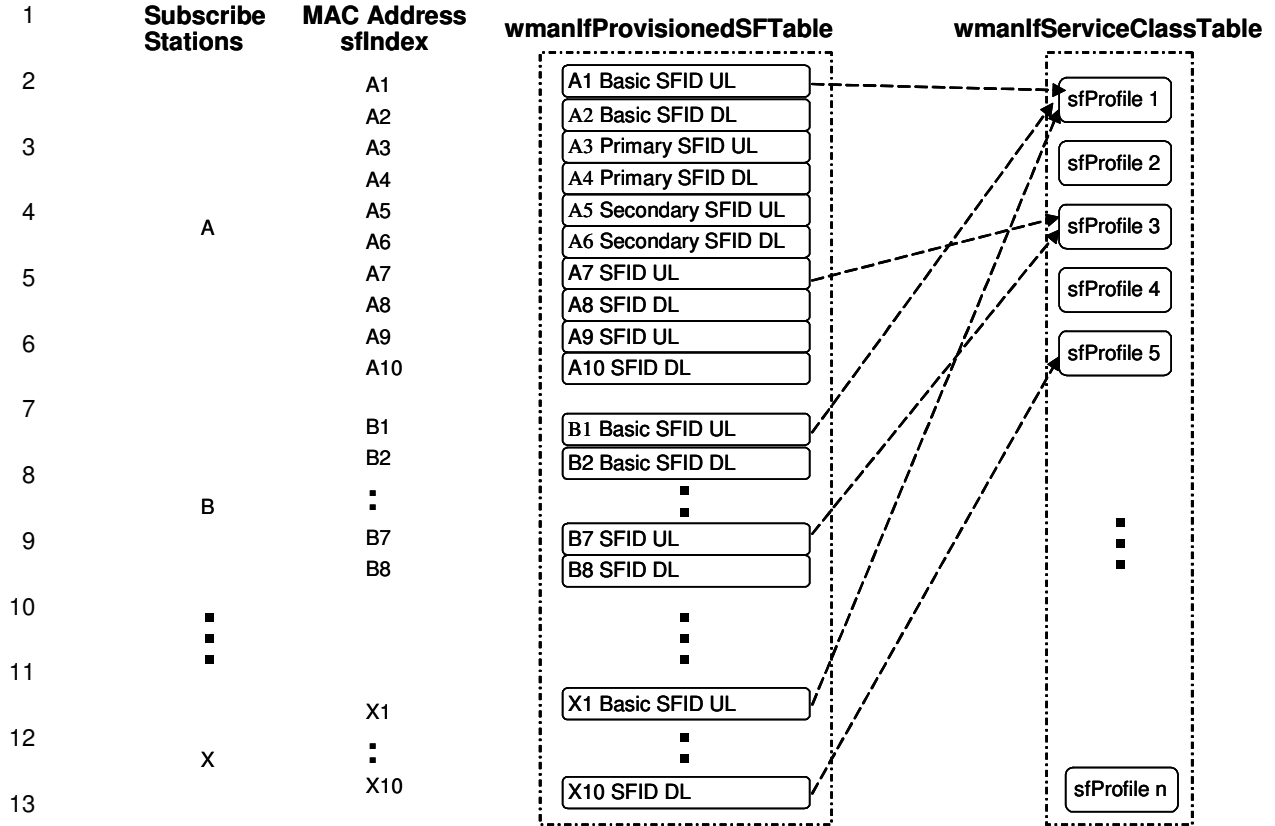


Figure 3 – Service Classes – Service Flows Mapping

15 **4.1.2.3 wmanIfBsClassifierRuleTable**

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

18 **4.1.3 wmanIfBsCps**

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

21 **4.1.3.1 wmanIfBsConfigurationTable**

22 This table contains objects for BS system parameters and constants as defined in
 23 section 10.1, Table 340 of [3]. It is indexed by BS Id.

24 **4.1.3.2 wmanIfBsChMeasurementTable**

25 This table is indexed by BS ifIndex and contains statistics about the channel
 26 measurement.

27 **4.1.4 wmanIfBsPkm**

28 wmanIfBsPkm group contains BS managed objects relating to the MAC CPS privacy
 29 management entity section in figure 1 of [3].

- 1 4.1.4.1 **wmanIfBsPkmBaselineTable**
2 This table is indexed by BS ifIndex and contains base station PKM operational
3 parameters described in section 10.2 and table 341 of [3].
- 4 4.1.4.2 **wmanIfBsPkmAuthTable**
5 This table is double indexed by ifIndex and SsMacAddress and contains runtime
6 subscriber station authentication and authorization parameters for each base station.
- 7 4.1.4.3 **wmanIfBsPkmTekTable**
8 This table is double indexed by ifIndex and SAId and contains runtime Security
9 association parameters for each base station.
- 10 4.1.5 **wmanIfBsNotification**
11 wmanIfBsNotification group contains BS traps to report fault events and exceptions,
12 such as power status, RSSI threshold crossing.
- 13 **4.2 wmanIfSsObjects**
- 14 4.2.1 **wmanSsSystem**
15 wmanIfSsSystem group contains subscriber station system level objects.
- 16 4.2.1.1 **wmanIfSsConfigFileEncodingTable**
17 This table is indexed by SS index, and contain configuration file information about the
18 subscriber station such as manufacturer, hardware model, serial number, and software
19 or firmware revision.
- 20 4.2.2 **wmanIfSsCps**
21 wmanIfSsCpsParameters group contains subscriber station manageable objects
22 relating to the MAC CPS management entity layer in figure 1 of [3].
- 23 4.2.2.1 **wmanIfSsConfigurationTable**
24 This table is indexed by SS Id and contains objects for SS system parameters and
25 constants as defined in section 10.1, Table 341 of [3].
- 26 4.2.2.2 **wmanIfSsStatisticsCountersTable**
27 This object contains the performance monitoring data for SS.
- 28 4.2.3 **wmanIfSsPkm**
29 wmanIfSsPkmParameters group contains subscriber station manageable objects
30 relating to the MAC CPS privacy management entity section in figure 1 of [3].
- 31 4.2.3.1 **wmanIfSsPkmAuthTable**
32 This table is indexed by SS MAC address and contains subscriber station
33 authentication and authorization parameters including those described in section 10.2
34 and table 342 of [3].

1 4.2.3.2 **wmanIfSsPkmTekTable**

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

4 4.2.3.3 **wmanIfSsPkmCertificatesTable**

5 This table is indexed by SS MAC address and contains subscriber station and SS
6 manufacturer certificates.

7 4.2.4 **wmanIfSsTraps**

8 wmanIfBsTraps group contains SS traps to report fault events and exceptions, such as
9 power status, RSSI threshold crossing.

10 **4.3 wmanIfCommonObjects**

11 4.3.1 **wmanIfCmnPacketCs**

12 4.3.1.1 **wmanIfCmnClassifierRuleTable**

13 wmanIfClassifierRuleTable is indexed by service flow ID and contains runtime classifier
14 rules screening criteria for each service flow as described in section 11.13.19.3.4 of [3].

15 4.3.2 **wmanIfCmnCps**

16 4.3.2.1 **wmanIfCmnServiceFlowTable**

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

21 A Service Flow is represented by parameters, such as

- 22 ▪ Service Flow common parameters, like SFID and CID
- 23 ▪ Classifiers associated with Service Flow, see [3] , 5.2.2, 5.2.5 – 5.2.7
- 24 ▪ Service Flow QoS parameters like QoS parameters of specific Service Flow,
25 like Max Sustained Traffic Rate, QoS status (admitted etc.)
- 26 ▪ Service Flow Header Suppression parameters like associated classifier and
27 PHS rule, see [3] , 5.2.4

28 4.3.2.2 **wmanIfCmnBsSsConfigurationTable**

29 This table is indexed by SS Id and contains objects for SS system parameters and
30 constants as defined in section 10.1, Table 341 of [3].

31 4.3.2.3 **wmanIfCmnSsChMeasurementTable**

32 This object contains the channel measurement table for SS.

1 4.3.3 **wmanIfCmnPrivacy**

2 4.3.3.1 **wmanIfCmnCryptoSuiteTable**

3 This table is doubly indexed by `ifIndex` and `wmanIfCryptoSuiteIndex` and contains
4 supported crypto suites for the particular SS and other crypto parameters such as key
5 lifetimes. See sections 11.9.14 and 11.9.15 of [3].

6 4.3.4 **wmanIfCmnOfdmPhy**

7 `wmanIfOfdmPhy` is a group containing objects specific to OFDM PHY.

8 4.3.4.1 **wmanIfOfdmUplinkChannelTable**

9 This table contains the uplink channels that the BS is able to receive. In the SS, this
10 table should have an entry indicating the uplink channel that the SS can transmit. Each
11 entry contains the parameters needed to describe uplink channel descriptor as defined
12 in section 11, Table 347 and 350 of [3].

13 4.3.4.2 **wmanIfOfdmDownlinkChannelTable**

14 This table contains the downlink channels that the BS is able to transmit. In the SS, this
15 table should have an entry indicating the downlink channel that the SS can receive.
16 Each entry contains the parameters needed to describe downlink channel descriptor as
17 defined in section 11, Table 356 of [3].

18 4.3.4.3 **wmanIfOfdmUcdBurstProfileTable**

19 Each entry in this table contains the parameters needed for the UCD burst profile as
20 defined in section 11, Table 354 of [3].

21 4.3.4.4 **wmanIfOfdmDcdBurstProfileTable**

22 `wmanIfDcdBurstProfileTable` – Each entry in this table contains the parameters
23 needed for the UCD burst profile as defined in section 11, Table 360 of [3].

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

```
2
3 WMAN-IF-MIB DEFINITIONS ::= BEGIN
4
5     IMPORTS
6         MODULE-IDENTITY,
7         OBJECT-TYPE,
8         NOTIFICATION-TYPE,
9         Unsigned32,
10        Integer32,
11        Counter32,
12        Counter64,
13        TimeTicks,
14        IpAddress,
15        transmission
16            FROM SNMPv2-SMI
17        SnmpAdminString
18            FROM SNMP-FRAMEWORK-MIB
19        TEXTUAL-CONVENTION,
20        MacAddress,
21        RowStatus,
22        TruthValue,
23        DateAndTime,
24        DisplayString,
25        TimeInterval,
26        TimeStamp
27            FROM SNMPv2-TC
28        InetAddressType, InetAddress
29            FROM INET-ADDRESS-MIB
30        OBJECT-GROUP,
31
32        MODULE-COMPLIANCE
33            FROM SNMPv2-CONF
34        ifIndex, InterfaceIndexOrZero
35            FROM IF-MIB;
36
37 wmanIfMib MODULE-IDENTITY
38     LAST-UPDATED      "0408260000Z" -- August 26, 2004
39     ORGANIZATION      "IETF IPCDN Working Group"
40     CONTACT-INFO
41         "              Joey Chou
42         Postal: Intel Corporation
43         5000 W. Chandler Blvd, Chandler, AZ 85227, USA
44         E-mail: joey.chou@intel.com
45
46         Russ Reynolds
47         Postal: Proxim Corporation
48         935 Stewart Drive, Sunnyvale, CA 94085, USA
49         E-mail: RReynolds@proxim.com
50
51         Shlomi Eini
```

```

1         Postal: Airspan Networks
2             Airport city 70100,Israel
3         E-mail: seini@airspan.com
4
5             Bogdan Moldoveanu
6         Postal: Redline Communications Inc.
7             302 Town Centre Blvd., Markham, ON L3R 0E8, Canada
8         E-mail: bmoldoveanu@redlinecommunications.com"
9
10        DESCRIPTION
11        "This MIB Module defines managed objects for 802.16 based
12        Subscriber Station and Base Station."
13        ::= { transmission 184 }
14
15        wmanIfMibObjects OBJECT IDENTIFIER ::= { wmanIfMib 1 }
16        wmanIfBsObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 1 }
17        wmanIfSsObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 2 }
18        wmanIfCommonObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 3 }
19
20        -- Textual Conventions
21        wmanIfSfsSchedulingType ::= TEXTUAL-CONVENTION
22            STATUS current
23            DESCRIPTION
24                "The scheduling service provided by a SC for an
25                upstream service flow. If the parameter is omitted
26                from an upstream QOS Parameter Set, this object takes
27                the value of bestEffort (2). This parameter must be
28                reported as undefined (1) for downstream QOS Parameter
29                Sets."
30            SYNTAX INTEGER {undefined(1),
31                    bestEffort(2),
32                    nonRealTimePollingService(3),
33                    realTimePollingService(4),
34                    unsolicitedGrantService(6)}
35
36        --
37        -- BS object group - containing tables and objects to be implemented in
38        -- the Base station
39        --
40        -- wmanIfBsSystem contain the Base Station system objects
41        wmanIfBsSystem OBJECT IDENTIFIER ::= { wmanIfBsObjects 1 }
42
43        wmanIfBsRegisteredSsTable OBJECT-TYPE
44            SYNTAX SEQUENCE OF wmanIfBsRegisteredSsEntry
45            MAX-ACCESS not-accessible
46            STATUS current
47            DESCRIPTION
48                "This table contains entries of Ss that have been
49                registered to the BS through REG-REQ message"
50            REFERENCE
51                "Section 6.3.2.3.7 in IEEE 802.16REVd/D5-2004"
52            ::= { wmanIfBsSystem 1 }
53
54        wmanIfBsRegisteredSsEntry OBJECT-TYPE

```



```

1      SYNTAX      WmanIfBsRegisteredSsEntry
2      MAX-ACCESS  not-accessible
3      STATUS      current
4      DESCRIPTION
5          "This table provides one row for each SS that has been
6              registered in the BS, and is indexed by
7              wmanIfBsSsIdIndex. The primary index is the ifIndex
8              with an ifType of propBwAp2Mp, indicating the BS sector
9              with which the SS is associated. wmanIfBsSsIdIndex
10             identifies the SS being registered."
11     INDEX { ifIndex, wmanIfBsSsIdIndex }
12     ::= { wmanIfBsRegisteredSsTable 1 }
13
14     WmanIfBsRegisteredSsEntry ::= SEQUENCE {
15         wmanIfBsSsIdIndex          Unsigned32,
16         wmanIfBsSsMacAddress       MacAddress,
17         wmanIfBsSsBasicCid         INTEGER,
18         wmanIfBsSsPrimaryCid       INTEGER,
19         wmanIfBsSsSecondaryCid     INTEGER,
20         wmanIfBsSsHmacTuple        OCTET STRING,
21         wmanIfBsSsUlCidSupport     INTEGER,
22         wmanIfBsSsManagementSupport INTEGER,
23         wmanIfBsSsArqSupport       INTEGER,
24         wmanIfBsSsDsxFLOWControl   INTEGER,
25         wmanIfBsSsMacCrcSupport    INTEGER,
26         wmanIfBsSsMcaFLOWControl   INTEGER,
27         wmanIfBsSsMcpGroupCidSupport INTEGER,
28         wmanIfBsSsPkmFLOWControl   INTEGER,
29         wmanIfBsSsAuthorizationPolicyControl BITS,
30         wmanIfBsSsMaxNumOfSupportedSA INTEGER,
31         wmanIfBsSsIpVersion        INTEGER,
32         wmanIfBsSsMacCsSupportBitMap BITS,
33         wmanIfBsSsMaxNumOfClassifier INTEGER,
34         wmanIfBsSsPhsSupport       INTEGER,
35         wmanIfBsSsIpManagementSupport INTEGER,
36         wmanIfBsSs2ndMgmtArqEnable TruthValue,
37         wmanIfBsSs2ndMgmtArqWindowSize INTEGER,
38         wmanIfBsSs2ndMgmtArqFragmentLifetime INTEGER,
39         wmanIfBsSs2ndMgmtArqSyncLossTimeout INTEGER,
40         wmanIfBsSs2ndMgmtArqDeliverInOrder TruthValue,
41         wmanIfBsSs2ndMgmtArqRxPurgeTimeout INTEGER,
42         wmanIfBsSsVendorIdEncoding OCTET STRING
43     }
44
45     wmanIfBsSsIdIndex OBJECT-TYPE
46         SYNTAX      Unsigned32 (1 .. 4294967295)
47         MAX-ACCESS  read-only
48         STATUS      current
49         DESCRIPTION
50             "wmanIfBsSsIdIndex identifies the SS that is registered."
51         ::= { wmanIfBsRegisteredSsEntry 1 }
52
53     wmanIfBsSsMacAddress OBJECT-TYPE
54         SYNTAX      MacAddress

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The MAC address of SS is received from the RNG-REQ
5           message. When SS registers, this MAC address is entered
6           into the table, and used as the identifier to the SS."
7      REFERENCE
8          "Section 6.3.2.3.6 in IEEE 802.16REVd/D5-2004"
9      ::= { wmanIfBsRegisteredSsEntry 2 }
10
11     wmanIfBsSsBasicCid OBJECT-TYPE
12         SYNTAX      INTEGER
13         MAX-ACCESS  read-only
14         STATUS      current
15         DESCRIPTION
16             "The value of this object indicates the SS's basic CID
17              that was sent in the RNG-RSP message."
18         REFERENCE
19             "Section 6.3.2.3.8 in IEEE 802.16REVd/D5-2004"
20         ::= { wmanIfBsRegisteredSsEntry 3 }
21
22     wmanIfBsSsPrimaryCid OBJECT-TYPE
23         SYNTAX      INTEGER
24         MAX-ACCESS  read-only
25         STATUS      current
26         DESCRIPTION
27             "The value of this object indicates the primary CID of the
28              SS received from the RNG-RSP message."
29         REFERENCE
30             "Section 6.3.2.3.8 in IEEE 802.16REVd/D5-2004"
31         ::= { wmanIfBsRegisteredSsEntry 4 }
32
33     wmanIfBsSsSecondaryCid OBJECT-TYPE
34         SYNTAX      INTEGER
35         MAX-ACCESS  read-only
36         STATUS      current
37         DESCRIPTION
38             "The value of this object indicates the secondary
39              management CID present in the REG-RSP message. The value
40              should be null if the 2nd management channel is not
41              available."
42         REFERENCE
43             "Section 6.4.2.3.8 in IEEE 802.16REVd/D5-2004"
44         ::= { wmanIfBsRegisteredSsEntry 5 }
45
46     wmanIfBsSsHmacTuple OBJECT-TYPE
47         SYNTAX      OCTET STRING
48         MAX-ACCESS  read-only
49         STATUS      current
50         DESCRIPTION
51             "This parameter contains the HMAC Key Sequence Number
52              concatenated with an HMAC-Digest message during the
53              authentication. The HMAC Key Sequence Number is stored
54              in the four least significant bits of the first byte of

```

```

1         the HMAC Tuple, and the most significant four bits
2         are reserved."
3     REFERENCE
4         "Section 11.1.2 in IEEE 802.16REvd/D5-2004"
5     ::= { wmanIfBsRegisteredSsEntry 6 }
6
7     wmanIfBsSsUplinkSupport OBJECT-TYPE
8         SYNTAX          INTEGER
9         MAX-ACCESS      read-only
10        STATUS          current
11        DESCRIPTION
12            "This object shows the number of Uplink CIDs the SS can
13            support."
14        REFERENCE
15            "Section 11.7.4 in IEEE 802.16REvd/D5-2004"
16        ::= { wmanIfBsRegisteredSsEntry 7 }
17
18        wmanIfBsSsManagementSupport OBJECT-TYPE
19            SYNTAX          INTEGER {unmanagedSs(0),
20                            managedSs(1)}
21            MAX-ACCESS      read-only
22            STATUS          current
23            DESCRIPTION
24                "This object indicates whether or not the SS is managed."
25            REFERENCE
26                "Section 11.7.1.1 in IEEE 802.16REvd/D5-2004"
27            ::= { wmanIfBsRegisteredSsEntry 8 }
28
29        wmanIfBsSsArqSupport OBJECT-TYPE
30            SYNTAX          INTEGER {arqNotSupported(0),
31                            arqSupported(1)}
32            MAX-ACCESS      read-only
33            STATUS          current
34            DESCRIPTION
35                "This object indicates whether the SS support ARQ."
36            REFERENCE
37                "Section 11.7.6.1 in IEEE 802.16REvd/D5-2004"
38            ::= { wmanIfBsRegisteredSsEntry 9 }
39
40        wmanIfBsSsDsxFlowControl OBJECT-TYPE
41            SYNTAX          INTEGER (0..255)
42            MAX-ACCESS      read-only
43            STATUS          current
44            DESCRIPTION
45                "This object specifies the maximum number of concurrent
46                DSA, DSC, or DSD transactions that may be outstanding."
47            REFERENCE
48                "Section 11.7.6.2 in IEEE 802.16REvd/D5-2004"
49            DEFVAL          { 0 }
50            ::= { wmanIfBsRegisteredSsEntry 10 }
51
52        wmanIfBsSsMacCrcSupport OBJECT-TYPE
53            SYNTAX          INTEGER {noMacCrcSupport(0),
54                            macCrcSupport(1)}

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This object indicates whether or not the SS supports MAC
5          level CRC."
6      REFERENCE
7          "Section 11.7.6.3 in IEEE 802.16REvd/D5-2004"
8      DEFVAL      { 1 }
9      ::= { wmanIfBsRegisteredSsEntry 11 }
10
11     wmanIfBsSsMcaFlowControl OBJECT-TYPE
12         SYNTAX      INTEGER (0..255)
13         MAX-ACCESS  read-only
14         STATUS      current
15         DESCRIPTION
16             "This object specifies the maximum number of concurrent
17             MCA transactions that may be outstanding."
18         REFERENCE
19             "Section 11.7.6.4 in IEEE 802.16REvd/D5-2004"
20         DEFVAL      { 0 }
21         ::= { wmanIfBsRegisteredSsEntry 12 }
22
23     wmanIfBsSsMcpGroupCidSupport OBJECT-TYPE
24         SYNTAX      INTEGER (0..255)
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "This object indicates the maximum number of
29             simultaneous Multicast Polling Groups the SS is
30             capable of belonging to."
31         REFERENCE
32             "Section 11.7.6.5 in IEEE 802.16REvd/D5-2004"
33         DEFVAL      { 0 }
34         ::= { wmanIfBsRegisteredSsEntry 13 }
35
36     wmanIfBsSsPkmFlowControl OBJECT-TYPE
37         SYNTAX      INTEGER (0..255)
38         MAX-ACCESS  read-only
39         STATUS      current
40         DESCRIPTION
41             "This object specifies the maximum number of concurrent PKM
42             transactions that may be outstanding."
43         REFERENCE
44             "Section 11.7.6.6 in IEEE 802.16REvd/D5-2004"
45         DEFVAL      { 0 }
46         ::= { wmanIfBsRegisteredSsEntry 14 }
47
48     wmanIfBsSsAuthorizationPolicyControl OBJECT-TYPE
49         SYNTAX      BITS {ieee802-16PrivacySupported(0),
50                         reserved1(1),
51                         reserved2(2),
52                         reserved3(3),
53                         reserved4(4),
54                         reserved5(5),

```

```

1             reserved6(6),
2             reserved7(7)}
3     MAX-ACCESS read-only
4     STATUS      current
5     DESCRIPTION
6         "This object specifies authorization policy that both SS and
7         BS need to negotiate and synchronize. A bit value of 0 =
8         not supported, 1 = supported. If this field is omitted, then
9         both SS and BS shall use the IEEE 802.16 security,
10        constituting X.509 digital certificates and the RSA public
11        key encryption algorithm, as authorization policy."
12    REFERENCE
13        "Section 11.7.8.7 in IEEE 802.16REvd/D5-2004"
14    ::= { wmanIfBsRegisteredSsEntry 15 }
15
16    wmanIfBsSsMaxNumOfSupportedSA OBJECT-TYPE
17        SYNTAX      INTEGER (0..255)
18        MAX-ACCESS read-only
19        STATUS      current
20        DESCRIPTION
21            "This field specifies maximum number of supported security
22            association of the SS."
23        REFERENCE
24            "Section 11.7.8.8 in IEEE 802.16REvd/D5-2004"
25        DEFVAL      { 1 }
26        ::= { wmanIfBsRegisteredSsEntry 16 }
27
28    wmanIfBsSsIpVersion OBJECT-TYPE
29        SYNTAX      INTEGER {ipv4(1),
30                        ipv6(2)}
31        MAX-ACCESS read-only
32        STATUS      current
33        DESCRIPTION
34            "This object indicates the version of IP used on the
35            Secondary Management Connection. The value should be numm
36            if the 2nd management CID doesn't exist."
37        REFERENCE
38            "Section 11.7.2.1 in IEEE 802.16REvd/D5-2004"
39        ::= { wmanIfBsRegisteredSsEntry 17 }
40
41    wmanIfBsSsMacCsSupportBitMap OBJECT-TYPE
42        SYNTAX      BITS {atm(0),
43                        packetIpv4(1),
44                        packetIpv6(2),
45                        packet802-3(3),
46                        packet802-1Q(4),
47                        packetIpv4Over802-3(5),
48                        packetIpv6Over802-3(6),
49                        packetIpv4Over802-1Q(7),
50                        packetIpv6Over802-1Q(8)}
51        MAX-ACCESS read-only
52        STATUS      current
53        DESCRIPTION
54            "This object indicates the set of MAC convergence

```

```

1         sublayer support. When a bit is set, it indicates
2         the corresponding CS feature is supported."
3     REFERENCE
4         "Section 11.7.5.1 in IEEE 802.16REVd/D5-2004"
5     ::= { wmanIfBsRegisteredSsEntry 18 }
6
7     wmanIfBsSsMaxNumOfClassifier OBJECT-TYPE
8         SYNTAX      INTEGER
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            "This object indicates the maximum number of admitted
13            Classifiers that the SS is allowed to have."
14        REFERENCE
15            "Section 11.7.5.2 in IEEE 802.16REVd/D5-2004"
16        DEFVAL      { 0 }
17        ::= { wmanIfBsRegisteredSsEntry 19 }
18
19        wmanIfBsSsPhsSupport OBJECT-TYPE
20            SYNTAX      INTEGER {noPhsSupport(0),
21                            atmPhsSupport(1),
22                            packetPhsSupport(2)}
23            MAX-ACCESS  read-only
24            STATUS      current
25            DESCRIPTION
26                "This object indicates indicates the level of PHS support."
27            REFERENCE
28                "Section 11.7.5.3 in IEEE 802.16REVd/D5-2004"
29            DEFVAL      { 0 }
30            ::= { wmanIfBsRegisteredSsEntry 20 }
31
32        wmanIfBsSsIpManagementSupport OBJECT-TYPE
33            SYNTAX      INTEGER {unmanaged(0),
34                            ipManaged(1)}
35            MAX-ACCESS  read-only
36            STATUS      current
37            DESCRIPTION
38                "The IP management mode parameter dictates whether
39                the provider intends to manage the SS on an ongoing
40                basis via IP-based mechanisms."
41            REFERENCE
42                "Section 11.7.3 in IEEE 802.16REVd/D5-2004"
43            ::= { wmanIfBsRegisteredSsEntry 21 }
44
45        wmanIfBsSs2ndMgmtArqEnable OBJECT-TYPE
46            SYNTAX      TruthValue
47            MAX-ACCESS  read-only
48            STATUS      current
49            DESCRIPTION
50                "True(1) ARQ enabling is requested for the 2nd
51                management channel."
52            REFERENCE
53                "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
54            ::= { wmanIfBsRegisteredSsEntry 22 }

```

```

1
2 wmanIfBsSs2ndMgmtArqWindowSize OBJECT-TYPE
3     SYNTAX      INTEGER (1 .. 1024)
4     MAX-ACCESS  read-only
5     STATUS      current
6     DESCRIPTION
7         "Indicates the maximum number of unacknowledged
8         fragments at any time for 2nd management channel."
9     REFERENCE
10        "Section 11.13.20 in IEEE 802.16REvd/D5-2004"
11        ::= { wmanIfBsRegisteredSsEntry 23 }
12
13 wmanIfBsSs2ndMgmtArqFragmentLifetime OBJECT-TYPE
14     SYNTAX      INTEGER (0 .. 65535)
15     UNITS       "10 us"
16     MAX-ACCESS  read-only
17     STATUS      current
18     DESCRIPTION
19         "The maximum time interval an ARQ fragment will be
20         managed by the transmitter ARQ machine, once
21         initial transmission of the fragment has occurred.
22         If transmission or retransmission of the fragment
23         is not acknowledged by the receiver before the
24         time limit is reached, the fragmnet is discarded.
25         A value of 0 means Infinite."
26     REFERENCE
27        "Section 11.13.20 in IEEE 802.16REvd/D5-2004"
28     DEFVAL      {0}
29     ::= { wmanIfBsRegisteredSsEntry 24 }
30
31 wmanIfBsSs2ndMgmtArqSyncLossTimeout OBJECT-TYPE
32     SYNTAX      INTEGER (0 .. 65535 )
33     UNITS       "10 us"
34     MAX-ACCESS  read-only
35     STATUS      current
36     DESCRIPTION
37         "The maximum interval before declaring a loss
38         of synchronization of the sender and receiver
39         state machines. A value of 0 means Infinite."
40     REFERENCE
41        "Section 11.13.20 in IEEE 802.16REvd/D5-2004"
42     DEFVAL      {0}
43     ::= { wmanIfBsRegisteredSsEntry 25 }
44
45 wmanIfBsSs2ndMgmtArqDeliverInOrder OBJECT-TYPE
46     SYNTAX      TruthValue
47     MAX-ACCESS  read-only
48     STATUS      current
49     DESCRIPTION
50         "Indicates whether or not data is to be delivered
51         by the receiving MAC to its client application
52         in the order in which data was handed off to the
53         originating MAC."
54     REFERENCE

```

```

1         "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
2         ::= { wmanIfBsRegisteredSsEntry 26 }
3
4     wmanIfBsSs2ndMgmtArqRxPurgeTimeout OBJECT-TYPE
5         SYNTAX      INTEGER (0 .. 65535)
6         UNITS       "10 us"
7         MAX-ACCESS  read-only
8         STATUS      current
9         DESCRIPTION
10            "Indicates the time interval the ARQ window is advanced
11             after a fragment is received. A value of 0 means Infinite."
12         REFERENCE
13            "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
14         DEFVAL     {0}
15         ::= { wmanIfBsRegisteredSsEntry 27 }
16
17     wmanIfBsSsVendorIdEncoding OBJECT-TYPE
18         SYNTAX      OCTET STRING (SIZE(3))
19         MAX-ACCESS  read-only
20         STATUS      current
21         DESCRIPTION
22            "The value field contains the vendor identification
23             specified by the 3 byte vendor-specific organizationally
24             unique identifier of the SS or BS MAC address. A vendor ID
25             used in a REG-REQ shall be the Vendor ID of the SS sending
26             the request. A vendor ID used in a REG-RSP shall be the
27             Vendor ID of the BS sending the response."
28         REFERENCE
29            "Section 11.1.5 in IEEE 802.16REVd/D5-2004"
30         ::= { wmanIfBsRegisteredSsEntry 28 }
31
32     --
33     -- wmanIfBsPacketCs contain the Base Station Packet Convergence Sublayer
34     -- objects
35     wmanIfBsPacketCs OBJECT IDENTIFIER ::= { wmanIfBsObjects 2 }
36
37     wmanIfBsProvisionedSfTable OBJECT-TYPE
38         SYNTAX      SEQUENCE OF WmanIfBsProvisionedSfEntry
39         MAX-ACCESS  not-accessible
40         STATUS      current
41         DESCRIPTION
42            "This table is doubly indexed (SS MAC address, SF ID) and
43             contains pre-provisioned service flow profiles, Per SS.
44             These connection parameters shall be provisioned for the SS
45             using DSA messages. NMS shall pre-provisioning the service
46             class table - wmanIfBsServiceClassTable by using
47             wmanIfBsServiceClassIndex, and packet classifier rule table
48             - wmanIfBsClassifierRuleTable by using wmanIfBsSfId"
49         REFERENCE
50            "Section 6.4.13 in IEEE 802.16REVd/D5-2004"
51         ::= { wmanIfBsPacketCs 1 }
52
53     wmanIfBsProvisionedSfEntry OBJECT-TYPE
54         SYNTAX      WmanIfBsProvisionedSfEntry

```



```

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

```

```

1         service class or QoS parameters for such service flow.
2         If no associated entry in wmanIfBsServiceClassTable
3         exists, this object returns a value of zero."
4 ::= { wmanIfBsProvisionedSfEntry 4 }
5
6 wmanIfBsServiceClassName OBJECT-TYPE
7     SYNTAX      DisplayString (SIZE(1..32))
8     MAX-ACCESS  read-create
9     STATUS      current
10    DESCRIPTION
11        "Refers to the Service Class Name"
12    REFERENCE
13        "Section 11.13.7 in IEEE 802.16REVd/D5-2004"
14 ::= { wmanIfBsProvisionedSfEntry 5 }
15
16 wmanIfBsSfState OBJECT-TYPE
17     SYNTAX      INTEGER {provisioned(1),
18                        admitted(2),
19                        active(3)}
20     MAX-ACCESS  read-create
21     STATUS      current
22     DESCRIPTION
23        "wmanIfBsSfState determines the state of a service flow.
24         provisioned state: A service flow is provisioned but
25         not resource is reserved yet
26         admitted state: service flow has resources reserved.
27         active state: has resources committed by the BS (e.g., is
28         actively sending maps containing unsolicited grants for a
29         UGS-based service flow)"
30     REFERENCE
31        "Section 6.4.13.6, in IEEE 802.16REVd/D5-2004"
32 ::= { wmanIfBsProvisionedSfEntry 6 }
33
34 wmanIfBsSfProvisionedTime OBJECT-TYPE
35     SYNTAX      TimeStamp
36     MAX-ACCESS  read-create
37     STATUS      current
38     DESCRIPTION
39        "Indicates the date and time when the service flow is
40         provisioned."
41 ::= { wmanIfBsProvisionedSfEntry 7 }
42
43 wmanIfBsProvisionedSfRowStatus OBJECT-TYPE
44     SYNTAX      RowStatus
45     MAX-ACCESS  read-create
46     STATUS      current
47     DESCRIPTION
48        "This object is used to create a new row or modify or
49         delete an existing row in this table.
50
51         If the implementator of this MIB has chosen not
52         to implement 'dynamic assignment' of profiles, this
53         object is not useful and should return noSuchName
54         upon SNMP request."

```

```

1      ::= { wmanIfBsProvisionedSfEntry 8 }
2
3  wmanIfBsServiceClasTable OBJECT-TYPE
4      SYNTAX      SEQUENCE OF wmanIfBsServiceClassEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8          "This table is provisioned and is indexed by
9          wmanIfBsQoSProfileIndex. Each entry of the table contains
10         corresponding service flow characteristic attributes
11         (e.g. QoS parameter set). The value of
12         wmanIfBsQoSProfileIndex is obtained from
13         wmanIfBsServiceClassIndex in wmanIfBsProvisionedSfTable"
14     REFERENCE
15         "Section 6.4.13.4 in IEEE 802.16REvD/D5-2004"
16     ::= { wmanIfBsPacketCs 2 }
17
18  wmanIfBsServiceClassEntry OBJECT-TYPE
19      SYNTAX      wmanIfBsServiceClassEntry
20      MAX-ACCESS  not-accessible
21      STATUS      current
22      DESCRIPTION
23          "This table provides one row for each service class"
24      INDEX { wmanIfBsQoSProfileIndex }
25      ::= { wmanIfBsServiceClasTable 1 }
26
27  wmanIfBsServiceClassEntry ::= SEQUENCE {
28      wmanIfBsQoSProfileIndex          INTEGER,
29      wmanIfBsQoSServiceClassName     DisplayString,
30      wmanIfBsQoSTrafficPriority       INTEGER,
31      wmanIfBsQoSMaxSustainedRate     INTEGER,
32      wmanIfBsQoSMaxTrafficBurst      INTEGER,
33      wmanIfBsQoSMinReservedRate     INTEGER,
34      wmanIfBsQOSToleratedJitter      INTEGER,
35      wmanIfBsQoSMaxLatency           INTEGER,
36      wmanIfBsQoSFixedVsVariableSduInd INTEGER,
37      wmanIfBsQOSSduSize              INTEGER,
38      wmanIfBsQoSScSchedulingType     wmanIfSfSchedulingType,
39      wmanIfBsQoSScArqEnable          TruthValue,
40      wmanIfBsQoSScArqWindowSize      INTEGER,
41      wmanIfBsQoSScArqFragmentLifetime INTEGER,
42      wmanIfBsQoSScArqSyncLossTimeout INTEGER,
43      wmanIfBsQoSScArqDeliverInOrder  TruthValue,
44      wmanIfBsQoSScArqRxPurgeTimeout  INTEGER,
45      wmanIfBsQoSScFragmentLen       INTEGER,
46      wmanIfBsQoSScMinRsvdTolerableRate INTEGER,
47      wmanIfBsQoSReqTxPolicy          BITS,
48      wmanIfBsQOSServiceClassRowStatus RowStatus
49  }
50
51  wmanIfBsQoSProfileIndex OBJECT-TYPE
52      SYNTAX      INTEGER (1 .. 1000)
53      MAX-ACCESS  not-accessible
54      STATUS      current

```

```

1      DESCRIPTION
2          "The index value which uniquely identifies an entry
3          in the wmanIfBsServiceClassTable"
4      ::= { wmanIfBsServiceClassEntry 1 }
5
6      wmanIfBsQoSServiceClassName OBJECT-TYPE
7          SYNTAX      DisplayString (SIZE(1..32))
8          MAX-ACCESS  read-create
9          STATUS      current
10         DESCRIPTION
11             "Refers to the Service Class Name"
12         REFERENCE
13             "Section 11.13.7 in IEEE 802.16REVd/D5-2004"
14         ::= { wmanIfBsServiceClassEntry 2 }
15
16         wmanIfBsQoSTrafficPriority OBJECT-TYPE
17             SYNTAX      INTEGER (0..7)
18             MAX-ACCESS  read-create
19             STATUS      current
20             DESCRIPTION
21                 "The value of this parameter specifies the priority
22                 assigned to a service flow. For uplink service flows,
23                 the BS should use this parameter when determining
24                 precedence in request service and grant generation,
25                 and the SS shall preferentially select contention
26                 Request opportunities for Priority Request CIDs
27                 based on this priority. Higher numbers indicate higher
28                 priority"
29             REFERENCE
30                 "Section 11.13.7 in IEEE 802.16REVd/D5-2004"
31             ::= { wmanIfBsServiceClassEntry 3 }
32
33         wmanIfBsQoSMaxSustainedRate OBJECT-TYPE
34             SYNTAX      INTEGER
35             UNITS      "bps"
36             MAX-ACCESS  read-create
37             STATUS      current
38             DESCRIPTION
39                 "This parameter defines the peak information rate
40                 of the service. The rate is expressed in bits per
41                 second and pertains to the SDUs at the input to
42                 the system."
43             REFERENCE
44                 "Section 11.13.8 in IEEE 802.16REVd/D5-2004"
45             ::= { wmanIfBsServiceClassEntry 4 }
46
47         wmanIfBsQoSMaxTrafficBurst OBJECT-TYPE
48             SYNTAX      INTEGER
49             UNITS      "byte"
50             MAX-ACCESS  read-create
51             STATUS      current
52             DESCRIPTION
53                 "This parameter defines the maximum burst size that
54                 must be accommodated for the service."

```

```

1      REFERENCE
2      "Section 11.13.9 in IEEE 802.16REVd/D5-2004"
3      ::= { wmanIfBsServiceClassEntry 5 }
4
5      wmanIfBsQoSMinReservedRate OBJECT-TYPE
6          SYNTAX      INTEGER
7          UNITS       "bps"
8          MAX-ACCESS  read-create
9          STATUS      current
10         DESCRIPTION
11             "This parameter specifies the minimum rate reserved
12             for this service flow."
13         REFERENCE
14             "Section 11.13.10 in IEEE 802.16REVd/D5-2004"
15         ::= { wmanIfBsServiceClassEntry 6 }
16
17         wmanIfBsQoSToLeratedJitter OBJECT-TYPE
18             SYNTAX      INTEGER
19             UNITS       "millisecond"
20             MAX-ACCESS  read-create
21             STATUS      current
22             DESCRIPTION
23                 "This parameter defines the Maximum delay
24                 variation (jitter) for the connection."
25             REFERENCE
26                 "Section 11.13.15 in IEEE 802.16REVd/D5-2004"
27             ::= { wmanIfBsServiceClassEntry 7 }
28
29         wmanIfBsQoSMaxLatency OBJECT-TYPE
30             SYNTAX      INTEGER
31             UNITS       "millisecond"
32             MAX-ACCESS  read-create
33             STATUS      current
34             DESCRIPTION
35                 "The value of this parameter specifies the maximum
36                 latency between the reception of a packet by the BS
37                 or SS on its network interface and the forwarding
38                 of the packet to its RF Interface."
39             REFERENCE
40                 "Section 11.13.16 in IEEE 802.16REVd/D5-2004"
41             ::= { wmanIfBsServiceClassEntry 8 }
42
43         wmanIfBsQoSFixedVsVariableSduInd OBJECT-TYPE
44             SYNTAX      INTEGER {variableLength(0),
45                             fixedLength(1)}
46             MAX-ACCESS  read-create
47             STATUS      current
48             DESCRIPTION
49                 "The value of this parameter specifies whether the SDUs
50                 on the service flow are fixed-length (0) or
51                 variable-length (1). The parameter is used only if
52                 packing is on for the service flow. The default value
53                 is 0, i.e.,variable-length SDUs."
54             REFERENCE

```

```

1           "Section 11.13.15 in IEEE 802.16REvd/D5-2004"
2     DEFVAL      { 0 }
3     ::= { wmanIfBsServiceClassEntry 9 }
4
5     wmanIfBsQoSsdusize OBJECT-TYPE
6         SYNTAX      INTEGER
7         UNITS       "byte"
8         MAX-ACCESS  read-create
9         STATUS      current
10        DESCRIPTION
11            "The value of this parameter specifies the length of the
12             SDU for a fixed-length SDU service flow. This parameter
13             is used only if packing is on and the service flow is
14             indicated as carrying fixed-length SDUs. The default
15             value is 49 bytes, i.e., VC-switched ATM cells with PHS.
16             The parameter is relevant for both ATM and Packet
17             Convergence Sublayers."
18        REFERENCE
19            "Section 11.13.17 in IEEE 802.16REvd/D4-2004"
20        DEFVAL      { 49 }
21        ::= { wmanIfBsServiceClassEntry 10 }
22
23        wmanIfBsQoSscSchedulingType OBJECT-TYPE
24            SYNTAX      wmanIfSfSchedulingType
25            MAX-ACCESS  read-create
26            STATUS      current
27            DESCRIPTION
28                "Specifies the upstream scheduling service used for
29                 upstream service flow. If the referenced parameter
30                 is not present in the corresponding 802.16 QoS
31                 Parameter Set of an upstream service flow, the
32                 default value of this object is bestEffort(2)."

```

```

1      REFERENCE
2      "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
3      ::= { wmanIfBsServiceClassEntry 13 }
4
5      wmanIfBsQosScArqFragmentLifetime OBJECT-TYPE
6      SYNTAX      INTEGER (0 .. 65535)
7      UNITS       "10 us"
8      MAX-ACCESS  read-create
9      STATUS      current
10     DESCRIPTION
11     "The maximum time interval an ARQ fragment will be
12     managed by the transmitter ARQ machine, once
13     initial transmission of the fragment has occurred.
14     If transmission or retransmission of the fragment
15     is not acknowledged by the receiver before the
16     time limit is reached, the fragmnet is discarded.
17     A value of 0 means Infinite."
18     REFERENCE
19     "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
20     DEFVAL      {0}
21     ::= { wmanIfBsServiceClassEntry 14 }
22
23     wmanIfBsQosScArqSyncLosTimeout OBJECT-TYPE
24     SYNTAX      INTEGER (0 .. 65535 )
25     UNITS       "10 us"
26     MAX-ACCESS  read-create
27     STATUS      current
28     DESCRIPTION
29     "The maximum interval before declaring a loss
30     of synchronization of the sender and receiver
31     state machines. A value of 0 means Infinite."
32     REFERENCE
33     "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
34     DEFVAL      {0}
35     ::= { wmanIfBsServiceClassEntry 15 }
36
37     wmanIfBsQosScArqDeliverInOrder OBJECT-TYPE
38     SYNTAX      TruthValue
39     MAX-ACCESS  read-create
40     STATUS      current
41     DESCRIPTION
42     "Indicates whether or not data is to be delivered
43     by the receiving MAC to its client application
44     in the order in which data was handed off to the
45     originating MAC."
46     REFERENCE
47     "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
48     ::= { wmanIfBsServiceClassEntry 16 }
49
50     wmanIfBsQosScArqRxPurgeTimeout OBJECT-TYPE
51     SYNTAX      INTEGER (0 .. 65535)
52     UNITS       "10 us"
53     MAX-ACCESS  read-create
54     STATUS      current

```

```

1      DESCRIPTION
2          "Indicates the time interval the ARQ window is advanced
3          after a fragment is received. A value of 0 means
4          Infinite."
5      REFERENCE
6          "Section 11.13.20 in IEEE 802.16REVd/D5-2004"
7      DEFVAL      {0}
8      ::= { wmanIfBsServiceClassEntry 17 }
9
10     wmanIfBsQoSscFragmentLen OBJECT-TYPE
11         SYNTAX      INTEGER (32 .. 2040)
12         UNITS       "byte"
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
34             "Section 11.13.11 in IEEE 802.16REVd/D5-2004"
35         ::= { wmanIfBsServiceClassEntry 19 }
36
37     wmanIfBsQoSReqTxPolicy OBJECT-TYPE
38         SYNTAX      BITS {noBroadcastBwReq(0),
39                          reserved1(1),
40                          noPiggybackReq(2),
41                          noFragmentData(3),
42                          noPHS(4),
43                          noSduPacking(5),
44                          noCrc(6),
45                          reserved2(7)}
46         MAX-ACCESS  read-create
47         STATUS      current
48         DESCRIPTION
49             "The value of this parameter provides the capability to
50             specify certain attributes for the associated service
51             flow. An attribute is enabled by setting the
52             corresponding bit position to 1."
53         REFERENCE  "Section 11.13.12 in IEEE 802.16REVd/D5-2004"
54         ::= { wmanIfBsServiceClassEntry 20 }

```



```

1
2  wmanIfBsQoSServiceClassRowStatus 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     ::= { wmanIfBsServiceClassEntry 21 }
15
16  wmanIfBsClassifierRuleTable OBJECT-TYPE
17      SYNTAX      SEQUENCE OF wmanIfBsClassifierRuleEntry
18      MAX-ACCESS  not-accessible
19      STATUS      current
20      DESCRIPTION
21          "This table contains packet classifier rules associated
22          with service flows."
23      REFERENCE
24          "Section 11.13.22.3.4 in IEEE 802.16REVd/D5-2004"
25     ::= { wmanIfBsPacketCs 3 }
26
27  wmanIfBsClassifierRuleEntry OBJECT-TYPE
28      SYNTAX      wmanIfBsClassifierRuleEntry
29      MAX-ACCESS  not-accessible
30      STATUS      current
31      DESCRIPTION
32          "This table provides one row for each packet classifier
33          rule, and is indexed by wmanIfBsSfId and
34          wmanIfBsClassifierRuleIndex. wmanIfBsSfIndex
35          identifies the service flow, while
36          wmanIfBsClassifierRuleIndex identifies the packet
37          classifier rule."
38      INDEX { wmanIfBsSfIndex, wmanIfBsClassifierRuleIndex }
39     ::= { wmanIfBsClassifierRuleTable 1 }
40
41  wmanIfBsClassifierRuleEntry ::= SEQUENCE {
42      wmanIfBsSfIndex          Unsigned32,
43      wmanIfBsClassifierRuleIndex  Unsigned32,
44      wmanIfBsClassifierRulePriority  INTEGER,
45      wmanIfBsClassifierRuleIpTosLow  OCTET STRING,
46      wmanIfBsClassifierRuleIpTosHigh  OCTET STRING,
47      wmanIfBsClassifierRuleIpTosMask  OCTET STRING,
48      wmanIfBsClassifierRuleIpProtocol  Integer32,
49      wmanIfBsClassifierRuleIpAddressType  InetAddressType,
50      wmanIfBsClassifierRuleIpSourceAddr  InetAddress,
51      wmanIfBsClassifierRuleIpSourceMask  InetAddress,
52      wmanIfBsClassifierRuleIpDestAddr  InetAddress,
53      wmanIfBsClassifierRuleIpDestMask  InetAddress,
54      wmanIfBsClassifierRuleSourcePortStart  Integer32,

```

```

1      wmanIfBsClassifierRuleSourcePortEnd      Integer32,
2      wmanIfBsClassifierRuleDestPortStart     Integer32,
3      wmanIfBsClassifierRuleDestPortEnd      Integer32,
4      wmanIfBsClassifierRuleDestMacAddr      MacAddress,
5      wmanIfBsClassifierRuleDestMacMask      MacAddress,
6      wmanIfBsClassifierRuleSourceMacAddr     MacAddress,
7      wmanIfBsClassifierRuleSourceMacMask    MacAddress,
8      wmanIfBsClassifierRuleEnetProtocolType  INTEGER,
9      wmanIfBsClassifierRuleEnetProtocol     Integer32,
10     wmanIfBsClassifierRuleUserPriLow       Integer32,
11     wmanIfBsClassifierRuleUserPriHigh      Integer32,
12     wmanIfBsClassifierRuleVlanId          Integer32,
13     wmanIfBsClassifierRuleState           INTEGER,
14     wmanIfBsClassifierRulePkts            Counter64,
15     wmanIfBsClassifierRuleRowStatus       RowStatus
16     }
17
18     wmanIfBsSfIndex OBJECT-TYPE
19         SYNTAX      Unsigned32 (1 .. 4294967295)
20         MAX-ACCESS  not-accessible
21         STATUS      current
22         DESCRIPTION
23             "A 32 bit quantity that uniquely identifies a service flow
24             to both the subscriber station and base station (BS)."
```

```

25         ::= { wmanIfBsClassifierRuleEntry 1 }
26
27     wmanIfBsClassifierRuleIndex OBJECT-TYPE
28         SYNTAX      Unsigned32 (1..4294967295)
29         MAX-ACCESS  not-accessible
30         STATUS      current
31         DESCRIPTION
32             "An index is assigned to a classifier in BS classifiers
33             table"
```

```

34         ::= { wmanIfBsClassifierRuleEntry 2 }
35
36     wmanIfBsClassifierRulePriority OBJECT-TYPE
37         SYNTAX      INTEGER (0..255)
38         MAX-ACCESS  read-create
39         STATUS      current
40         DESCRIPTION
41             "The value specifies the priority for the Classifier, which
42             is used for determining the order of the Classifier. A
43             higher value indicates higher priority. Classifiers may
44             have priorities in the range 0..255."
```

```

45         REFERENCE
46             "Section 11.13.19.3.4.1 in IEEE 802.16REVd/D4-2004"
```

```

47         DEFVAL      { 0 }
48         ::= { wmanIfBsClassifierRuleEntry 3 }
49
50     wmanIfBsClassifierRuleIpTosLow OBJECT-TYPE
51         SYNTAX      OCTET STRING (SIZE(1))
52         MAX-ACCESS  read-create
53         STATUS      current
54         DESCRIPTION
```

```

1           "The low value of a range of TOS byte values. If the
2           referenced parameter is not present in a classifier, this
3           object reports the value of 0."
4     REFERENCE
5           "Section 11.13.19.3.4.2 in IEEE 802.16REvd/D5-2004"
6     ::= { wmanIfBsClassifierRuleEntry 4 }
7
8     wmanIfBsClassifierRuleIpTosHigh OBJECT-TYPE
9       SYNTAX      OCTET STRING (SIZE(1))
10      MAX-ACCESS  read-create
11      STATUS      current
12      DESCRIPTION
13        "The 8-bit high value of a range of TOS byte values.
14         If the referenced parameter is not present in a classifier,
15         this object reports the value of 0."
16      REFERENCE
17        "Section 11.13.19.3.4.2 in IEEE 802.16REvd/D5-2004"
18      ::= { wmanIfBsClassifierRuleEntry 5 }
19
20     wmanIfBsClassifierRuleIpTosMask OBJECT-TYPE
21       SYNTAX      OCTET STRING (SIZE(1))
22      MAX-ACCESS  read-create
23      STATUS      current
24      DESCRIPTION
25        "The mask value is bitwise ANDed with TOS byte in an IP
26         packet and this value is used check range checking of
27         TosLow and TosHigh. If the referenced parameter is not
28         present in a classifier, this object reports the value
29         of 0."
30      REFERENCE
31        "Section 11.13.19.3.4.2 in IEEE 802.16REvd/D5-2004"
32      ::= { wmanIfBsClassifierRuleEntry 6 }
33
34     wmanIfBsClassifierRuleIpProtocol OBJECT-TYPE
35       SYNTAX      Integer32 (0..255)
36      MAX-ACCESS  read-create
37      STATUS      current
38      DESCRIPTION
39        "This object indicates the value of the IP Protocol field
40         required for IP packets to match this rule. If the
41         referenced parameter is not present in a classifier, this
42         object reports the value of 0."
43      REFERENCE
44        "Section 11.13.19.3.4.3 in IEEE 802.16REvd/D5-2004"
45      ::= { wmanIfBsClassifierRuleEntry 7 }
46
47     wmanIfBsClassifierRuleIpAddressType OBJECT-TYPE
48       SYNTAX      InetAddressType
49      MAX-ACCESS  read-create
50      STATUS      current
51      DESCRIPTION
52        "The type of the internet address for
53         wmanIfBsClassifierRuleIpSourceAddr,
54         wmanIfBsClassifierRuleIpSourceMask,

```

```

1           wmanIfBsClassifierRuleIpDestAddr, and
2           wmanIfBsClassifierRuleIpDestMask.
3           If the referenced parameter is not present in a classifier,
4           this object reports the value of ipv4(1)."
5     REFERENCE
6       "Section 11.13.19.3.4.4 in IEEE 802.16REVd/D5-2004"
7     ::= { wmanIfBsClassifierRuleEntry 8 }
8
9     wmanIfBsClassifierRuleIpSourceAddr OBJECT-TYPE
10      SYNTAX      InetAddress
11      MAX-ACCESS  read-create
12      STATUS      current
13      DESCRIPTION
14        "This object specifies the value of the IP Source Address
15         required for packets to match this rule. An IP packet
16         matches the rule when the packet ip source address bitwise
17         ANDED with the wmanIfBsClassifierRuleIpSourceMask value
18         equals the wmanIfBsClassifierRuleIpSourceAddr value.
19         If the referenced parameter is not present n a classifier,
20         this object reports the value of 0.0.0.0."
21      REFERENCE
22        "Section 11.13.19.3.4.4 in IEEE 802.16REVd/D5-2004"
23      ::= { wmanIfBsClassifierRuleEntry 9 }
24
25     wmanIfBsClassifierRuleIpSourceMask OBJECT-TYPE
26      SYNTAX      InetAddress
27      MAX-ACCESS  read-create
28      STATUS      current
29      DESCRIPTION
30        "This object specifies which bits of a packet's IP Source
31         Address that are compared to match this rule. An IP packet
32         matches the rule when the packet source address bitwise
33         ANDED with the
34         wmanIfBsClassifierRuleIpSourceMask value equals the
35         wmanIfBsClassifierRuleIpSourceAddr value.
36         If the referenced parameter is not present in a classifier,
37         this object reports the value of 0.0.0.0."
38      REFERENCE
39        "Section 11.13.19.3.4.4 in IEEE 802.16REVd/D5-2004"
40      ::= { wmanIfBsClassifierRuleEntry 10 }
41
42     wmanIfBsClassifierRuleIpDestAddr OBJECT-TYPE
43      SYNTAX      InetAddress
44      MAX-ACCESS  read-create
45      STATUS      current
46      DESCRIPTION
47        "This object specifies the value of the IP Destination
48         Address required for packets to match this rule. An IP
49         packet matches the rule when the packet IP destination
50         address bitwise ANDED with the
51         wmanIfBsClassifierRuleIpDestMask value equals the
52         wmanIfBsClassifierRuleIpDestAddr value.
53         If the referenced parameter is not present in a
54         classifier, this object reports the value of 0.0.0.0."

```

```

1      REFERENCE
2      "Section 11.13.19.3.4.5 in IEEE 802.16REvd/D5-2004"
3      ::= { wmanIfBsClassifierRuleEntry 11 }
4
5      wmanIfBsClassifierRuleIpDestMask OBJECT-TYPE
6          SYNTAX      InetAddress
7          MAX-ACCESS  read-create
8          STATUS      current
9          DESCRIPTION
10             "This object specifies which bits of a packet's IP
11              Destination Address that are compared to match this rule.
12              An IP packet matches the rule when the packet destination
13              address bitwise ANDed with the
14              wmanIfBsClassifierRuleIpDestMask value equals the
15              wmanIfBsClassifierRuleIpDestAddr value.
16              If the referenced parameter is not present in a classifier
17              , this object reports the value of 0.0.0.0."
18          REFERENCE
19             "Section 11.13.19.3.4.5 in IEEE 802.16REvd/D5-2004"
20             ::= { wmanIfBsClassifierRuleEntry 12 }
21
22      wmanIfBsClassifierRuleSourcePortStart OBJECT-TYPE
23          SYNTAX      Integer32 (0..65535)
24          MAX-ACCESS  read-create
25          STATUS      current
26          DESCRIPTION
27             "This object specifies the low end inclusive range of
28              TCP/UDP source port numbers to which a packet is compared.
29              This object is irrelevant for non-TCP/UDP IP packets.
30              If the referenced parameter is not present in a
31              classifier, this object reports the value of 0."
32          REFERENCE
33             "Section 11.13.19.3.4.6 in IEEE 802.16REvd/D5-2004"
34             ::= { wmanIfBsClassifierRuleEntry 13 }
35
36      wmanIfBsClassifierRuleSourcePortEnd OBJECT-TYPE
37          SYNTAX      Integer32 (0..65535)
38          MAX-ACCESS  read-create
39          STATUS      current
40          DESCRIPTION
41             "This object specifies the high end inclusive range of
42              TCP/UDP source port numbers to which a packet is compared.
43              This object is irrelevant for non-TCP/UDP IP packets.
44              If the referenced parameter is not present in a classifier,
45              this object reports the value of 65535."
46          REFERENCE
47             "Section 11.13.19.3.4.6 in IEEE 802.16REvd/D5-2004"
48             ::= { wmanIfBsClassifierRuleEntry 14 }
49
50      wmanIfBsClassifierRuleDestPortStart OBJECT-TYPE
51          SYNTAX      Integer32 (0..65535)
52          MAX-ACCESS  read-create
53          STATUS      current
54          DESCRIPTION

```

```
1           "This object specifies the low end inclusive range of
2           TCP/UDP destination port numbers to which a packet is
3           compared. If the referenced parameter is not present
4           in a classifier, this object reports the value of 0."
5     REFERENCE
6           "Section 11.13.19.3.4.7 in IEEE 802.16REvd/D5-2004"
7     ::= { wmanIfBsClassifierRuleEntry 15 }
8
9     wmanIfBsClassifierRuleDestPortEnd OBJECT-TYPE
10      SYNTAX      Integer32 (0..65535)
11      MAX-ACCESS  read-create
12      STATUS      current
13      DESCRIPTION
14          "This object specifies the high end inclusive range of
15          TCP/UDP destination port numbers to which a packet is
16          compared. If the referenced parameter is not present
17          in a classifier, this object reports the value of
18          65535."
19      REFERENCE
20          "Section 11.13.19.3.4.7 in IEEE 802.16REvd/D5-2004"
21      ::= { wmanIfBsClassifierRuleEntry 16 }
22
23     wmanIfBsClassifierRuleDestMacAddr OBJECT-TYPE
24      SYNTAX      MacAddress
25      MAX-ACCESS  read-create
26      STATUS      current
27      DESCRIPTION
28          "An Ethernet packet matches an entry when its destination
29          MAC address bitwise ANDed with
30          wmanIfBsClassifierRuleDestMacMask equals the value of
31          wmanIfBsClassifierRuleDestMacAddr. If the referenced
32          parameter is not present in a classifier, this object
33          reports the value of '000000000000'H."
34      REFERENCE
35          "Section 11.13.19.3.4.8 in IEEE 802.16REvd/D5-2004"
36      ::= { wmanIfBsClassifierRuleEntry 17 }
37
38     wmanIfBsClassifierRuleDestMacMask OBJECT-TYPE
39      SYNTAX      MacAddress
40      MAX-ACCESS  read-create
41      STATUS      current
42      DESCRIPTION
43          "An Ethernet packet matches an entry when its destination
44          MAC address bitwise ANDed with
45          wmanIfBsClassifierRuleDestMacMask equals the value of
46          wmanIfBsClassifierRuleDestMacAddr. If the referenced
47          parameter is not present in a classifier, this object
48          reports the value of '000000000000'H."
49      REFERENCE
50          "Section 11.13.19.3.4.8 in IEEE 802.16REvd/D5-2004"
51      ::= { wmanIfBsClassifierRuleEntry 18 }
52
53     wmanIfBsClassifierRuleSourceMacAddr OBJECT-TYPE
54      SYNTAX      MacAddress
```

```

1      MAX-ACCESS  read-create
2      STATUS      current
3      DESCRIPTION
4          "An Ethernet packet matches this entry when its source
5          MAC address bitwise ANDed with
6          wmanIfBsClassifierRuleSourceMacMask equals the value
7          of wmanIfBsClassifierRuleSourceMacAddr. If the
8          referenced parameter is not present in a classifier,
9          this object reports the value of '00000000000'H."
10     REFERENCE
11         "Section 11.13.19.3.4.9 in IEEE 802.16REvd/D5-2004"
12     ::= { wmanIfBsClassifierRuleEntry 19 }
13
14     wmanIfBsClassifierRuleSourceMacMask OBJECT-TYPE
15         SYNTAX      MacAddress
16         MAX-ACCESS  read-create
17         STATUS      current
18         DESCRIPTION
19             "An Ethernet packet matches an entry when its destination
20             MAC address bitwise ANDed with
21             wmanIfBsClassifierRuleSourceMacMask equals the value of
22             wmanIfBsClassifierRuleSourceMacAddr. If the referenced
23             parameter is not present in a classifier, this object
24             reports the value of '00000000000'H."
25         REFERENCE
26             "Section 11.13.19.3.4.9 in IEEE 802.16REvd/D5-2004"
27         ::= { wmanIfBsClassifierRuleEntry 20 }
28
29     wmanIfBsClassifierRuleEnetProtocolType OBJECT-TYPE
30         SYNTAX      INTEGER {none(0),
31                             ethertype(1),
32                             dsap(2)}
33         MAX-ACCESS  read-create
34         STATUS      current
35         DESCRIPTION
36             "This object indicates the format of the layer 3 protocol
37             id in the Ethernet packet. A value of none(0) means that
38             the rule does not use the layer 3 protocol type as a
39             matching criteria. A value of ethertype(1) means that the
40             rule applies only to frames which contains an EtherType
41             value. Ethertype values are contained in packets using
42             the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
43             Sub-Network Access Protocol (SNAP) encapsulation formats.
44             A value of dsap(2) means that the rule applies only to
45             frames using the IEEE802.3 encapsulation format with a
46             Destination Service Access Point (DSAP) other than 0xAA
47             (which is reserved for SNAP). If the Ethernet frame
48             contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
49             this object applies to the embedded EtherType field within
50             the 802.1P/Q header. If the referenced parameter is not
51             present in a classifier, this object reports the value of
52             0."
53         REFERENCE
54             "Section 11.13.19.3.4.10 in IEEE 802.16REvd/D5-2004"

```

```

1      ::= { wmanIfBsClassifierRuleEntry 21 }
2
3  wmanIfBsClassifierRuleEnetProtocol OBJECT-TYPE
4      SYNTAX      Integer32 (0..65535)
5      MAX-ACCESS  read-create
6      STATUS      current
7      DESCRIPTION
8          "If wmanIfBsClassifierRuleEnetProtocolType is none(0),
9           this object is ignored when considering whether a packet
10          matches the current rule.
11          If wmanIfBsClassifierRuleEnetProtocolType is ethertype(1),
12          this object gives the 16-bit value of the EtherType that
13          the packet must match in order to match the rule.
14          If wmanIfBsClassifierRuleEnetProtocolType is dsap(2), the
15          lower 8 bits of this object's value must match the DSAP
16          byte of the packet in order to match the rule.
17          If the Ethernet frame contains an 802.1P/Q Tag header
18          (i.e. EtherType 0x8100), this object applies to the
19          embedded EtherType field within the 802.1P/Q header.
20          If the referenced parameter is not present in the
21          classifier, the value of this object is reported as 0."
22      REFERENCE
23          "Section 11.13.19.3.4.10 in IEEE 802.16REVd/D5-2004"
24      ::= { wmanIfBsClassifierRuleEntry 22 }
25
26  wmanIfBsClassifierRuleUserPriLow OBJECT-TYPE
27      SYNTAX      Integer32 (0..7)
28      MAX-ACCESS  read-create
29      STATUS      current
30      DESCRIPTION
31          "This object applies only to Ethernet frames using the
32          802.1P/Q tag header (indicated with EtherType 0x8100).
33          Such frames include a 16-bit Tag that contains a 3 bit
34          Priority field and a 12 bit VLAN number.
35          Tagged Ethernet packets must have a 3-bit Priority field
36          within the range of wmanIfBsClassifierRulePriLow and
37          wmanIfBsClassifierRulePriHigh in order to match this
38          rule.
39          If the referenced parameter is not present in the
40          classifier, the value of this object is reported as 0."
41      REFERENCE
42          "Section 11.13.19.3.4.11 in IEEE 802.16REVd/D5-2004"
43      ::= { wmanIfBsClassifierRuleEntry 23 }
44
45  wmanIfBsClassifierRuleUserPriHigh OBJECT-TYPE
46      SYNTAX      Integer32 (0..7)
47      MAX-ACCESS  read-create
48      STATUS      current
49      DESCRIPTION
50          "This object applies only to Ethernet frames using the
51          802.1P/Q tag header (indicated with EtherType 0x8100).
52          Such frames include a 16-bit Tag that contains a 3 bit
53          Priority field and a 12 bit VLAN number.
54          Tagged Ethernet packets must have a 3-bit Priority

```



```

1         field within the range of wmanIfBsClassifierRulePriLow
2         and wmanIfBsClassifierRulePriHigh in order to match
3         this rule.
4         If the referenced parameter is not present in the
5         classifier, the value of this object is reported as 7."
6     REFERENCE
7         "Section 11.13.19.3.4.11 in IEEE 802.16REvd/D5-2004"
8     ::= { wmanIfBsClassifierRuleEntry 24 }
9
10    wmanIfBsClassifierRuleVlanId OBJECT-TYPE
11        SYNTAX      Integer32 (0..4095)
12        MAX-ACCESS  read-create
13        STATUS      current
14        DESCRIPTION
15            "This object applies only to Ethernet frames using the
16            802.1P/Q tag header.
17            If this object's value is nonzero, tagged packets must
18            have a VLAN Identifier that matches the value in order
19            to match the rule.
20            Only the least significant 12 bits of this object's
21            value are valid.
22            If the referenced parameter is not present in the
23            classifier, the value of this object is reported as 0."
24        REFERENCE
25            "Section 11.13.19.3.4.12 in IEEE 802.16REvd/D5-2004"
26        ::= { wmanIfBsClassifierRuleEntry 25 }
27
28    wmanIfBsClassifierRuleState OBJECT-TYPE
29        SYNTAX      INTEGER {active(1),
30                        inactive(2)}
31        MAX-ACCESS  read-create
32        STATUS      current
33        DESCRIPTION
34            "This object indicates whether or not the classifier is
35            enabled to classify packets to a Service Flow.
36            If the referenced parameter is not present in the
37            classifier, the value of this object is reported
38            as active(1)."
39        REFERENCE
40            "Section 11.13.19.3.4.1 in IEEE 802.16REvd/D5-2004"
41        ::= { wmanIfBsClassifierRuleEntry 26 }
42
43    wmanIfBsClassifierRulePkts OBJECT-TYPE
44        SYNTAX      Counter64
45        MAX-ACCESS  read-create
46        STATUS      current
47        DESCRIPTION
48            "This object counts the number of packets that have
49            been classified using this entry."
50        REFERENCE
51            "Section 11.13.19.3.4.1 in IEEE 802.16REvd/D5-2004"
52        ::= { wmanIfBsClassifierRuleEntry 27 }
53
54    wmanIfBsClassifierRuleRowStatus OBJECT-TYPE

```

```

1      SYNTAX      RowStatus
2      MAX-ACCESS  read-create
3      STATUS      current
4      DESCRIPTION
5          "This object is used to create a new row or modify or
6          delete an existing row in this table.
7
8          If the implementator of this MIB has chosen not
9          to implement 'dynamic assignment' of profiles, this
10         object is not useful and should return noSuchName
11         upon SNMP request."
12     ::= { wmanIfBsClassifierRuleEntry 28 }
13
14     wmanIfBsSsPacketCounterTable OBJECT-TYPE
15         SYNTAX      SEQUENCE OF wmanIfBsSsPacketCounterEntry
16         MAX-ACCESS  not-accessible
17         STATUS      current
18         DESCRIPTION
19             "This table contains counters to keep track of the number
20             of packets or octets that have been received or
21             transmitted on the per service flow basis."
22     ::= { wmanIfBsPacketCs 4 }
23
24     wmanIfBsSsPacketCounterEntry OBJECT-TYPE
25         SYNTAX      wmanIfBsSsPacketCounterEntry
26         MAX-ACCESS  not-accessible
27         STATUS      current
28         DESCRIPTION
29             "This table provides one row for each service flow, and
30             is indexed by wmanIfBsSsSfIndex and
31             wmanIfBsSsMacAddress."
32     INDEX { wmanIfBsSsSfIndex, wmanIfBsSsMacAddr }
33     ::= { wmanIfBsSsPacketCounterTable 1 }
34
35     wmanIfBsSsPacketCounterEntry ::= SEQUENCE {
36         wmanIfBsSsSfIndex      Unsigned32,
37         wmanIfBsSsMacAddr      MacAddress,
38         wmanIfBsSsSfDirection  INTEGER,
39         wmanIfBsSsMacSduCount  Counter64,
40         wmanIfBsSsOctetCount   Counter64,
41         wmanIfBsSsResetCounter  INTEGER,
42         wmanIfBsSsResetCounterTime  TimeStamp
43     }
44
45     wmanIfBsSsSfIndex OBJECT-TYPE
46         SYNTAX      Unsigned32 (1 .. 4294967295)
47         MAX-ACCESS  not-accessible
48         STATUS      current
49         DESCRIPTION
50             "A 32 bit quantity that uniquely identifies a service flow."
51     ::= { wmanIfBsSsPacketCounterEntry 1 }
52
53     wmanIfBsSsMacAddr OBJECT-TYPE
54         SYNTAX      MacAddress

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The MAC address of the SS, where the service flow resides.
5           It can be used as the index to associate service flows
6           with the SS."
7      ::= { wmanIfBsSsPacketCounterEntry 2 }
8
9      wmanIfBsSsSfdirection OBJECT-TYPE
10     SYNTAX      INTEGER {transmit(1),
11                  receive(2)}
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "An attribute indicating whether the packet counter is on
16         transmit or receive direction from the BS perspective."
17     ::= { wmanIfBsSsPacketCounterEntry 3 }
18
19     wmanIfBsSsMacSduCount OBJECT-TYPE
20     SYNTAX      Counter64
21     MAX-ACCESS  read-only
22     STATUS      current
23     DESCRIPTION
24         "This object counts the number of MAC SDUs that have
25         been transmitted or received."
26     ::= { wmanIfBsSsPacketCounterEntry 4 }
27
28     wmanIfBsSsOctetCount OBJECT-TYPE
29     SYNTAX      Counter64
30     MAX-ACCESS  read-only
31     STATUS      current
32     DESCRIPTION
33         "This object counts the number of octets that have
34         been transmitted or received."
35     ::= { wmanIfBsSsPacketCounterEntry 5 }
36
37     wmanIfBsSsResetCounter OBJECT-TYPE
38     SYNTAX      INTEGER {null(0),
39                  resetCounter(1)}
40     MAX-ACCESS  read-write
41     STATUS      current
42     DESCRIPTION
43         "When SET this attribute to resetCounter(1), the
44         corresponding entry of packet counters will be reset.
45         A GET operation performed on this object will always
46         return null(0). The counter is normally reset after
47         the packet count information is retrieved. "
48     ::= { wmanIfBsSsPacketCounterEntry 6 }
49
50     wmanIfBsSsResetCounterTime OBJECT-TYPE
51     SYNTAX      TimeStamp
52     MAX-ACCESS  read-create
53     STATUS      current
54     DESCRIPTION

```

```

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

```

```

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

```

```

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

```

```

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

```

```

1      DESCRIPTION
2          "This object is used to create a new row or modify or
3          delete an existing row in this table.
4
5          If the implementator of this MIB has chosen not
6          to implement 'dynamic assignment' of profiles, this
7          object is not useful and should return noSuchName
8          upon SNMP request."
9      ::= { wmanIfBsConfigurationEntry 17 }
10
11     --
12     -- Base Station statistics counters
13     --
14     wmanIfBsStatisticCounter OBJECT IDENTIFIER ::= { wmanIfBsCps 2 }
15
16     wmanIfBsChMeasurementTable OBJECT-TYPE
17         SYNTAX      SEQUENCE OF      WmanIfBsChMeasurementEntry
18         MAX-ACCESS  not-accessible
19         STATUS      current
20         DESCRIPTION
21             "This table contains channel measurement information
22             on the uplink signal received from SS. The table shall
23             be maintained as FIFO to store measurement samples that
24             can be used to create RSSI and CINR histogram report.
25             When the measurement entry for a SS reaches the limit,
26             the oldest entry shall be deleted as the new entry is
27             added to the table."
28         ::= { wmanIfBsStatisticCounter 1 }
29
30     wmanIfBsChMeasurementEntry OBJECT-TYPE
31         SYNTAX      WmanIfBsChMeasurementEntry
32         MAX-ACCESS  not-accessible
33         STATUS      current
34         DESCRIPTION
35             "Each entry in the table contains RSSI and CINR
36             signal quality measurement on signal received from the SS.
37             The primary index is the ifIndex with ifType of propBWA2Mp
38             identifying the BS sector. wmanIfChSsidIndex identifies
39             the SS from which the signal was received.
40             wmanIfBsHistogramIndex is the index to histogram samples.
41             Since there is no time stamp in the table,
42             wmanIfBsHistogramIndex should be increased monotonically,
43             and warps around when it reaches the limit. "
44         INDEX      { ifIndex, wmanIfBsChSsidIndex,
45                     wmanIfBsHistogramIndex }
46         ::= { wmanIfBsChMeasurementTable 1 }
47
48     wmanIfBsChMeasurementEntry ::= SEQUENCE {
49         wmanIfBsChSsidIndex      Unsigned32,
50         wmanIfBsHistogramIndex   Unsigned32,
51         wmanIfBsChannelNumber    INTEGER,
52         wmanIfBsStartFrame       INTEGER,
53         wmanIfBsDuration          INTEGER,
54         wmanIfBsBasicReport       BITS,

```



```

1          wmanIfBsMeanCinrReport          INTEGER,
2          wmanIfBsMeanRssiReport          INTEGER,
3          wmanIfBsStdDeviationCinrReport  INTEGER,
4          wmanIfBsStdDeviationRssiReport  INTEGER}
5
6  wmanIfBsChSsIdIndex OBJECT-TYPE
7      SYNTAX      Unsigned32 (1 .. 4294967295)
8      MAX-ACCESS  read-only
9      STATUS      current
10     DESCRIPTION
11         "wmanIfBsChIdIndex identifies the SS providing the
12         channel measurement."
13     REFERENCE
14         "Section 6.4.2.3.5 in IEEE 802.16REvD/D5-2004"
15     ::= { wmanIfBsChMeasurementEntry 1 }
16
17  wmanIfBsHistogramIndex OBJECT-TYPE
18     SYNTAX      Unsigned32 (1 .. 4294967295)
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "wmanIfBsHistogramIndex identifies the histogram samples
23         in the table for each subscriber station."
24     ::= { wmanIfBsChMeasurementEntry 2 }
25
26  wmanIfBsChannelNumber OBJECT-TYPE
27     SYNTAX      INTEGER
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31         "Physical channel number to be reported on is only
32         applicable to licence exampt band. For licenced band,
33         this parameter should be null."
34     REFERENCE
35         "Section 8.5.1 in IEEE 802.16REvD/D5-2004"
36     ::= { wmanIfBsChMeasurementEntry 3 }
37
38  wmanIfBsStartFrame OBJECT-TYPE
39     SYNTAX      INTEGER
40     MAX-ACCESS  read-only
41     STATUS      current
42     DESCRIPTION
43         "Frame number in which measurement for this channel
44         started."
45     REFERENCE
46         "Section 11.12 in IEEE 802.16REvD/D5-2004"
47     ::= { wmanIfBsChMeasurementEntry 4 }
48
49  wmanIfBsDuration OBJECT-TYPE
50     SYNTAX      INTEGER
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54         "Cumulative measurement duration on the channel in

```

```

1         multiples of Ts. For any value exceeding 0xFFFFFFFF,
2         report 0xFFFFFFFF."
3     REFERENCE
4         "Section 11.12 in IEEE 802.16REvD/D5-2004"
5     ::= { wmanIfBsChMeasurementEntry 5 }
6
7     wmanIfBsBasicReport OBJECT-TYPE
8         SYNTAX      BITS {wirelessHuman(0),
9                     unknownTransmission(1),
10                    primaryUser(2),
11                    channegNotMeasured(3)}
12         MAX-ACCESS  read-only
13         STATUS      current
14         DESCRIPTION
15             "Bit #0: wirelessHUMAN detected on the channel
16             Bit #1: Unknown transmissions detected on the channel
17             Bit #2: Primary User detected on the channel
18             Bit #3: Unmeasured. Channel not measured"
19         REFERENCE
20             "Section 11.12 in IEEE 802.16REvD/D5-2004"
21     ::= { wmanIfBsChMeasurementEntry 6 }
22
23     wmanIfBsMeanCinrReport OBJECT-TYPE
24         SYNTAX      INTEGER
25         MAX-ACCESS  read-only
26         STATUS      current
27         DESCRIPTION
28             "Mean CINR report."
29         REFERENCE
30             "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
31             802.16REvD/D5-2004"
32     ::= { wmanIfBsChMeasurementEntry 7 }
33
34     wmanIfBsMeanRssiReport OBJECT-TYPE
35         SYNTAX      INTEGER
36         MAX-ACCESS  read-only
37         STATUS      current
38         DESCRIPTION
39             "Mean RSSI report."
40         REFERENCE
41             "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
42             802.16REvD/D5-2004"
43     ::= { wmanIfBsChMeasurementEntry 8 }
44
45     wmanIfBsStdDeviationCinrReport OBJECT-TYPE
46         SYNTAX      INTEGER
47         MAX-ACCESS  read-only
48         STATUS      current
49         DESCRIPTION
50             "Standard deviation CINR report."
51         REFERENCE
52             "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
53             802.16REvD/D5-2004"
54     ::= { wmanIfBsChMeasurementEntry 9 }

```

```

1
2  wmanIfBsStdDeviationRssiReport OBJECT-TYPE
3      SYNTAX      INTEGER
4      MAX-ACCESS  read-only
5      STATUS      current
6      DESCRIPTION
7          "Standard deviation RSSI report."
8      REFERENCE
9          "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
10         802.16REVD/D5-2004"
11     ::= { wmanIfBsChMeasurementEntry 10 }
12
13 --
14 -- Base station PKM group
15 -- wmanIfBsPkmObjects contain the Base Station Privacy Sublayer objects
16 wmanIfBsPkmObjects OBJECT IDENTIFIER ::= { wmanIfBsObjects 4 }
17
18 --
19 -- Table wmanIfBsPkmBaseTable
20 --
21 wmanIfBsPkmBaseTable OBJECT-TYPE
22     SYNTAX      SEQUENCE OF wmanIfBsPkmBaseEntry
23     MAX-ACCESS  not-accessible
24     STATUS      current
25     DESCRIPTION
26         "This table describes the basic PKM attributes of each Base
27         Station wireless interface."
28     ::= { wmanIfBsPkmObjects 1 }
29
30 wmanIfBsPkmBaseEntry OBJECT-TYPE
31     SYNTAX      wmanIfBsPkmBaseEntry
32     MAX-ACCESS  not-accessible
33     STATUS      current
34     DESCRIPTION
35         "Each entry contains objects describing attributes of one
36         BS wireless interface."
37     INDEX      { ifIndex }
38     ::= { wmanIfBsPkmBaseTable 1 }
39
40 wmanIfBsPkmBaseEntry ::= SEQUENCE {
41     wmanIfBsPkmDefaultAuthLifetime      Integer32,
42     wmanIfBsPkmDefaultTEKLifetime       Integer32,
43     wmanIfBsPkmDefaultSelfSigManufCertTrust INTEGER,
44     wmanIfBsPkmCheckCertValidityPeriods TruthValue,
45     wmanIfBsPkmAuthentInfos              Counter32,
46     wmanIfBsPkmAuthRequests              Counter32,
47     wmanIfBsPkmAuthReplies               Counter32,
48     wmanIfBsPkmAuthRejects               Counter32,
49     wmanIfBsPkmAuthInvalids              Counter32
50 }
51
52 wmanIfBsPkmDefaultAuthLifetime OBJECT-TYPE
53     SYNTAX      Integer32 (86400..6048000)
54     UNITS      "seconds"

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the default lifetime, in
5          seconds, the BS assigns to a new authorization key."
6      REFERENCE
7          "Table 341 in IEEE 802.16REvd/D5-2004"
8      DEFVAL      { 604800 }
9      ::= { wmanIfBsPkmBaseEntry 1 }
10
11  wmanIfBsPkmDefaultTEKLifetime OBJECT-TYPE
12      SYNTAX      Integer32 (1800..604800)
13      UNITS       "seconds"
14      MAX-ACCESS  read-write
15      STATUS      current
16      DESCRIPTION
17          "The value of this object is the default lifetime, in
18          seconds, the BS assigns to a new Traffic Encryption
19          Key(TEK)."

```

```

1      DESCRIPTION
2          "The value of this object is the count of times the BS has
3          received an Authentication Information message from any
4          SS."
5      ::= { wmanIfBsPkmBaseEntry 5 }
6
7      wmanIfBsPkmAuthRequests 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             received an Authorization Request message from any SS"
14         ::= { wmanIfBsPkmBaseEntry 6 }
15
16         wmanIfBsPkmAuthReplies 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 Reply message to any SS."
23             ::= { wmanIfBsPkmBaseEntry 7 }
24
25         wmanIfBsPkmAuthRejects 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 Reject message to any SS."
32             ::= { wmanIfBsPkmBaseEntry 8 }
33
34         wmanIfBsPkmAuthInvalids OBJECT-TYPE
35             SYNTAX      Counter32
36             MAX-ACCESS  read-only
37             STATUS      current
38             DESCRIPTION
39                 "The value of this object is the count of times the BS has
40                 transmitted an Authorization Invalid message to any SS."
41             ::= { wmanIfBsPkmBaseEntry 9 }
42
43         --
44         -- Table wmanIfBsPkmAuthTable
45         --
46         wmanIfBsPkmAuthTable OBJECT-TYPE
47             SYNTAX      SEQUENCE OF wmanIfBsPkmAuthEntry
48             MAX-ACCESS  not-accessible
49             STATUS      current
50             DESCRIPTION
51                 "This table describes the attributes of each SS
52                 authorization association. The BS maintains one
53                 authorization association with each Baseline
54                 Privacy-enabled SS on each BS wireless interface."

```

```

1      ::= { wmanIfBsPkmObjects 2 }
2
3  wmanIfBsPkmAuthEntry OBJECT-TYPE
4      SYNTAX      WmanIfBsPkmAuthEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8          "Each entry contains objects describing attributes of one
9          authorization association. The BS MUST create one entry per
10         SS per wireless interface, based on the receipt of an
11         Authorization Request message, and MUST not delete the
12         entry before the SS authorization permanently expires."
13     INDEX      { ifIndex, wmanIfBsPkmAuthSsMacAddress }
14     ::= { wmanIfBsPkmAuthTable 1 }
15
16  wmanIfBsPkmAuthEntry ::= SEQUENCE {
17      wmanIfBsPkmAuthSsMacAddress      MacAddress,
18      wmanIfBsPkmAuthSsPublicKey      OCTET STRING,
19      wmanIfBsPkmAuthSsKeySequenceNumber Integer32,
20      wmanIfBsPkmAuthSsExpiresOld     DateAndTime,
21      wmanIfBsPkmAuthSsExpiresNew    DateAndTime,
22      wmanIfBsPkmAuthSsLifetime      Integer32,
23      wmanIfBsPkmAuthSsReset         INTEGER,
24      wmanIfBsPkmAuthSsInfos         Counter64,
25      wmanIfBsPkmAuthSsRequests      Counter64,
26      wmanIfBsPkmAuthSsReplies       Counter64,
27      wmanIfBsPkmAuthSsRejects       Counter64,
28      wmanIfBsPkmAuthSsInvalids      Counter64,
29      wmanIfBsPkmAuthRejectErrorCode INTEGER,
30      wmanIfBsPkmAuthRejectErrorString SnmpAdminString,
31      wmanIfBsPkmAuthInvalidErrorCode INTEGER,
32      wmanIfBsPkmAuthInvalidErrorString SnmpAdminString,
33      wmanIfBsPkmAuthPrimarySAId     Integer32,
34      wmanIfBsPkmAuthBpkmSsCertValid INTEGER,
35      wmanIfBsPkmAuthBpkmSsCert      OCTET STRING
36  }
37
38  wmanIfBsPkmAuthSsMacAddress OBJECT-TYPE
39      SYNTAX      MacAddress
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "The value of this object is the physical address of the SS
44          to which the authorization association applies."
45      ::= { wmanIfBsPkmAuthEntry 1 }
46
47  wmanIfBsPkmAuthSsPublicKey OBJECT-TYPE
48      SYNTAX      OCTET STRING (SIZE (140))
49      MAX-ACCESS  read-only
50      STATUS      current
51      DESCRIPTION
52          "The value of this object is a DER-encoded RSAPublicKey
53          ASN.1 type string, as defined in the RSA Encryption
54          Standard (PKCS #1) [10], corresponding to the public key of

```

```

1         the SS. The 74, 106, 140, 204, and 270 byte key encoding
2         lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
3         and 2048 public moduli respectively. This is a zero-length
4         string if the BS does not retain the public key."
5     ::= { wmanIfBsPkmAuthEntry 2 }
6
7     wmanIfBsPkmAuthSsKeySequenceNumber OBJECT-TYPE
8         SYNTAX      Integer32 (0..15)
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            "The value of this object is the most recent authorization
13            key sequence number for this SS."
14        ::= { wmanIfBsPkmAuthEntry 3 }
15
16        wmanIfBsPkmAuthSsExpiresOld OBJECT-TYPE
17            SYNTAX      DateAndTime
18            MAX-ACCESS  read-only
19            STATUS      current
20            DESCRIPTION
21                "The value of this object is the actual clock time for
22                expiration of the immediate predecessor of the most recent
23                authorization key for this FSM. If this FSM has only one
24                authorization key, then the value is the time of activation
25                of this FSM."
26            ::= { wmanIfBsPkmAuthEntry 4 }
27
28        wmanIfBsPkmAuthSsExpiresNew OBJECT-TYPE
29            SYNTAX      DateAndTime
30            MAX-ACCESS  read-only
31            STATUS      current
32            DESCRIPTION
33                "The value of this object is the actual clock time for
34                expiration of the most recent authorization key for this
35                FSM"
36            ::= { wmanIfBsPkmAuthEntry 5 }
37
38        wmanIfBsPkmAuthSsLifetime OBJECT-TYPE
39            SYNTAX      Integer32 (86400..604800)
40            UNITS      "seconds"
41            MAX-ACCESS  read-write
42            STATUS      current
43            DESCRIPTION
44                "The vaue of this object is the lifetime, in seconds, the
45                BS assigns to an authorization key for this SS."
46            REFERENCE
47                "Table 341 in IEEE 802.16REvd/D5-2004"
48            DEFVAL      { 604800 }
49            ::= { wmanIfBsPkmAuthEntry 6 }
50
51        wmanIfBsPkmAuthSsReset OBJECT-TYPE
52            SYNTAX      INTEGER {noResetRequested(1),
53                        invalidateAuth(2),
54                        sendAuthInvalid(3),

```

```

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

```



```

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

```

```

1         "IEEE 802.16 standard; Table 371"
2     ::= { wmanIfBsPkmAuthEntry 15 }
3
4     wmanIfBsPkmAuthInvalidErrorString OBJECT-TYPE
5         SYNTAX      SnmpAdminString (SIZE (0..128))
6         MAX-ACCESS  read-only
7         STATUS      current
8         DESCRIPTION
9             "The value of this object is the Display-String in most
10            recent Authorization Invalid message transmitted to the SS.
11            This is a zero length string if no Authorization Invalid
12            message has been transmitted to the SS."
13     ::= { wmanIfBsPkmAuthEntry 16 }
14
15     wmanIfBsPkmAuthPrimarySAId OBJECT-TYPE
16         SYNTAX      Integer32 (0..65536)
17         MAX-ACCESS  read-only
18         STATUS      current
19         DESCRIPTION
20             "The value of this object is the Primary Security
21            Association identifier."
22         REFERENCE
23             "IEEE 802.16 standard; 11.9.7"
24     ::= { wmanIfBsPkmAuthEntry 17 }
25
26     wmanIfBsPkmAuthBpkmSsCertValid OBJECT-TYPE
27         SYNTAX      INTEGER {unknown (0),
28                             validSsChained (1),
29                             validSsTrusted (2),
30                             invalidSsUntrusted (3),
31                             invalidCAUntrusted (4),
32                             invalidSsOther (5),
33                             invalidCAOther (6) }
34         MAX-ACCESS  read-only
35         STATUS      current
36         DESCRIPTION
37             "Contains the reason why a SS's certificate is deemed valid
38            or invalid. Return unknown if the SS is running PKM mode.
39            ValidSsChained means the certificate is valid because it
40            chains to a valid certificate. ValidSsTrusted means the
41            certificate is valid because it has been provisioned to be
42            trusted. InvalidSsUntrusted means the certificate is
43            invalid because it has been provisioned to be untrusted.
44            InvalidCAUntrusted means the certificate is invalid
45            because it chains to an untrusted certificate.
46            InvalidSsOther and InvalidCAOther refer to errors in
47            parsing, validity periods, etc, which are attributable to
48            the SS certificate or its chain respectively."
49     ::= { wmanIfBsPkmAuthEntry 18 }
50
51     wmanIfBsPkmAuthBpkmSsCert OBJECT-TYPE
52         SYNTAX      OCTET STRING
53         MAX-ACCESS  read-only
54         STATUS      current

```

```

1      DESCRIPTION
2          "The X509 SS Certificate sent as part of a PKM
3          Authorization Request."
4      ::= { wmanIfBsPkmAuthEntry 19 }
5
6      --
7      -- Table wmanIfBsPkmTEKTable
8      wmanIfBsPkmTEKTable OBJECT-TYPE
9          SYNTAX      SEQUENCE OF      WmanIfBsPkmTEKEntry
10         MAX-ACCESS  not-accessible
11         STATUS      current
12         DESCRIPTION
13             "This table describes the attributes of each Traffic
14             Encryption Key (TEK) association. The BS maintains one TEK
15             association per SAID on each BS wireless interface."
16         ::= { wmanIfBsPkmObjects 3 }
17
18     wmanIfBsPkmTEKEntry OBJECT-TYPE
19         SYNTAX      WmanIfBsPkmTEKEntry
20         MAX-ACCESS  not-accessible
21         STATUS      current
22         DESCRIPTION
23             "Each entry contains objects describing attributes of one
24             TEK association on a particular BS wireless interface. The
25             BS MUST create one entry per SAID per wireless interface,
26             based on the receipt of a Key Request message, and MUST not
27             delete the entry before the SS authorization for the SAID
28             permanently expires."
29         INDEX      { ifIndex, wmanIfBsPkmTEKSAID }
30         ::= { wmanIfBsPkmTEKTable 1 }
31
32     wmanIfBsPkmTEKEntry ::= SEQUENCE {
33         wmanIfBsPkmTEKSAID                Integer32,
34         wmanIfBsPkmTEKSAType              INTEGER,
35         wmanIfBsPkmTEKDataEncryptAlg      INTEGER,
36         wmanIfBsPkmTEKDataAuthentAlg     INTEGER,
37         wmanIfBsPkmTEKEncryptAlg         INTEGER,
38         wmanIfBsPkmTEKLifetime            Integer32,
39         wmanIfBsPkmTEKKeySequenceNumber   Integer32,
40         wmanIfBsPkmTEKExpiresOld          DateAndTime,
41         wmanIfBsPkmTEKExpiresNew         DateAndTime,
42         wmanIfBsPkmTEKReset               TruthValue,
43         wmanIfBsPkmKeyRequests            Counter32,
44         wmanIfBsPkmKeyReplies             Counter32,
45         wmanIfBsPkmKeyRejects            Counter32,
46         wmanIfBsPkmTEKInvalids           Counter32,
47         wmanIfBsPkmKeyRejectErrorCode     INTEGER,
48         wmanIfBsPkmKeyRejectErrorString   SnmpAdminString,
49         wmanIfBsPkmTEKInvalidErrorCode   INTEGER,
50         wmanIfBsPkmTEKInvalidErrorString SnmpAdminString
51     }
52
53     wmanIfBsPkmTEKSAID OBJECT-TYPE
54         SYNTAX      Integer32 (0..65536)

```

```

1      MAX-ACCESS  not-accessible
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the WiMAX Security Association
5          ID (SAID)."

```

```

1         algorithm being utilized."
2     REFERENCE
3         "IEEE 802.16 standard; Table 303"
4     ::= { wmanIfBsPkmTEKEntry 5 }
5
6     wmanIfBsPkmTEKLifetime OBJECT-TYPE
7         SYNTAX      Integer32 (1800..604800)
8         UNITS       "seconds"
9         MAX-ACCESS  read-write
10        STATUS      current
11        DESCRIPTION
12            "The value of this object is the lifetime, in seconds, the
13            BS assigns to keys for this TEK association."
14        REFERENCE
15            "Table 341 in IEEE 802.16REVD/D5-2004"
16        DEFVAL      { 43200 }
17        ::= { wmanIfBsPkmTEKEntry 6 }
18
19        wmanIfBsPkmTEKKeySequenceNumber OBJECT-TYPE
20            SYNTAX      Integer32 (0..3)
21            MAX-ACCESS  read-only
22            STATUS      current
23            DESCRIPTION
24                "The value of this object is the most recent TEK key
25                sequence number for this SAID."
26            REFERENCE
27                "IEEE 802.16 standard; 11.9.5"
28            ::= { wmanIfBsPkmTEKEntry 7 }
29
30        wmanIfBsPkmTEKExpiresOld OBJECT-TYPE
31            SYNTAX      DateAndTime
32            MAX-ACCESS  read-only
33            STATUS      current
34            DESCRIPTION
35                "The value of this object is the actual clock time for
36                expiration of the immediate predecessor of the most recent
37                TEK for this FSM. If this FSM has only one TEK, then the
38                value is the time of activation of this FSM."
39            ::= { wmanIfBsPkmTEKEntry 8 }
40
41        wmanIfBsPkmTEKExpiresNew OBJECT-TYPE
42            SYNTAX      DateAndTime
43            MAX-ACCESS  read-only
44            STATUS      current
45            DESCRIPTION
46                "The value of this object is the actual clock time for
47                expiration of the most recent TEK for this FSM."
48            ::= { wmanIfBsPkmTEKEntry 9 }
49
50        wmanIfBsPkmTEKReset OBJECT-TYPE
51            SYNTAX      TruthValue
52            MAX-ACCESS  read-write
53            STATUS      current
54            DESCRIPTION

```

```

1           "Setting this object to TRUE causes the BS to invalidate
2           the current active TEK(s) (plural due to key transition
3           periods), and to generate a new TEK for the associated
4           SAID; the BS MAY also generate an unsolicited TEK Invalid
5           message, to optimize the TEK synchronization between the BS
6           and the SS. Reading this object always returns  FALSE."
7       ::= { wmanIfBsPkmTEKEntry 10 }
8
9   wmanIfBsPkmKeyRequests OBJECT-TYPE
10      SYNTAX      Counter32
11      MAX-ACCESS  read-only
12      STATUS      current
13      DESCRIPTION
14          "The value of this object is the count of times the BS has
15          received a Key Request message."
16      ::= { wmanIfBsPkmTEKEntry 11 }
17
18   wmanIfBsPkmKeyReplies OBJECT-TYPE
19      SYNTAX      Counter32
20      MAX-ACCESS  read-only
21      STATUS      current
22      DESCRIPTION
23          "The value of this object is the count of times the BS has
24          transmitted a Key Reply message."
25      ::= { wmanIfBsPkmTEKEntry 12 }
26
27   wmanIfBsPkmKeyRejects OBJECT-TYPE
28      SYNTAX      Counter32
29      MAX-ACCESS  read-only
30      STATUS      current
31      DESCRIPTION
32          "The value of this object is the count of times the BS has
33          transmitted a Key Reject message."
34      ::= { wmanIfBsPkmTEKEntry 13 }
35
36   wmanIfBsPkmTEKInvalids OBJECT-TYPE
37      SYNTAX      Counter32
38      MAX-ACCESS  read-only
39      STATUS      current
40      DESCRIPTION
41          "The value of this object is the count of times the BS has
42          transmitted a TEK Invalid message."
43      ::= { wmanIfBsPkmTEKEntry 14 }
44
45   wmanIfBsPkmKeyRejectErrorCode OBJECT-TYPE
46      SYNTAX      INTEGER {noInformation(0),
47                      unauthorizedSaid(2)
48                      }
49      MAX-ACCESS  read-only
50      STATUS      current
51      DESCRIPTION
52          "The value of this object is the enumerated; description of
53          the Error-Code in the most recent Key Reject message sent
54          in response to a Key Request for this SAID."

```

```

1      REFERENCE
2          "IEEE 802.16 standard; Table 371"
3      ::= { wmanIfBsPkmTEKEntry 15 }
4
5      wmanIfBsPkmKeyRejectErrorString OBJECT-TYPE
6          SYNTAX      SnmpAdminString (SIZE (0..128))
7          MAX-ACCESS  read-only
8          STATUS      current
9          DESCRIPTION
10             "The value of this object is the Display-String in the most
11             recent Key Reject message sent in response to a Key Request
12             for this SAID. This is a zero length string if no Key
13             Reject message has been received since reboot."
14      ::= { wmanIfBsPkmTEKEntry 16 }
15
16      wmanIfBsPkmTEKInvalidErrorCode OBJECT-TYPE
17          SYNTAX      INTEGER {noInformation(0),
18                          invalidKeySequence(4)}
19          MAX-ACCESS  read-only
20          STATUS      current
21          DESCRIPTION
22             "The value of this object is the enumerated description of
23             the Error-Code in the most recent TEK Invalid message sent
24             in association with this SAID."
25          REFERENCE
26             "IEEE 802.16 standard; Table 371"
27      ::= { wmanIfBsPkmTEKEntry 17 }
28
29      wmanIfBsPkmTEKInvalidErrorString OBJECT-TYPE
30          SYNTAX      SnmpAdminString (SIZE (0..128))
31          MAX-ACCESS  read-only
32          STATUS      current
33          DESCRIPTION
34             "The value of this object is the Display-String in the most
35             recent TEK Invalid message sent in association with this
36             SAID. This is a zero length string if no TEK Invalid
37             message has been received since reboot."
38      ::= { wmanIfBsPkmTEKEntry 18 }
39
40      --
41      -- Base station Notification Group
42      -- wmanIfBsNotificationObjects contains the BS SNMP Trap objects
43      wmanIfBsNotification OBJECT IDENTIFIER ::= { wmanIfBsObjects 5 }
44      wmanIfBsTrapDefinitions OBJECT IDENTIFIER ::= { wmanIfBsNotification 1 }
45      wmanIfBsTrapControl OBJECT IDENTIFIER ::= { wmanIfBsNotification 2 }
46
47      wmanIfBsTrapControlRegister OBJECT-TYPE
48          SYNTAX      BITS {wmanBSSsStatusNotification (0),
49                          wmanBSSsDynamicServiceFail (1),
50                          wmanBsPowerStatusChange (2),
51                          wmanBsFanStatusChange (3),
52                          wmanBsTemperatureChange (4),
53                          wmanBSSsRssiStatusChange (5),
54                          wmanBSSsBPKMFail (6)}

```

```

1           }
2     MAX-ACCESS read-write
3     STATUS    current
4     DESCRIPTION
5         "The object is used to enable Base Station traps. From left
6         to right, the set bit indicates the corresponding Base
7         Station trap is enabled."
8     ::= { wmanIfBsTrapControl 1 }
9
10    --
11    -- BS threshold Definitions
12    wmanIfBsThresholdConfigTable OBJECT-TYPE
13        SYNTAX      SEQUENCE OF wmanIfBsThresholdConfigEntry
14        MAX-ACCESS  not-accessible
15        STATUS      current
16        DESCRIPTION
17            "This table contains threshold objects to be used to detect
18            the threshold crossing events."
19        ::= { wmanIfBsTrapDefinitions 1 }
20
21    wmanIfBsThresholdConfigEntry OBJECT-TYPE
22        SYNTAX      wmanIfBsThresholdConfigEntry
23        MAX-ACCESS  not-accessible
24        STATUS      current
25        DESCRIPTION
26            "This table provides one row for each BS sector, and is
27            ifIndex."
28        INDEX      { ifIndex }
29        ::= { wmanIfBsThresholdConfigTable 1 }
30
31    wmanIfBsThresholdConfigEntry ::= SEQUENCE {
32        wmanIfBsRssiLowThreshold          INTEGER,
33        wmanIfBsRssiHighThreshold        INTEGER,
34        wmanIfBsTempLowAlarmThreshold    INTEGER,
35        wmanIfBsTempLowAlarmRestoredThreshold  INTEGER,
36        wmanIfBsTempHighAlarmThreshold  INTEGER,
37        wmanIfBsTempHighAlarmRestoredThreshold  INTEGER
38    }
39
40    wmanIfBsRssiLowThreshold OBJECT-TYPE
41        SYNTAX      INTEGER
42        UNITS       "dBm"
43        MAX-ACCESS  read-write
44        STATUS      current
45        DESCRIPTION
46            "Low threshold for generating the RSSI alarm trap.
47            The detection of RSSI alarm will be disabled until the
48            RSSI goes above wmanIfBsRssiHighThreshold"
49        ::= { wmanIfBsThresholdConfigEntry 1 }
50
51    wmanIfBsRssiHighThreshold OBJECT-TYPE
52        SYNTAX      INTEGER
53        UNITS       "dBm"
54        MAX-ACCESS  read-write

```



```

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

```

```

1      MAX-ACCESS not-accessible
2      STATUS current
3      DESCRIPTION
4          "This table contains SS notification objects that have been
5              reported by the trap."
6      ::= { wmanIfBsTrapDefinitions 2 }
7
8      wmanIfBsSsNotificationObjectsEntry OBJECT-TYPE
9          SYNTAX WmanIfBsSsNotificationObjectsEntry
10         MAX-ACCESS not-accessible
11         STATUS current
12         DESCRIPTION
13             "This table provides one row for each SS that has
14                 generated traps, and is double indexed by
15                 wmanIfBsTrapSsId and ifIndex for BS sector."
16         INDEX { ifIndex, wmanIfBsTrapSsId }
17         ::= { wmanIfBsSsNotificationObjectsTable 1 }
18
19         wmanIfBsSsNotificationObjectsEntry ::= SEQUENCE {
20             wmanIfBsTrapSsId Unsigned32,
21             wmanIfBsSsMacAddress MacAddress,
22             wmanIfBsSsStatusValue INTEGER,
23             wmanIfBsSsStatusInfo OCTET STRING,
24             wmanIfBsDynamicServiceType INTEGER,
25             wmanIfBsDynamicServiceFailReason OCTET STRING,
26             wmanIfBsSsRssiStatus INTEGER,
27             wmanIfBsSsRssiStatusInfo OCTET STRING
28         }
29
30         wmanIfBsTrapSsId OBJECT-TYPE
31             SYNTAX Unsigned32 (1 .. 4294967295)
32             MAX-ACCESS read-only
33             STATUS current
34             DESCRIPTION
35                 "wmanIfBsTrapSsId identifies the entry in
36                     wmanIfBsSsNotificationObjectsTable."
37             ::= { wmanIfBsSsNotificationObjectsEntry 1 }
38
39         wmanIfBsSsStatusValue OBJECT-TYPE
40             SYNTAX INTEGER {ssInitRangingSucc(1),
41                 ssInitRangingFail(2),
42                 ssRegistered(3),
43                 ssRegistrationFail(4),
44                 ssDeregistered(5),
45                 ssBasicCapabilitySucc(6),
46                 ssBasicCapabilityFail(7),
47                 ssAuthorizationSucc(8),
48                 ssAuthorizationFail(9),
49                 tftpSucc(10),
50                 tftpFail(11),
51                 sfCreationSucc(12),
52                 sfCreationFail(13)
53             }
54     }

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "This object indicates the status of a SS, as it goes
5          through network entry and initialization procedure."
6      ::= { wmanIfBsSsNotificationObjectsEntry 2 }
7
8      wmanIfBsSsStatusInfo  OBJECT-TYPE
9          SYNTAX      OCTET STRING
10         MAX-ACCESS  read-only
11         STATUS      current
12         DESCRIPTION
13             "This object indicates the reason of SS's status change."
14         ::= { wmanIfBsSsNotificationObjectsEntry 3 }
15
16         wmanIfBsDynamicServiceType  OBJECT-TYPE
17             SYNTAX      INTEGER {bsSfCreationReq(1),
18                             bsSfCreationRsp(2),
19                             bsSfCreationAck(3)
20
21             }
22         MAX-ACCESS  read-only
23         STATUS      current
24         DESCRIPTION
25             "This object indicates the dynamic service flow
26             creation command type."
27         ::= { wmanIfBsSsNotificationObjectsEntry 4 }
28
29         wmanIfBsDynamicServiceFailReason  OBJECT-TYPE
30             SYNTAX      OCTET STRING
31             MAX-ACCESS  read-only
32             STATUS      current
33             DESCRIPTION
34                 "This object indicates the reason why the service flow
35                 cration has failed."
36             ::= { wmanIfBsSsNotificationObjectsEntry 5 }
37
38         wmanIfBsSsRssiStatus  OBJECT-TYPE
39             SYNTAX      INTEGER {bsRssiAlarm(1),
40                             bsRssiNoAlarm(2)
41
42             }
43         MAX-ACCESS  read-only
44         STATUS      current
45         DESCRIPTION
46             "A RSSI alarm is generated if the RSSI is lower than
47             wmanIfBsLowRssiThreshold."
48         ::= { wmanIfBsSsNotificationObjectsEntry 6 }
49
50         wmanIfBsSsRssiStatusInfo  OBJECT-TYPE
51             SYNTAX      OCTET STRING
52             MAX-ACCESS  read-only
53             STATUS      current
54             DESCRIPTION

```

```

1           "This object indicates the reason why RSSI alarm is
2             generated."
3           ::= { wmanIfBsSsNotificationObjectsEntry 7 }
4
5           --
6           -- Subscriber station Notification Trap Definitions
7           wmanBsSsStatusNotificationTrap NOTIFICATION-TYPE
8             OBJECTS      {ifIndex,
9                           wmanIfBsTrapSsId,
10                          wmanIfBsSsMacAddress,
11                          wmanIfBsSsStatusValue,
12                          wmanIfBsSsStatusInfo
13                         }
14             STATUS      current
15             DESCRIPTION
16               "This trap reports the status of a SS. Based on this
17                 notification the NMS will issue an alarm with certain
18                 severity depending on the status and the reason received."
19             ::= { wmanIfBsTrapDefinitions 3 }
20
21           wmanBsSsDynamicServiceFailTrap NOTIFICATION-TYPE
22             OBJECTS      {ifIndex,
23                           wmanIfBsTrapSsId,
24                           wmanIfBsSsMacAddress,
25                           wmanIfBsDynamicServiceType,
26                           wmanIfBsDynamicServiceFailReason
27                         }
28             STATUS      current
29             DESCRIPTION
30               "An event to report the failure of a dynamic service
31                 operation happened during the dynamic services process
32                 and detected in the Bs side."
33             ::= { wmanIfBsTrapDefinitions 4 }
34
35           wmanBsSsRssiStatusChangeTrap NOTIFICATION-TYPE
36             OBJECTS      {ifIndex,
37                           wmanIfBsTrapSsId,
38                           wmanIfBsSsMacAddress,
39                           wmanIfBsSsRssiStatus,
40                           wmanIfBsSsRssiStatusInfo
41                         }
42             STATUS      current
43             DESCRIPTION
44               "An event to report that the uplink RSSI is below
45                 wmanIfBsLowRssiThreshold, or above
46                 wmanIfBsHighRssiThreshold after restore."
47             ::= { wmanIfBsTrapDefinitions 5 }
48
49           wmanBsSsBPKMFailTrap NOTIFICATION-TYPE
50             OBJECTS      {wmanIfBsSsMacAddress}
51             STATUS      current
52             DESCRIPTION
53               "An event to report the failure of a BPKM operation."
54             ::= { wmanIfBsTrapDefinitions 6 }

```

```

1
2  --
3  -- Base station Notification Object Definitions
4  wmanIfBsNotificationObjectsTable OBJECT-TYPE
5      SYNTAX      SEQUENCE OF WmanIfBsNotificationObjectsEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "This table contains BS notification objects that have been
10         reported by the trap."
11         ::= { wmanIfBsTrapDefinitions 7 }
12
13  wmanIfBsNotificationObjectsEntry OBJECT-TYPE
14      SYNTAX      WmanIfBsNotificationObjectsEntry
15      MAX-ACCESS  not-accessible
16      STATUS      current
17      DESCRIPTION
18          "This table provides one row for each BS sector that has
19         generated traps, and is indexed by ifIndex."
20      INDEX      { ifIndex }
21      ::= { wmanIfBsNotificationObjectsTable 1 }
22
23  WmanIfBsNotificationObjectsEntry ::= SEQUENCE {
24      wmanIfBsPowerStatus          INTEGER,
25      wmanIfBsFanStatus            INTEGER,
26      wmanIfBsTemperatureStatus    INTEGER,
27      wmanIfBsPowerStatusInfo      OCTET STRING,
28      wmanIfBsFanStatusInfo        OCTET STRING,
29      wmanIfBsTemperatureStatusInfo OCTET STRING
30  }
31
32  wmanIfBsPowerStatus OBJECT-TYPE
33      SYNTAX      INTEGER {priOnSecStandby(0),
34                      secOnPriStandby(1),
35                      priOnSecFailed(2),
36                      secOnPriFailed(3)
37                      }
38      MAX-ACCESS  read-only
39      STATUS      current
40      DESCRIPTION
41          "Describes the status of the power supply in BS."
42      ::= { wmanIfBsNotificationObjectsEntry 1 }
43
44  wmanIfBsFanStatus OBJECT-TYPE
45      SYNTAX      INTEGER {fanFail(1),
46                      fanSucc(2)
47                      }
48      MAX-ACCESS  read-only
49      STATUS      current
50      DESCRIPTION
51          "Describes the status of the fan in BS."
52      ::= { wmanIfBsNotificationObjectsEntry 2 }
53
54  wmanIfBsTemperatureStatus OBJECT-TYPE

```

```

1      SYNTAX      INTEGER {lowTempReached(1),
2                      highTempReached(2),
3                      temperatureNormal(3)
4                      }
5      MAX-ACCESS  read-only
6      STATUS      current
7      DESCRIPTION
8          "lowTempReached event is generated when temperature goes
9          below wmanIfBsTempLowAlarmThreshold.
10         temperatureNormal event is generated when temperature
11         goes above wmanIfBsTempLowAlarmRestoredThreshold or
12         below wmanIfBsTempHighAlarmRestoredThreshold after alarm.
13         highTempReached event is generated when temperature goes
14         above wmanIfBsTempHighAlarmThreshold."
15     ::= { wmanIfBsNotificationObjectsEntry 3 }
16
17 wmanIfBsPowerStatusInfo OBJECT-TYPE
18     SYNTAX      OCTET STRING
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22         "Display the power supply ststus in text form."
23     ::= { wmanIfBsNotificationObjectsEntry 4 }
24
25 wmanIfBsFanStatusInfo OBJECT-TYPE
26     SYNTAX      OCTET STRING
27     MAX-ACCESS  read-write
28     STATUS      current
29     DESCRIPTION
30         "Display the fan ststus in text form."
31     ::= { wmanIfBsNotificationObjectsEntry 5 }
32
33 wmanIfBsTemperatureStatusInfo OBJECT-TYPE
34     SYNTAX      OCTET STRING
35     MAX-ACCESS  read-write
36     STATUS      current
37     DESCRIPTION
38         "Display the temperature ststus in text form."
39     ::= { wmanIfBsNotificationObjectsEntry 6 }
40
41 --
42 -- Base station Notification Trap Definitions
43 wmanBSPowerStatusChangeTrap NOTIFICATION-TYPE
44     OBJECTS      {wmanIfBsPowerStatus,
45                  wmanIfBsPowerStatusInfo
46                  }
47     STATUS      current
48     DESCRIPTION
49         "An event to report a change in the status of the power
50         supply in BS. Typically it represents a failure."
51     ::= { wmanIfBsTrapDefinitions 8 }
52
53 wmanBSFanStatusTrap NOTIFICATION-TYPE
54     OBJECTS      {wmanIfBsFanStatus,

```

```

1           wmanIfBSFanStatusInfo
2           }
3     STATUS      current
4     DESCRIPTION
5       "An event to report the status of the fan inside the BS."
6     ::= { wmanIfBSTrapDefinitions 9 }
7
8     wmanBSTemperatureChangeTrap NOTIFICATION-TYPE
9       OBJECTS      {wmanIfBSTemperatureStatus,
10                  wmanIfBSTemperatureStatusInfo
11                  }
12      STATUS      current
13      DESCRIPTION
14        "An alarm event will be generated when the temperature goes
15        above wmanIfBSTempHighAlarmThreshold or below
16        wmanIfBSTempLowAlarmThreshold. An event reporting the alarm
17        has disappeared when the temperature goes below
18        wmanIfBSTempHighAlarmRestoredThreshold or above
19        wmanIfBSTempLowAlarmRestoredThreshold."
20      ::= { wmanIfBSTrapDefinitions 10 }
21
22      --
23      -- SS object group - containing tables and objects to be implemented in
24      -- the Subscriber station
25      --
26      -- wmanIfSsSystem contain the Subscriber Station System objects
27      wmanIfSsSystem OBJECT IDENTIFIER ::= { wmanIfSsObjects 1 }
28
29      wmanIfSsConfigFileEncodingTable OBJECT-TYPE
30        SYNTAX      SEQUENCE OF wmanIfSsConfigFileEncodingEntry
31        MAX-ACCESS  not-accessible
32        STATUS      current
33        DESCRIPTION
34          "This table contains configuration file encoding
35          information of the SS."
36        REFERENCE
37          "Section 11.2 in IEEE 802.16REVD/D5-2004"
38        ::= { wmanIfSsSystem 1 }
39
40      wmanIfSsConfigFileEncodingEntry OBJECT-TYPE
41        SYNTAX      wmanIfSsConfigFileEncodingEntry
42        MAX-ACCESS  not-accessible
43        STATUS      current
44        DESCRIPTION
45          "This table has only one entry, and is indexed
46          by ifIndex."
47        INDEX { ifIndex }
48        ::= { wmanIfSsConfigFileEncodingTable 1 }
49
50      wmanIfSsConfigFileEncodingEntry ::= SEQUENCE {
51        wmanIfSsMicConfigSetting      OCTET STRING,
52        wmanIfSsVendorId               OCTET STRING,
53        wmanIfSsHwId                  OCTET STRING,
54        wmanIfSsSwVersion              OCTET STRING,

```

```

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

```



```

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

```

```

1      wmanIfsStoDRetryPeriod          INTEGER,
2      wmanIfsSt1Timeout               INTEGER,
3      wmanIfsSt2Timeout               INTEGER,
4      wmanIfsSt3Timeout               INTEGER,
5      wmanIfsSt4Timeout               INTEGER,
6      wmanIfsSt6Timeout               INTEGER,
7      wmanIfsSt12Timeout              INTEGER,
8      wmanIfsSt14Timeout              INTEGER,
9      wmanIfsSt16Timeout              INTEGER,
10     wmanIfsSt18Timeout               INTEGER,
11     wmanIfsSt19Timeout               INTEGER,
12     wmanIfsSt20Timeout               INTEGER,
13     wmanIfsSt21Timeout               INTEGER,
14     wmanIfsSsBCRequestRetries       INTEGER,
15     wmanIfsStftpCpltRetries         INTEGER,
16     wmanIfsSt26Timeout               INTEGER,
17     wmanIfsSDLManagProcTime          INTEGER,
18     wmanIfsSConfigurationRowStatus  RowStatus
19   }
20
21   wmanIfsSLostDLMapInterval OBJECT-TYPE
22     SYNTAX      INTEGER(0..600)
23     UNITS       "milliseconds"
24     MAX-ACCESS  read-write
25     STATUS      current
26     DESCRIPTION
27       "Time since last received DL-MAP message before downlink
28       synchronization is considered lost in ms."
29     ::= { wmanIfsSConfigurationEntry 1 }
30
31   wmanIfsSLostULMapInterval OBJECT-TYPE
32     SYNTAX      INTEGER(0..600)
33     UNITS       "milliseconds"
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37       "Time since last received UL-MAP message before downlink
38       synchronization is considered lost in ms."
39     ::= { wmanIfsSConfigurationEntry 2 }
40
41   wmanIfsSContentionRangRetries OBJECT-TYPE
42     SYNTAX      INTEGER(16..65535)
43     MAX-ACCESS  read-write
44     STATUS      current
45     DESCRIPTION
46       "Number of retries on contention Ranging Requests."
47     ::= { wmanIfsSConfigurationEntry 3 }
48
49   wmanIfsSRequestRetries OBJECT-TYPE
50     SYNTAX      INTEGER(16..65535)
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54       "Number of retries on bandwidth allocation requests."

```

```

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

```

```

1          SYNTAX      INTEGER(3..65535)
2          MAX-ACCESS  read-write
3          STATUS      current
4          DESCRIPTION
5              "Number of Retries per Time of Day Retry Period."
6          ::= { wmanIfsConfigurationEntry 11 }
7
8          wmanIfsStoDRetryPeriod OBJECT-TYPE
9              SYNTAX      INTEGER(5..65535)
10             UNITS        "minutes"
11             MAX-ACCESS  read-write
12             STATUS      current
13             DESCRIPTION
14                 "Time of Day Retry Period."
15             ::= { wmanIfsConfigurationEntry 12 }
16
17             wmanIfsSt1Timeout OBJECT-TYPE
18                 SYNTAX      INTEGER(0..50000)
19                 UNITS        "milliseconds"
20                 MAX-ACCESS  read-write
21                 STATUS      current
22                 DESCRIPTION
23                     "Wait for DCD timeout in ms."
24                 ::= { wmanIfsConfigurationEntry 13 }
25
26             wmanIfsSt2Timeout OBJECT-TYPE
27                 SYNTAX      INTEGER(0..10000)
28                 UNITS        "milliseconds"
29                 MAX-ACCESS  read-write
30                 STATUS      current
31                 DESCRIPTION
32                     "Wait for broadcast ranging timeout in ms."
33                 ::= { wmanIfsConfigurationEntry 14 }
34
35             wmanIfsSt3Timeout OBJECT-TYPE
36                 SYNTAX      INTEGER(0..200)
37                 UNITS        "milliseconds"
38                 MAX-ACCESS  read-write
39                 STATUS      current
40                 DESCRIPTION
41                     "Ranging Response reception timeout following the
42                     transmission of a Ranging Request in ms."
43                 ::= { wmanIfsConfigurationEntry 15 }
44
45             wmanIfsSt4Timeout OBJECT-TYPE
46                 SYNTAX      INTEGER(30..35)
47                 UNITS        "seconds"
48                 MAX-ACCESS  read-write
49                 STATUS      current
50                 DESCRIPTION
51                     "Wait for unicast ranging opportunity. If the pending until
52                     complete field was used earlier by this SS, then the value
53                     of that field shall be added to this interval in s."
54                 ::= { wmanIfsConfigurationEntry 16 }

```

```
1
2 wmanIfsSt6Timeout OBJECT-TYPE
3     SYNTAX      INTEGER(0..3000)
4     UNITS       "milliseconds"
5     MAX-ACCESS  read-write
6     STATUS      current
7     DESCRIPTION
8         "Wait for registration response in ms."
9     ::= { wmanIfsConfigurationEntry 17 }
10
11 wmanIfsSt12Timeout OBJECT-TYPE
12     SYNTAX      INTEGER (0..50000)
13     UNITS       "milliseconds"
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "Wait for UCD descriptor in ms."
18     ::= { wmanIfsConfigurationEntry 18 }
19
20 wmanIfsSt14Timeout OBJECT-TYPE
21     SYNTAX      INTEGER(0..200)
22     UNITS       "milliseconds"
23     MAX-ACCESS  read-write
24     STATUS      current
25     DESCRIPTION
26         "Wait for DSX-RVD Timeout in ms."
27     ::= { wmanIfsConfigurationEntry 19 }
28
29 wmanIfsSt16Timeout OBJECT-TYPE
30     SYNTAX      INTEGER(10..65535)
31     UNITS       "milliseconds"
32     MAX-ACCESS  read-write
33     STATUS      current
34     DESCRIPTION
35         "wait for bandwidth request grant in ms."
36     ::= { wmanIfsConfigurationEntry 20 }
37
38 wmanIfsSt18Timeout OBJECT-TYPE
39     SYNTAX      INTEGER(0..65535)
40     UNITS       "milliseconds"
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "wait for SBC-RSP timeout in ms."
45     ::= { wmanIfsConfigurationEntry 21 }
46
47 wmanIfsSt19Timeout OBJECT-TYPE
48     SYNTAX      INTEGER(0..65535)
49     UNITS       "milliseconds"
50     MAX-ACCESS  read-write
51     STATUS      current
52     DESCRIPTION
53         "Time DL-channel remains unusable in ms."
54     ::= { wmanIfsConfigurationEntry 22 }
```

```

1
2  wmanIfSsT20Timeout OBJECT-TYPE
3      SYNTAX      INTEGER(0..65535)
4      UNITS       "milliseconds"
5      MAX-ACCESS  read-write
6      STATUS      current
7      DESCRIPTION
8          "Time SS searches for preambles on a given channel in ms."
9      ::= { wmanIfSsConfigurationEntry 23 }
10
11 wmanIfSsT21Timeout OBJECT-TYPE
12     SYNTAX      INTEGER(0..10000)
13     UNITS       "milliseconds"
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "Time SS searches for DL-MAP on a given channel in ms."
18     ::= { wmanIfSsConfigurationEntry 24 }
19
20 wmanIfSsSBCRequestRetries OBJECT-TYPE
21     SYNTAX      INTEGER(3..16)
22     MAX-ACCESS  read-write
23     STATUS      current
24     DESCRIPTION
25         "Number of retries on SBC Request."
26     ::= { wmanIfSsConfigurationEntry 25 }
27
28 wmanIfSsTftpCpltRetries OBJECT-TYPE
29     SYNTAX      INTEGER(3..16)
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33         "Number of retries on TFTP-CPLT."
34     ::= { wmanIfSsConfigurationEntry 26 }
35
36 wmanIfSsT26Timeout OBJECT-TYPE
37     SYNTAX      INTEGER(10..200)
38     UNITS       "milliseconds"
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "Wait for TFTP-RSP in ms."
43     ::= { wmanIfSsConfigurationEntry 27 }
44
45 wmanIfSsDLManagProcTime OBJECT-TYPE
46     SYNTAX      INTEGER(0..200)
47     UNITS       "micro seconds"
48     MAX-ACCESS  read-write
49     STATUS      current
50     DESCRIPTION
51         "Max. time between reception of Fast Power Control
52         management message and compliance to its instructions
53         by SS in us."
54     ::= { wmanIfSsConfigurationEntry 28 }

```

```

1
2 wmanIfSsConfigurationRowStatus 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     ::= { wmanIfSsConfigurationEntry 29 }
15
16 -- Subscriber station PKM group
17 -- wmanIfSsPkmObjects contain the Subscriber Station Privacy Sublayer
18 -- objects
19 wmanIfSsPkmObjects OBJECT IDENTIFIER ::= { wmanIfSsObjects 3 }
20
21 --
22 -- Table wmanIfSsPkmBaseTable
23 --
24 wmanIfSsPkmBaseTable OBJECT-TYPE
25     SYNTAX      SEQUENCE OF wmanIfSsPkmBaseEntry
26     MAX-ACCESS  not-accessible
27     STATUS      current
28     DESCRIPTION
29         "This table describes the basic PKM attributes of each
30         SS wireless interface."
31     ::= { wmanIfSsPkmObjects 1 }
32
33 wmanIfSsPkmBaseEntry OBJECT-TYPE
34     SYNTAX      wmanIfSsPkmBaseEntry
35     MAX-ACCESS  not-accessible
36     STATUS      current
37     DESCRIPTION
38         "Each entry contains objects describing attributes of one
39         SS wireless interface."
40     INDEX      { ifIndex }
41     ::= { wmanIfSsPkmBaseTable 1 }
42
43 wmanIfSsPkmBaseEntry ::= SEQUENCE {
44     wmanIfSsPkmPrivacyEnable          TruthValue,
45     wmanIfSsPkmPublicKey              OCTET STRING,
46     wmanIfSsPkmAuthGraceTime         Integer32,
47     wmanIfSsPkmTEKGraceTime          Integer32,
48     wmanIfSsPkmAuthWaitTimeout       Integer32,
49     wmanIfSsPkmReauthWaitTimeout     Integer32,
50     wmanIfSsPkmOpWaitTimeout         Integer32,
51     wmanIfSsPkmRekeyWaitTimeout      Integer32,
52     wmanIfSsPkmAuthRejectWaitTimeout Integer32
53 }
54

```

```

1  wmanIfSsPkmPrivacyEnable OBJECT-TYPE
2      SYNTAX      TruthValue
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "This object identifies whether this SS is provisioned to
7          run Baseline Privacy Plus."
8      ::= { wmanIfSsPkmBaseEntry 1 }
9
10 wmanIfSsPkmPublicKey OBJECT-TYPE
11     SYNTAX      OCTET STRING (SIZE (140))
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "The value of this object is a DER-encoded RSAPublicKey
16         ASN.1 type string, as defined in the RSA Encryption
17         Standard (PKCS#1) [10], corresponding to the public key of
18         the SS. The 74, 106, 140, 204, and 270 byte key encoding
19         lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
20         and 2048 public moduli respectively."
21     ::= { wmanIfSsPkmBaseEntry 2 }
22
23 wmanIfSsPkmAuthGraceTime OBJECT-TYPE
24     SYNTAX      Integer32 (300..3024000)
25     UNITS       "seconds"
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "The value of this object is the grace time for an
30         authorization key. A SS is expected to start trying to get
31         a new authorization key beginning AuthGraceTime seconds
32         before the authorization key actually expires."
33     REFERENCE
34         "Table 341 in IEEE 802.16REVD/D5-2004"
35     DEFVAL      { 600 }
36     ::= { wmanIfSsPkmBaseEntry 3 }
37
38 wmanIfSsPkmTEKGraceTime OBJECT-TYPE
39     SYNTAX      Integer32 (300..3024000)
40     UNITS       "seconds"
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44         "The value of this object is the grace time for the TEK in
45         seconds. The SS is expected to start trying to acquire a
46         new TEK beginning TEK GraceTime seconds before the
47         expiration of the most recent TEK."
48     REFERENCE
49         "Table 341 in IEEE 802.16REVD/D5-2004"
50     DEFVAL      { 3600 }
51     ::= { wmanIfSsPkmBaseEntry 4 }
52
53 wmanIfSsPkmAuthWaitTimeout OBJECT-TYPE
54     SYNTAX      Integer32 (2..30)

```



```

1      UNITS      "seconds"
2      MAX-ACCESS read-only
3      STATUS     current
4      DESCRIPTION
5          "The value of this object is the Authorize wait Timeout."
6      REFERENCE
7          "Table 341 in IEEE 802.16REvd/D5-2004"
8      DEFVAL     { 10 }
9      ::= { wmanIfSsPkmBaseEntry 5 }
10
11     wmanIfSsPkmReauthwaitTimeout OBJECT-TYPE
12         SYNTAX      Integer32 (2..30)
13         UNITS       "seconds"
14         MAX-ACCESS read-only
15         STATUS      current
16         DESCRIPTION
17             "The value of this object is the Reauthorize wait Timeout
18             in seconds."
19         REFERENCE
20             "Table 341 in IEEE 802.16REvd/D5-2004"
21         DEFVAL     { 10 }
22         ::= { wmanIfSsPkmBaseEntry 6 }
23
24     wmanIfSsPkmOpwaitTimeout OBJECT-TYPE
25         SYNTAX      Integer32 (1..10)
26         UNITS       "seconds"
27         MAX-ACCESS read-only
28         STATUS      current
29         DESCRIPTION
30             "The value of this object is the Operational wait Timeout
31             in seconds."
32         REFERENCE
33             "Table 341 in IEEE 802.16REvd/D5-2004"
34         DEFVAL     { 1 }
35         ::= { wmanIfSsPkmBaseEntry 7 }
36
37     wmanIfSsPkmRekeywaitTimeout OBJECT-TYPE
38         SYNTAX      Integer32 (1..10)
39         UNITS       "seconds"
40         MAX-ACCESS read-only
41         STATUS      current
42         DESCRIPTION
43             "The value of this object is the Rekey wait Timeout in
44             seconds."
45         REFERENCE
46             "Table 341 in IEEE 802.16REvd/D5-2004"
47         DEFVAL     { 1 }
48         ::= { wmanIfSsPkmBaseEntry 8 }
49
50     wmanIfSsPkmAuthRejectwaitTimeout OBJECT-TYPE
51         SYNTAX      Integer32 (10..600)
52         UNITS       "seconds"
53         MAX-ACCESS read-only
54         STATUS      current

```

```

1      DESCRIPTION
2          "The value of this object is the Authorization Reject Wait
3          Timeout in seconds."
4      REFERENCE
5          "Table 341 in IEEE 802.16REVd/D5-2004"
6      DEFVAL      { 60 }
7      ::= { wmanIfSsPkmBaseEntry 9 }
8
9      --
10     -- Table wmanIfSsPkmAuthTable
11     --
12     wmanIfSsPkmAuthTable OBJECT-TYPE
13         SYNTAX      SEQUENCE OF      wmanIfSsPkmAuthEntry
14         MAX-ACCESS  not-accessible
15         STATUS      current
16         DESCRIPTION
17             "This table describes the PKM attributes related
18             to the authorization for each SS wireless interface."
19         ::= { wmanIfSsPkmObjects 2 }
20
21     wmanIfSsPkmAuthEntry OBJECT-TYPE
22         SYNTAX      wmanIfSsPkmAuthEntry
23         MAX-ACCESS  not-accessible
24         STATUS      current
25         DESCRIPTION
26             "Each entry contains objects describing attributes of one
27             SS wireless interface."
28         INDEX      { ifIndex }
29         ::= { wmanIfSsPkmAuthTable 1 }
30
31     wmanIfSsPkmAuthEntry ::= SEQUENCE {
32         wmanIfSsPkmAuthState          INTEGER,
33         wmanIfSsPkmAuthKeySequenceNumber  Integer32,
34         wmanIfSsPkmAuthExpiresOld      DateAndTime,
35         wmanIfSsPkmAuthExpiresNew      DateAndTime,
36         wmanIfSsPkmAuthReset           TruthValue,
37         wmanIfSsPkmAuthentInfos         Counter32,
38         wmanIfSsPkmAuthRequests         Counter32,
39         wmanIfSsPkmAuthReplies         Counter32,
40         wmanIfSsPkmAuthRejects         Counter32,
41         wmanIfSsPkmAuthInvalids        Counter32,
42         wmanIfSsPkmAuthRejectErrorCode  INTEGER,
43         wmanIfSsPkmAuthRejectErrorString SnmpAdminString,
44         wmanIfSsPkmAuthInvalidErrorCode INTEGER,
45         wmanIfSsPkmAuthInvalidErrorString SnmpAdminString
46     }
47
48     wmanIfSsPkmAuthState OBJECT-TYPE
49         SYNTAX      INTEGER {start(1),
50             authwait(2),
51             authorized(3),
52             reauthwait(4),
53             authRejectwait(5),
54             silent(6)}

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          "The value of this object is the state of the SS
5          authorization FSM. The start state indicates that FSM is
6          in its initial state."
7      ::= { wmanIfSsPkmAuthEntry 1 }
8
9      wmanIfSsPkmAuthKeySequenceNumber OBJECT-TYPE
10     SYNTAX      Integer32 (0..15)
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         "The value of this object is the most recent authorization
15         key sequence number for this FSM."
16     ::= { wmanIfSsPkmAuthEntry 2 }
17
18     wmanIfSsPkmAuthExpiresOld OBJECT-TYPE
19     SYNTAX      DateAndTime
20     MAX-ACCESS  read-only
21     STATUS      current
22     DESCRIPTION
23         "The value of this object is the actual clock time for
24         expiration of the immediate predecessor of the most recent
25         authorization key for this FSM. If this FSM has only one
26         authorization key, then the value is the time of activation
27         of this FSM."
28     ::= { wmanIfSsPkmAuthEntry 3 }
29
30     wmanIfSsPkmAuthExpiresNew OBJECT-TYPE
31     SYNTAX      DateAndTime
32     MAX-ACCESS  read-only
33     STATUS      current
34     DESCRIPTION
35         "The value of this object is the actual clock time for
36         expiration of the most recent authorization key for this
37         FSM."
38     ::= { wmanIfSsPkmAuthEntry 4 }
39
40     wmanIfSsPkmAuthReset OBJECT-TYPE
41     SYNTAX      TruthValue
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45         "Setting this object to TRUE generates a Reauthorize event
46         in the authorization FSM. Reading this object always
47         returns FALSE."
48     ::= { wmanIfSsPkmAuthEntry 5 }
49
50     wmanIfSsPkmAuthentInfos OBJECT-TYPE
51     SYNTAX      Counter32
52     MAX-ACCESS  read-only
53     STATUS      current
54     DESCRIPTION

```

```

1           "The value of this object is the count of times the SS has
2           transmitted an Authentication Information message."
3       ::= { wmanIfSsPkmAuthEntry 6 }
4
5   wmanIfSsPkmAuthRequests OBJECT-TYPE
6       SYNTAX      Counter32
7       MAX-ACCESS  read-only
8       STATUS      current
9       DESCRIPTION
10          "The value of this object is the count of times the SS has
11          transmitted an Authorization Request message."
12      ::= { wmanIfSsPkmAuthEntry 7 }
13
14  wmanIfSsPkmAuthReplies OBJECT-TYPE
15      SYNTAX      Counter32
16      MAX-ACCESS  read-only
17      STATUS      current
18      DESCRIPTION
19          "The value of this object is the count of times the SS has
20          received an Authorization Reply message."
21      ::= { wmanIfSsPkmAuthEntry 8 }
22
23  wmanIfSsPkmAuthRejects OBJECT-TYPE
24      SYNTAX      Counter32
25      MAX-ACCESS  read-only
26      STATUS      current
27      DESCRIPTION
28          "The value of this object is the count of times the SS has
29          received an Authorization Reject message."
30      ::= { wmanIfSsPkmAuthEntry 9 }
31
32  wmanIfSsPkmAuthInvalids OBJECT-TYPE
33      SYNTAX      Counter32
34      MAX-ACCESS  read-only
35      STATUS      current
36      DESCRIPTION
37          "The value of this object is the count of times the SS has
38          received an Authorization Invalid message."
39      ::= { wmanIfSsPkmAuthEntry 10 }
40
41  wmanIfSsPkmAuthRejectErrorCode OBJECT-TYPE
42      SYNTAX      INTEGER {none(1),
43                  unknown(2),
44                  unauthorizedSs(3),
45                  unauthorizedSaid(4),
46                  permanentAuthorizationFailure(8),
47                  timeOfDayNotAcquired(11)}
48      MAX-ACCESS  read-only
49      STATUS      current
50      DESCRIPTION
51          "The value of this object is the enumerated description of
52          the Error-Code in most recent Authorization Reject message
53          received by the SS. This has value unknown(2)if the last
54          Error-Code value was 0, and none(1) if no Authorization

```

```

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

```

```

1      ::= { wmanIfSsPkmObjects 3 }
2
3  wmanIfSsPkmTEKEntry OBJECT-TYPE
4      SYNTAX      WmanIfSsPkmTEKEntry
5      MAX-ACCESS  not-accessible
6      STATUS      current
7      DESCRIPTION
8          "Each entry contains objects describing the TEK association
9          attributes of one SAID. The SS MUST create one entry per
10         SAID, regardless of whether the SAID was obtained from a
11         Registration Response message, from an Authorization Reply
12         message, or from any dynamic SAID establishment
13         mechanisms."
14     INDEX      { ifIndex, wmanIfSsPkmTEKSAID }
15     ::= { wmanIfSsPkmTEKTable 1 }
16
17  wmanIfSsPkmTEKEntry ::= SEQUENCE {
18      wmanIfSsPkmTEKSAID          Integer32,
19      wmanIfSsPkmTEKSAType        INTEGER,
20      wmanIfSsPkmTEKDataEncryptAlg  INTEGER,
21      wmanIfSsPkmTEKDataAuthentAlg  INTEGER,
22      wmanIfSsPkmTEKEncryptAlg      INTEGER,
23      wmanIfSsPkmTEKState           INTEGER,
24      wmanIfSsPkmTEKKeySequenceNumber Integer32,
25      wmanIfSsPkmTEKExpiresOld      DateAndTime,
26      wmanIfSsPkmTEKExpiresNew     DateAndTime,
27      wmanIfSsPkmTEKKeyRequests     Counter32,
28      wmanIfSsPkmTEKKeyReplies      Counter32,
29      wmanIfSsPkmTEKKeyRejects      Counter32,
30      wmanIfSsPkmTEKInvalids        Counter32,
31      wmanIfSsPkmTEKAuthPends       Counter32,
32      wmanIfSsPkmTEKKeyRejectErrorCode INTEGER,
33      wmanIfSsPkmTEKKeyRejectErrorString SnmpAdminString,
34      wmanIfSsPkmTEKInvalidErrorCode INTEGER,
35      wmanIfSsPkmTEKInvalidErrorString SnmpAdminString
36  }
37
38  wmanIfSsPkmTEKSAID OBJECT-TYPE
39      SYNTAX      Integer32 (1..16383)
40      MAX-ACCESS  not-accessible
41      STATUS      current
42      DESCRIPTION
43          "The value of this object is the WiMAX Security Association
44          ID (SAID)."
```

```

45     ::= { wmanIfSsPkmTEKEntry 1 }
46
47  wmanIfSsPkmTEKSAType OBJECT-TYPE
48      SYNTAX      INTEGER {primarySA(0),
49                      staticSA(1),
50                      dynamicSA(2)}
51      MAX-ACCESS  read-only
52      STATUS      current
53      DESCRIPTION
54          "The value of this object is the type of security

```

```

1         association."
2     REFERENCE
3         "IEEE 802.16 standard; 11.9.18"
4     ::= { wmanIfSsPkmTEKEntry 2 }
5
6     wmanIfSsPkmTEKDataEncryptAlg OBJECT-TYPE
7         SYNTAX      INTEGER { none(0),
8                     des56CbcMode(1) }
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            "The value of this object is the data encryption algorithm
13            being utilized."
14        REFERENCE
15            "IEEE 802.16 standard; Table 301"
16        ::= { wmanIfSsPkmTEKEntry 3 }
17
18        wmanIfSsPkmTEKDataAuthentAlg OBJECT-TYPE
19            SYNTAX      INTEGER { none(0) }
20            MAX-ACCESS  read-only
21            STATUS      current
22            DESCRIPTION
23                "The value of this object is the data authentication
24                algorithm being utilized."
25            REFERENCE
26                "IEEE 802.16 standard; Table 302"
27            ::= { wmanIfSsPkmTEKEntry 4 }
28
29        wmanIfSsPkmTEKEncryptAlg OBJECT-TYPE
30            SYNTAX      INTEGER { tripleDES(0),
31                                rsa1024(1) }
32            MAX-ACCESS  read-only
33            STATUS      current
34            DESCRIPTION
35                "The value of this object is the TEK key encryption
36                algorithm for this cryptographic suite capability."
37            REFERENCE
38                "IEEE 802.16 standard; Table 303"
39            ::= { wmanIfSsPkmTEKEntry 5 }
40
41        wmanIfSsPkmTEKState OBJECT-TYPE
42            SYNTAX      INTEGER { start(1),
43                                opWait(2),
44                                opReauthWait(3),
45                                operational(4),
46                                rekeyWait(5),
47                                rekeyReauthWait(6) }
48            MAX-ACCESS  read-only
49            STATUS      current
50            DESCRIPTION
51                "The value of this object is the state of the indicated TEK
52                FSM. The start(1) state indicates that FSM is in its
53                initial state."
54            ::= { wmanIfSsPkmTEKEntry 6 }

```

```
1
2 wmanIfSsPkmTEKKeySequenceNumber OBJECT-TYPE
3     SYNTAX      Integer32 (0..3)
4     MAX-ACCESS  read-only
5     STATUS      current
6     DESCRIPTION
7         "The value of this object is the most recent TEK key
8         sequence number for this TEK FSM."
9     REFERENCE
10        "IEEE 802.16 standard; 11.9.5"
11    ::= { wmanIfSsPkmTEKEntry 7 }
12
13 wmanIfSsPkmTEKExpiresOld OBJECT-TYPE
14     SYNTAX      DateAndTime
15     MAX-ACCESS  read-only
16     STATUS      current
17     DESCRIPTION
18         "The value of this object is the actual clock time for
19         expiration of the immediate predecessor of the most recent
20         TEK for this FSM. If this FSM has only one TEK, then the
21         value is the time of activation of this FSM."
22    ::= { wmanIfSsPkmTEKEntry 8 }
23
24 wmanIfSsPkmTEKExpiresNew OBJECT-TYPE
25     SYNTAX      DateAndTime
26     MAX-ACCESS  read-only
27     STATUS      current
28     DESCRIPTION
29         "The value of this object is the actual clock time for
30         expiration of the most recent TEK for this FSM."
31    ::= { wmanIfSsPkmTEKEntry 9 }
32
33 wmanIfSsPkmTEKKeyRequests OBJECT-TYPE
34     SYNTAX      Counter32
35     MAX-ACCESS  read-only
36     STATUS      current
37     DESCRIPTION
38         "The value of this object is the count of times the SS has
39         transmitted a Key Request message."
40    ::= { wmanIfSsPkmTEKEntry 10 }
41
42 wmanIfSsPkmTEKKeyReplies OBJECT-TYPE
43     SYNTAX      Counter32
44     MAX-ACCESS  read-only
45     STATUS      current
46     DESCRIPTION
47         "The value of this object is the count of times the SS has
48         received a Key Reply message, including a message whose
49         authentication failed."
50    ::= { wmanIfSsPkmTEKEntry 11 }
51
52 wmanIfSsPkmTEKKeyRejects OBJECT-TYPE
53     SYNTAX      Counter32
54     MAX-ACCESS  read-only
```



```

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

```

```

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

```

```

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

```

```

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

```

```

1      "Indicating the value of the unknown TLV."
2      ::= { wmanIfSsTrapDefinitions 6 }
3
4  wmanIfSsDynamicServiceType OBJECT-TYPE
5      SYNTAX      INTEGER {ssSfCreationReq(1),
6                  sssfCreationRsp(2),
7                  sssfCreationAck(3)
8
9                  }
10     MAX-ACCESS  read-only
11     STATUS      current
12     DESCRIPTION
13         "This object indicates the dynamic service flow
14         creation command type."
15     ::= { wmanIfSsTrapDefinitions 7 }
16
17  wmanIfSsDynamicServiceFailReason OBJECT-TYPE
18     SYNTAX      OCTET STRING
19     MAX-ACCESS  read-only
20     STATUS      current
21     DESCRIPTION
22         "This object indicates the reason why the service flow
23         cration has failed."
24     ::= { wmanIfSsTrapDefinitions 8 }
25
26  wmanIfSsRssiStatus OBJECT-TYPE
27     SYNTAX      INTEGER {ssRssiAlarm(1),
28                         ssRssiNoAlarm(2)
29
30                         }
31     MAX-ACCESS  read-only
32     STATUS      current
33     DESCRIPTION
34         "A RSSI alarm is generated if the RSSI is lower than
35         wmanIfSsRssiLowThreshold, or above
36         wmanIfSsRssiHighThreshold after alarm is restored."
37     ::= { wmanIfSsTrapDefinitions 9 }
38
39  wmanIfSsRssiStatusInfo OBJECT-TYPE
40     SYNTAX      OCTET STRING
41     MAX-ACCESS  read-only
42     STATUS      current
43     DESCRIPTION
44         "This object indicates the reason why RSSI event is
45         generated."
46     ::= { wmanIfSsTrapDefinitions 10 }
47
48  --
49  -- Common object group - containing common tables and objects to be
50  -- implemented in both Base Station and Subscriber Station
51  --
52  -- wmanIfCmnPacketCs contain the Packet Convergence Sublayer objects
53  -- that are common to both Base Station and Subscriber Station
54  wmanIfCmnPacketCs OBJECT IDENTIFIER ::= { wmanIfCommonObjects 1 }

```

```

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

```

```

1
2  wmanIfCmnClassifierRuleIndex OBJECT-TYPE
3      SYNTAX      Unsigned32 (1..4294967295)
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "An index is assigned to each classifier in the classifiers
8           table"
9      ::= { wmanIfCmnClassifierRuleEntry 1 }
10
11 wmanIfCmnCpsSfIndex OBJECT-TYPE
12     SYNTAX      Unsigned32 (1 .. 4294967295)
13     MAX-ACCESS  not-accessible
14     STATUS      current
15     DESCRIPTION
16         "A 32 bit quantity that uniquely identifies a service flow
17         to both the subscriber station and base station (BS)."

```

```

1         If the referenced parameter is not present in a classifier,
2         this object reports the value of 0."
3     REFERENCE
4         "Section 11.13.19.3.4.2 in IEEE 802.16REvd/D5-2004"
5     ::= { wmanIfCmnClassifierRuleEntry 5 }
6
7     wmanIfCmnClassifierRuleIpTosMask OBJECT-TYPE
8         SYNTAX      OCTET STRING (SIZE(1))
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            "The mask value is bitwise ANDed with TOS byte in an IP
13            packet and this value is used check range checking of
14            TosLow and TosHigh. If the referenced parameter is not
15            present in a classifier, this object reports the value
16            of 0."
17        REFERENCE
18            "Section 11.13.19.3.4.2 in IEEE 802.16REvd/D5-2004"
19        ::= { wmanIfCmnClassifierRuleEntry 6 }
20
21        wmanIfCmnClassifierRuleIpProtocol OBJECT-TYPE
22            SYNTAX      Integer32 (0..255)
23            MAX-ACCESS  read-only
24            STATUS      current
25            DESCRIPTION
26                "This object indicates the value of the IP Protocol field
27                required for IP packets to match this rule. If the
28                referenced parameter is not present in a classifier, this
29                object reports the value of 0."
30            REFERENCE
31                "Section 11.13.19.3.4.3 in IEEE 802.16REvd/D5-2004"
32            ::= { wmanIfCmnClassifierRuleEntry 7 }
33
34        wmanIfCmnClassifierRuleIpAddressType OBJECT-TYPE
35            SYNTAX      InetAddressType
36            MAX-ACCESS  read-only
37            STATUS      current
38            DESCRIPTION
39                "The type of the internet address for
40                wmanIfCmnClassifierRuleIpSourceAddr,
41                wmanIfCmnClassifierRuleIpSourceMask,
42                wmanIfCmnClassifierRuleIpDestAddr, and
43                wmanIfCmnClassifierRuleIpDestMask.
44                If the referenced parameter is not present in a classifier,
45                this object reports the value of ipv4(1)."

```



```

1         "This object specifies the value of the IP Source Address
2         required for packets to match this rule. An IP packet
3         matches the rule when the packet ip source address bitwise
4         ANDED with the wmanIfCmnClassifierRuleIpSourceMask value
5         equals the wmanIfCmnClassifierRuleIpSourceAddr value.
6         If the referenced parameter is not present n a classifier,
7         this object reports the value of 0.0.0.0."
8     REFERENCE
9         "Section 11.13.19.3.4.4 in IEEE 802.16REvd/D5-2004"
10    ::= { wmanIfCmnClassifierRuleEntry 9 }
11
12    wmanIfCmnClassifierRuleIpSourceMask OBJECT-TYPE
13        SYNTAX      InetAddress
14        MAX-ACCESS  read-only
15        STATUS      current
16        DESCRIPTION
17            "This object specifies which bits of a packet's IP Source
18            Address that are compared to match this rule. An IP packet
19            matches the rule when the packet source address bitwise
20            ANDED with the
21            wmanIfCmnClassifierRuleIpSourceMask value equals the
22            wmanIfCmnClassifierRuleIpSourceAddr value.
23            If the referenced parameter is not present in a classifier,
24            this object reports the value of 0.0.0.0."
25        REFERENCE
26            "Section 11.13.19.3.4.4 in IEEE 802.16REvd/D5-2004"
27        ::= { wmanIfCmnClassifierRuleEntry 10 }
28
29    wmanIfCmnClassifierRuleIpDestAddr OBJECT-TYPE
30        SYNTAX      InetAddress
31        MAX-ACCESS  read-only
32        STATUS      current
33        DESCRIPTION
34            "This object specifies the value of the IP Destination
35            Address required for packets to match this rule. An IP
36            packet matches the rule when the packet IP destination
37            address bitwise ANDED with the
38            wmanIfCmnClassifierRuleIpDestMask value equals the
39            wmanIfCmnClassifierRuleIpDestAddr value.
40            If the referenced parameter is not present in a
41            classifier, this object reports the value of 0.0.0.0."
42        REFERENCE
43            "Section 11.13.19.3.4.5 in IEEE 802.16REvd/D5-2004"
44        ::= { wmanIfCmnClassifierRuleEntry 11 }
45
46    wmanIfCmnClassifierRuleIpDestMask OBJECT-TYPE
47        SYNTAX      InetAddress
48        MAX-ACCESS  read-only
49        STATUS      current
50        DESCRIPTION
51            "This object specifies which bits of a packet's IP
52            Destination Address that are compared to match this rule.
53            An IP packet matches the rule when the packet destination
54            address bitwise ANDED with the

```

```

1           wmanIfCmnClassifierRuleIpDestMask value equals the
2           wmanIfCmnClassifierRuleIpDestAddr value.
3           If the referenced parameter is not present in a classifier
4           , this object reports the value of 0.0.0.0."
5   REFERENCE
6           "Section 11.13.19.3.4.5 in IEEE 802.16REvd/D5-2004"
7   ::= { wmanIfCmnClassifierRuleEntry 12 }
8
9   wmanIfCmnClassifierRuleSourcePortStart OBJECT-TYPE
10          SYNTAX      Integer32 (0..65535)
11          MAX-ACCESS  read-only
12          STATUS      current
13          DESCRIPTION
14              "This object specifies the low end inclusive range of
15              TCP/UDP source port numbers to which a packet is compared
16              . This object is irrelevant for non-TCP/UDP IP packets.
17              If the referenced parameter is not present in a
18              classifier, this object reports the value of 0."
19          REFERENCE
20              "Section 11.13.19.3.4.6 in IEEE 802.16REvd/D5-2004"
21          ::= { wmanIfCmnClassifierRuleEntry 13 }
22
23          wmanIfCmnClassifierRuleSourcePortEnd OBJECT-TYPE
24          SYNTAX      Integer32 (0..65535)
25          MAX-ACCESS  read-only
26          STATUS      current
27          DESCRIPTION
28              "This object specifies the high end inclusive range of
29              TCP/UDP source port numbers to which a packet is compared.
30              This object is irrelevant for non-TCP/UDP IP packets.
31              If the referenced parameter is not present in a classifier,
32              this object reports the value of 65535."
33          REFERENCE
34              "Section 11.13.19.3.4.6 in IEEE 802.16REvd/D5-2004"
35          ::= { wmanIfCmnClassifierRuleEntry 14 }
36
37          wmanIfCmnClassifierRuleDestPortStart OBJECT-TYPE
38          SYNTAX      Integer32 (0..65535)
39          MAX-ACCESS  read-only
40          STATUS      current
41          DESCRIPTION
42              "This object specifies the low end inclusive range of
43              TCP/UDP destination port numbers to which a packet is
44              compared. If the referenced parameter is not present
45              in a classifier, this object reports the value of 0."
46          REFERENCE
47              "Section 11.13.19.3.4.7 in IEEE 802.16REvd/D5-2004"
48          ::= { wmanIfCmnClassifierRuleEntry 15 }
49
50          wmanIfCmnClassifierRuleDestPortEnd OBJECT-TYPE
51          SYNTAX      Integer32 (0..65535)
52          MAX-ACCESS  read-only
53          STATUS      current
54          DESCRIPTION

```

```

1         "This object specifies the high end inclusive range of
2         TCP/UDP destination port numbers to which a packet is
3         compared. If the referenced parameter is not present
4         in a classifier, this object reports the value of
5         65535."
6     REFERENCE
7         "Section 11.13.19.3.4.7 in IEEE 802.16REvd/D5-2004"
8     ::= { wmanIfCmnClassifierRuleEntry 16 }
9
10    wmanIfCmnClassifierRuleDestMacAddr OBJECT-TYPE
11        SYNTAX      MacAddress
12        MAX-ACCESS  read-only
13        STATUS      current
14        DESCRIPTION
15            "An Ethernet packet matches an entry when its destination
16            MAC address bitwise ANDed with
17            wmanIfCmnClassifierRuleDestMacMask equals the value of
18            wmanIfCmnClassifierRuleDestMacAddr. If the referenced
19            parameter is not present in a classifier, this object
20            reports the value of '000000000000'H."
21        REFERENCE
22            "Section 11.13.19.3.4.8 in IEEE 802.16REvd/D5-2004"
23        ::= { wmanIfCmnClassifierRuleEntry 17 }
24
25    wmanIfCmnClassifierRuleDestMacMask OBJECT-TYPE
26        SYNTAX      MacAddress
27        MAX-ACCESS  read-only
28        STATUS      current
29        DESCRIPTION
30            "An Ethernet packet matches an entry when its destination
31            MAC address bitwise ANDed with
32            wmanIfCmnClassifierRuleDestMacMask equals the value of
33            wmanIfCmnClassifierRuleDestMacAddr. If the referenced
34            parameter is not present in a classifier, this object
35            reports the value of '000000000000'H."
36        REFERENCE
37            "Section 11.13.19.3.4.8 in IEEE 802.16REvd/D5-2004"
38        ::= { wmanIfCmnClassifierRuleEntry 18 }
39
40    wmanIfCmnClassifierRuleSourceMacAddr OBJECT-TYPE
41        SYNTAX      MacAddress
42        MAX-ACCESS  read-only
43        STATUS      current
44        DESCRIPTION
45            "An Ethernet packet matches this entry when its source
46            MAC address bitwise ANDed with
47            wmanIfCmnClassifierRuleSourceMacMask equals the value
48            of wmanIfCmnClassifierRuleSourceMacAddr. If the
49            referenced parameter is not present in a classifier,
50            this object reports the value of '000000000000'H."
51        REFERENCE
52            "Section 11.13.19.3.4.9 in IEEE 802.16REvd/D5-2004"
53        ::= { wmanIfCmnClassifierRuleEntry 19 }
54

```

```

1  wmanIfCmnClassifierRuleSourceMacMask OBJECT-TYPE
2      SYNTAX      MacAddress
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "An Ethernet packet matches an entry when its destination
7          MAC address bitwise ANDed with
8          wmanIfCmnClassifierRuleSourceMacMask equals the value of
9          wmanIfCmnClassifierRuleSourceMacAddr. If the referenced
10         parameter is not present in a classifier, this object
11         reports the value of '000000000000'H."
12     REFERENCE
13         "Section 11.13.19.3.4.9 in IEEE 802.16REvd/D5-2004"
14     ::= { wmanIfCmnClassifierRuleEntry 20 }
15
16  wmanIfCmnClassifierRuleEnetProtocolType OBJECT-TYPE
17      SYNTAX      INTEGER {none(0),
18                    ethertype(1),
19                    dsap(2)}
20      MAX-ACCESS  read-only
21      STATUS      current
22      DESCRIPTION
23          "This object indicates the format of the layer 3 protocol
24          id in the Ethernet packet. A value of none(0) means that
25          the rule does not use the layer 3 protocol type as a
26          matching criteria. A value of ethertype(1) means that the
27          rule applies only to frames which contains an EtherType
28          value. Ethertype values are contained in packets using
29          the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
30          Sub-Network Access Protocol (SNAP) encapsulation formats.
31          A value of dsap(2) means that the rule applies only to
32          frames using the IEEE802.3 encapsulation format with a
33          Destination Service Access Point (DSAP) other than 0xAA
34          (which is reserved for SNAP). If the Ethernet frame
35          contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
36          this object applies to the embedded EtherType field within
37          the 802.1P/Q header. If the referenced parameter is not
38          present in a classifier, this object reports the value of
39          0."
40     REFERENCE
41         "Section 11.13.19.3.4.10 in IEEE 802.16REvd/D5-2004"
42     ::= { wmanIfCmnClassifierRuleEntry 21 }
43
44  wmanIfCmnClassifierRuleEnetProtocol OBJECT-TYPE
45      SYNTAX      Integer32 (0..65535)
46      MAX-ACCESS  read-only
47      STATUS      current
48      DESCRIPTION
49          "If wmanIfCmnClassifierRuleEnetProtocolType is none(0),
50          this object is ignored when considering whether a packet
51          matches the current rule.
52          If wmanIfCmnClassifierRuleEnetProtocolType is ethertype(1),
53          this object gives the 16-bit value of the EtherType that
54          the packet must match in order to match the rule."

```

```

1       If wmanIfCmnClassifierRuleEnetProtocolType is dsap(2), the
2       lower 8 bits of this object's value must match the DSAP
3       byte of the packet in order to match the rule.
4       If the Ethernet frame contains an 802.1P/Q Tag header
5       (i.e. EtherType 0x8100), this object applies to the
6       embedded EtherType field within the 802.1P/Q header.
7       If the referenced parameter is not present in the
8       classifier, the value of this object is reported as 0."
9       REFERENCE
10      "Section 11.13.19.3.4.10 in IEEE 802.16REvd/D5-2004"
11      ::= { wmanIfCmnClassifierRuleEntry 22 }
12
13      wmanIfCmnClassifierRuleUserPriLow OBJECT-TYPE
14          SYNTAX      Integer32 (0..7)
15          MAX-ACCESS  read-only
16          STATUS      current
17          DESCRIPTION
18              "This object applies only to Ethernet frames using the
19              802.1P/Q tag header (indicated with EtherType 0x8100).
20              Such frames include a 16-bit Tag that contains a 3 bit
21              Priority field and a 12 bit VLAN number.
22              Tagged Ethernet packets must have a 3-bit Priority field
23              within the range of wmanIfCmnClassifierRulePriLow and
24              wmanIfCmnClassifierRulePriHigh in order to match this
25              rule.
26              If the referenced parameter is not present in the
27              classifier, the value of this object is reported as 0."
28          REFERENCE
29              "Section 11.13.19.3.4.11 in IEEE 802.16REvd/D5-2004"
30          ::= { wmanIfCmnClassifierRuleEntry 23 }
31
32      wmanIfCmnClassifierRuleUserPriHigh OBJECT-TYPE
33          SYNTAX      Integer32 (0..7)
34          MAX-ACCESS  read-only
35          STATUS      current
36          DESCRIPTION
37              "This object applies only to Ethernet frames using the
38              802.1P/Q tag header (indicated with EtherType 0x8100).
39              Such frames include a 16-bit Tag that contains a 3 bit
40              Priority field and a 12 bit VLAN number.
41              Tagged Ethernet packets must have a 3-bit Priority
42              field within the range of wmanIfCmnClassifierRulePriLow
43              and wmanIfCmnClassifierRulePriHigh in order to match
44              this rule.
45              If the referenced parameter is not present in the
46              classifier, the value of this object is reported as 7."
47          REFERENCE
48              "Section 11.13.19.3.4.11 in IEEE 802.16REvd/D5-2004"
49          ::= { wmanIfCmnClassifierRuleEntry 24 }
50
51      wmanIfCmnClassifierRuleVlanId OBJECT-TYPE
52          SYNTAX      Integer32 (0..4095)
53          MAX-ACCESS  read-only
54          STATUS      current

```

```

1      DESCRIPTION
2          "This object applies only to Ethernet frames using the
3          802.1P/Q tag header.
4          If this object's value is nonzero, tagged packets must
5          have a VLAN Identifier that matches the value in order
6          to match the rule.
7          Only the least significant 12 bits of this object's
8          value are valid.
9          If the referenced parameter is not present in the
10         classifier, the value of this object is reported as 0."
11     REFERENCE
12         "Section 11.13.19.3.4.12 in IEEE 802.16REVd/D5-2004"
13     ::= { wmanIfCmnClassifierRuleEntry 25 }
14
15     wmanIfCmnClassifierRuleState OBJECT-TYPE
16         SYNTAX      INTEGER {active(1),
17                       inactive(2)}
18         MAX-ACCESS  read-only
19         STATUS      current
20         DESCRIPTION
21             "This object indicates whether or not the classifier is
22             enabled to classify packets to a Service Flow.
23             If the referenced parameter is not present in the
24             classifier, the value of this object is reported
25             as active(1)."
26         ::= { wmanIfCmnClassifierRuleEntry 26 }
27
28     wmanIfCmnClassifierRulePkts OBJECT-TYPE
29         SYNTAX      Counter64
30         MAX-ACCESS  read-only
31         STATUS      current
32         DESCRIPTION
33             "This object counts the number of packets that have
34             been classified using this entry."
35         ::= { wmanIfCmnClassifierRuleEntry 27 }
36
37     wmanIfCmnClassifierRuleRowStatus OBJECT-TYPE
38         SYNTAX      RowStatus
39         MAX-ACCESS  read-only
40         STATUS      current
41         DESCRIPTION
42             "This object is used to create a new row or modify or
43             delete an existing row in this table.
44
45             If the implementator of this MIB has chosen not
46             to implement 'dynamic assignment' of profiles, this
47             object is not useful and should return noSuchName
48             upon SNMP request."
49         ::= { wmanIfCmnClassifierRuleEntry 28 }
50
51     --
52     -- wmanIfCmnCps contain the Common Part Sublayer objects that are
53     -- common to both Base Station and Subscriber Station
54     wmanIfCmnCps OBJECT IDENTIFIER ::= { wmanIfCommonObjects 2 }

```

```

1
2  wmanIfCmnCpsServiceFlowTable OBJECT-TYPE
3      SYNTAX      SEQUENCE OF wmanIfCmnCpsServiceFlowEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "This table contains Service Flows that are created in
8           both BS and SS."
9      ::= { wmanIfCmnCps 1 }
10
11 wmanIfCmnCpsServiceFlowEntry OBJECT-TYPE
12     SYNTAX      wmanIfCmnCpsServiceFlowEntry
13     MAX-ACCESS  not-accessible
14     STATUS      current
15     DESCRIPTION
16         "This table provides one row for each service flow, and is
17         indexed by wmanIfCmnCpsSfId. The value of wmanIfCmnCpsSfId
18         is obtained from wmanIfBsSfId."
19     INDEX      { wmanIfCmnCpsSfId }
20     ::= { wmanIfCmnCpsServiceFlowTable 1 }
21
22 wmanIfCmnCpsServiceFlowEntry ::= SEQUENCE {
23     wmanIfCmnCpsSfId                Unsigned32,
24     wmanIfCmnCpsSfCid                INTEGER,
25     wmanIfCmnCpsSfDirection          INTEGER,
26     wmanIfCmnCpsSfState              INTEGER,
27     wmanIfCmnCpsServiceClassName    DisplayString,
28     wmanIfCmnCpsTrafficPriority       INTEGER,
29     wmanIfCmnCpsMaxSustainedRate     INTEGER,
30     wmanIfCmnCpsMaxTrafficBurst      INTEGER,
31     wmanIfCmnCpsMinReservedRate     INTEGER,
32     wmanIfCmnCpsToleratedJitter     INTEGER,
33     wmanIfCmnCpsMaxLatency           INTEGER,
34     wmanIfCmnCpsFixedVsVariableSduInd INTEGER,
35     wmanIfCmnCpsSduSize              INTEGER,
36     wmanIfCmnCpsSfSchedulingType     wmanIfSfsSchedulingType,
37     wmanIfCmnCpsArqEnable            TruthValue,
38     wmanIfCmnCpsArqWindowSize        INTEGER,
39     wmanIfCmnCpsArqFragmentLifetime  INTEGER,
40     wmanIfCmnCpsArqSyncLossTimeout   INTEGER,
41     wmanIfCmnCpsArqDeliverInOrder   TruthValue,
42     wmanIfCmnCpsArqRxPurgeTimeout    INTEGER,
43     wmanIfCmnCpsFragmentLen         INTEGER,
44     wmanIfCmnCpsMinRsvdTolerableRate INTEGER,
45     wmanIfCmnCpsReqTxPolicy          BITS
46     }
47
48 wmanIfCmnCpsSfId OBJECT-TYPE
49     SYNTAX      Unsigned32 ( 1 .. 4294967295)
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "A 32 bit quantity that uniquely identifies a service flow
54         to both the subscriber station and base station (BS)."
```

```

1      ::= { wmanIfCmnCpsServiceFlowEntry 1 }
2
3  wmanIfCmnCpsSfCid OBJECT-TYPE
4      SYNTAX      INTEGER
5      MAX-ACCESS  read-only
6      STATUS      current
7      DESCRIPTION
8          "A 16 bit channel identifier to identify the connection
9          being created by DSA."
10     ::= { wmanIfCmnCpsServiceFlowEntry 2 }
11
12  wmanIfCmnCpsSfDirection OBJECT-TYPE
13      SYNTAX      INTEGER {downstream(1),
14                          upstream(2)}
15      MAX-ACCESS  read-only
16      STATUS      current
17      DESCRIPTION
18          "An attribute indicating the service flow is downstream or
19          upstream."
20     ::= { wmanIfCmnCpsServiceFlowEntry 3 }
21
22  wmanIfCmnCpsSfState OBJECT-TYPE
23      SYNTAX      INTEGER {provisioned(1),
24                          admitted(2),
25                          active(3)}
26      MAX-ACCESS  read-only
27      STATUS      current
28      DESCRIPTION
29          "wmanIfCmnCpsSfState indicates the service flow state:
30          Provisioned, AdmittedState(2), and Active service flow
31          state."
32      REFERENCE
33          "Section 6.4.13.6, in IEEE 802.16REVd/D5-2004"
34     ::= { wmanIfCmnCpsServiceFlowEntry 4 }
35
36  wmanIfCmnCpsServiceClassName OBJECT-TYPE
37      SYNTAX      DisplayString
38      MAX-ACCESS  read-only
39      STATUS      current
40      DESCRIPTION
41          "Refers to the Service Class Name"
42      REFERENCE
43          "Section 11.13.3 in IEEE 802.16REVd/D5-2004"
44     ::= { wmanIfCmnCpsServiceFlowEntry 5 }
45
46  wmanIfCmnCpsTrafficPriority OBJECT-TYPE
47      SYNTAX      INTEGER
48      MAX-ACCESS  read-only
49      STATUS      current
50      DESCRIPTION
51          "The value of this parameter specifies the priority
52          assigned to a service flow. For uplink service flows,
53          the BS should use this parameter when determining
54          precedence in request service and grant generation,

```



```
1         and the SS shall preferentially select contention
2         Request opportunities for Priority Request CIDs
3         based on this priority"
4     REFERENCE
5         "Section 11.13.7 in IEEE 802.16REVd/D5-2004"
6     ::= { wmanIfCmnCpsServiceFlowEntry 6 }
7
8     wmanIfCmnCpsMaxSustainedRate OBJECT-TYPE
9         SYNTAX      INTEGER
10        UNITS       "bps"
11        MAX-ACCESS  read-only
12        STATUS      current
13        DESCRIPTION
14            "This parameter defines the peak information rate
15            of the service. The rate is expressed in bits per
16            second and pertains to the SDUs at the input to
17            the system."
18        REFERENCE
19            "Section 11.13.8 in IEEE 802.16REVd/D5-2004"
20        ::= { wmanIfCmnCpsServiceFlowEntry 7 }
21
22    wmanIfCmnCpsMaxTrafficBurst OBJECT-TYPE
23        SYNTAX      INTEGER
24        UNITS       "byte"
25        MAX-ACCESS  read-only
26        STATUS      current
27        DESCRIPTION
28            "This parameter defines the maximum burst size that
29            must be accommodated for the service."
30        REFERENCE
31            "Section 11.13.9 in IEEE 802.16REVd/D5-2004"
32        ::= { wmanIfCmnCpsServiceFlowEntry 8 }
33
34    wmanIfCmnCpsMinReservedRate OBJECT-TYPE
35        SYNTAX      INTEGER
36        UNITS       "byte"
37        MAX-ACCESS  read-only
38        STATUS      current
39        DESCRIPTION
40            "This parameter specifies the minimum rate reserved
41            for this service flow."
42        REFERENCE
43            "Section 11.13.10 in IEEE 802.16REVd/D5-2004"
44        ::= { wmanIfCmnCpsServiceFlowEntry 9 }
45
46    wmanIfCmnCpsToleratedJitter OBJECT-TYPE
47        SYNTAX      INTEGER
48        UNITS       "millisecond"
49        MAX-ACCESS  read-only
50        STATUS      current
51        DESCRIPTION
52            "This parameter defines the Maximum delay
53            variation (jitter) for the connection."
54        REFERENCE
```

```

1      "Section 11.13.15 in IEEE 802.16REVd/D5-2004"
2      ::= { wmanIfCmnCpsServiceFlowEntry 10 }
3
4      wmanIfCmnCpsMaxLatency OBJECT-TYPE
5          SYNTAX      INTEGER
6          UNITS        "millisecond"
7          MAX-ACCESS  read-only
8          STATUS      current
9          DESCRIPTION
10             "The value of this parameter specifies the maximum
11             latency between the reception of a packet by the BS
12             or SS on its network interface and the forwarding
13             of the packet to its RF Interface."
14          REFERENCE
15             "Section 11.13.16 in IEEE 802.16REVd/D5-2004"
16          ::= { wmanIfCmnCpsServiceFlowEntry 11 }
17
18      wmanIfCmnCpsFixedVsVariableSduInd OBJECT-TYPE
19          SYNTAX      INTEGER {variableLengthSdu(0),
20                          fixedLengthSdu(1)}
21          MAX-ACCESS  read-only
22          STATUS      current
23          DESCRIPTION
24             "The value of this parameter specifies whether the SDUs
25             on the service flow are fixed-length (0) or
26             variable-length (1). The parameter is used only if
27             packing is on for the service flow. The default value
28             is 0, i.e., variable-length SDUs."
29          REFERENCE
30             "Section 11.13.15 in IEEE 802.16REVd/D5-2004"
31          DEFVAL      { 0 }
32          ::= { wmanIfCmnCpsServiceFlowEntry 12 }
33
34      wmanIfCmnCpsSduSize OBJECT-TYPE
35          SYNTAX      INTEGER
36          UNITS        "byte"
37          MAX-ACCESS  read-only
38          STATUS      current
39          DESCRIPTION
40             "The value of this parameter specifies the length of the
41             SDU for a fixed-length SDU service flow. This parameter
42             is used only if packing is on and the service flow is
43             indicated as carrying fixed-length SDUs. The default
44             value is 49 bytes, i.e., VC-switched ATM cells with PHS.
45             The parameter is relevant for both ATM and Packet
46             Convergence Sublayers."
47          REFERENCE
48             "Section 11.13.17 in IEEE 802.16REVd/D5-2004"
49          DEFVAL      { 49 }
50          ::= { wmanIfCmnCpsServiceFlowEntry 13 }
51
52      wmanIfCmnCpsSfsSchedulingType OBJECT-TYPE
53          SYNTAX      wmanIfSfsSchedulingType
54          MAX-ACCESS  read-only

```

```

1      STATUS      current
2      DESCRIPTION
3          "Specifies the upstream scheduling service used for
4          upstream service flow. If the referenced parameter
5          is not present in the corresponding 802.16 QoS
6          Parameter Set of an upstream service flow, the
7          default value of this object is bestEffort(2)."

```

```

1
2 wmanIfCmnCpsArqDeliverInOrder OBJECT-TYPE
3     SYNTAX      TruthValue
4     MAX-ACCESS  read-only
5     STATUS      current
6     DESCRIPTION
7         "Indicates whether or not data is to be delivered
8         by the receiving MAC to its client application
9         in the order in which data was handed off to the
10        originating MAC."
11    ::= { wmanIfCmnCpsServiceFlowEntry 19 }
12
13 wmanIfCmnCpsArqRxPurgeTimeout OBJECT-TYPE
14     SYNTAX      INTEGER (0 .. 65535)
15     UNITS       "10 us"
16     MAX-ACCESS  read-only
17     STATUS      current
18     DESCRIPTION
19         "Indicates the time interval the ARQ window is advanced
20         after a fragment is received. A value of 0 means
21         Infinite."
22    ::= { wmanIfCmnCpsServiceFlowEntry 20}
23
24 wmanIfCmnCpsFragmentLen OBJECT-TYPE
25     SYNTAX      INTEGER (32 .. 2040)
26     UNITS       "byte"
27     MAX-ACCESS  read-only
28     STATUS      current
29     DESCRIPTION
30         "The maximum size fragment a transmitter shall form
31         or a receiver shall expect to receive."
32    ::= { wmanIfCmnCpsServiceFlowEntry 21 }
33
34 wmanIfCmnCpsMinRsvdTolerableRate OBJECT-TYPE
35     SYNTAX      INTEGER
36     UNITS       "bps"
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
48         "Section 11.13.11 in IEEE 802.16REVd/D5-2004"
49    ::= { wmanIfCmnCpsServiceFlowEntry 22 }
50
51 wmanIfCmnCpsReqTxPolicy OBJECT-TYPE
52     SYNTAX      BITS {noBroadcastBwReq(0),
53                    reserved1(1),
54                    noPiggybackReq(2),

```

```

1           noFragmentData(3),
2           noPHS(4),
3           noSduPacking(5),
4           noCrc(6),
5           reserved2(7)}
6     MAX-ACCESS    read-only
7     STATUS        current
8     DESCRIPTION
9         "The value of this parameter provides the capability to
10        specify certain attributes for the associated service
11        flow. An attribute is enabled by setting the
12        corresponding bit position to 1."
13    REFERENCE
14        "Section 11.13.12 in IEEE 802.16REVd/D5-2004"
15    ::= { wmanIfCmnCpsServiceFlowEntry 23 }
16
17    --
18    -- wmanIfCmnBsSsConfigurationTable contains global parameters
19    -- common in BS and SS
20    wmanIfCmnBsSsConfigurationTable OBJECT-TYPE
21        SYNTAX      SEQUENCE OF WmanIfCmnBsSsConfigurationEntry
22        MAX-ACCESS  not-accessible
23        STATUS      current
24        DESCRIPTION
25            "This table provides one row for each BS sector that
26            contains the system parameters common in both SS and
27            BS. All SSSs shall have the same parameters as the BS
28            to which the SSSs are associated."
29        ::= { wmanIfCmnCps 2 }
30
31    wmanIfCmnBsSsConfigurationEntry OBJECT-TYPE
32        SYNTAX      WmanIfCmnBsSsConfigurationEntry
33        MAX-ACCESS  not-accessible
34        STATUS      current
35        DESCRIPTION
36            "This table is indexed by ifIndex, indicating BS
37            sector."
38        INDEX { ifIndex }
39        ::= { wmanIfCmnBsSsConfigurationTable 1 }
40
41    WmanIfCmnBsSsConfigurationEntry ::= SEQUENCE {
42        wmanIfCmnInvitedRangRetries    INTEGER,
43        wmanIfCmnMinislotSize          INTEGER,
44        wmanIfCmnDSxReqRetries        INTEGER,
45        wmanIfCmnDSxRespRetries       INTEGER,
46        wmanIfCmnT7Timeout             INTEGER,
47        wmanIfCmnT8Timeout             INTEGER,
48        wmanIfCmnT10Timeout            INTEGER,
49        wmanIfCmnT22Timeout            INTEGER,
50        wmanIfCmnBsSsConfigurationRowStatus  RowStatus
51    }
52
53    wmanIfCmnInvitedRangRetries OBJECT-TYPE
54        SYNTAX      INTEGER(16..65535)

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "Number of retries on inviting Ranging Requests."
5      ::= { wmanIfCmnBsSsConfigurationEntry 1 }
6
7      wmanIfCmnMinislotSize OBJECT-TYPE
8          SYNTAX      INTEGER (1..100)
9          MAX-ACCESS  read-write
10         STATUS      current
11         DESCRIPTION
12             "Size of minislot for uplink transmission. Shall be a power
13             of 2 (in units of PS)."
14         ::= { wmanIfCmnBsSsConfigurationEntry 2 }
15
16         wmanIfCmnDSxReqRetries OBJECT-TYPE
17             SYNTAX      INTEGER
18             MAX-ACCESS  read-write
19             STATUS      current
20             DESCRIPTION
21                 "Number of Timeout Retries on DSA/DSC/DSD Requests."
22             DEFVAL      { 3 }
23             ::= { wmanIfCmnBsSsConfigurationEntry 3 }
24
25         wmanIfCmnDSxRespRetries OBJECT-TYPE
26             SYNTAX      INTEGER
27             MAX-ACCESS  read-write
28             STATUS      current
29             DESCRIPTION
30                 "Number of Timeout Retries on DSA/DSC/DSD Responses."
31             DEFVAL      { 3 }
32             ::= { wmanIfCmnBsSsConfigurationEntry 4 }
33
34         wmanIfCmnT7Timeout OBJECT-TYPE
35             SYNTAX      INTEGER(0 .. 1000)
36             UNITS        "milliseconds"
37             MAX-ACCESS  read-write
38             STATUS      current
39             DESCRIPTION
40                 "Wait for DSA/DSC/DSD Response Timeout in ms."
41             ::= { wmanIfCmnBsSsConfigurationEntry 5 }
42
43         wmanIfCmnT8Timeout OBJECT-TYPE
44             SYNTAX      INTEGER(0 .. 300)
45             UNITS        "milliseconds"
46             MAX-ACCESS  read-write
47             STATUS      current
48             DESCRIPTION
49                 "Wait for DSA/DSC/DSD Acknowledge Timeout in ms."
50             ::= { wmanIfCmnBsSsConfigurationEntry 6 }
51
52         wmanIfCmnT10Timeout OBJECT-TYPE
53             SYNTAX      INTEGER(0 .. 3000)
54             UNITS        "milliseconds"

```

```

1      MAX-ACCESS  read-write
2      STATUS      current
3      DESCRIPTION
4          "Wait for Transaction End timeout in ms."
5      ::= { wmanIfCmnBsSsConfigurationEntry 7 }
6
7      wmanIfCmnT22Timeout OBJECT-TYPE
8          SYNTAX      INTEGER(0 .. 500)
9          UNITS        "milliseconds"
10         MAX-ACCESS  read-write
11         STATUS      current
12         DESCRIPTION
13             "Wait for ARQ Reset in ms."
14         ::= { wmanIfCmnBsSsConfigurationEntry 8 }
15
16         wmanIfCmnBsSsConfigurationRowStatus OBJECT-TYPE
17             SYNTAX      RowStatus
18             MAX-ACCESS  read-create
19             STATUS      current
20             DESCRIPTION
21                 "This object is used to create a new row or modify or
22                 delete an existing row in this table.
23
24                 If the implementator of this MIB has chosen not
25                 to implement 'dynamic assignment' of profiles, this
26                 object is not useful and should return noSuchName
27                 upon SNMP request."
28             ::= { wmanIfCmnBsSsConfigurationEntry 9 }
29
30         --
31         -- wmanIfCmnSsStatCounter contain the performance statistics information
32         wmanIfCmnSsStatCounter OBJECT IDENTIFIER ::= { wmanIfCmnCps 3 }
33
34         wmanIfCmnSsChMeasurementTable OBJECT-TYPE
35             SYNTAX      SEQUENCE OF wmanIfCmnSsChMeasurementEntry
36             MAX-ACCESS  not-accessible
37             STATUS      current
38             DESCRIPTION
39                 "This table contains channel measurement information
40                 for each SS. BS retrieves the channel measurement
41                 information from REP-REQ/RSP messages. This table contains
42                 channel measurement information on the downlink signal
43                 sent to SS."
44             ::= { wmanIfCmnSsStatCounter 1 }
45
46         wmanIfCmnSsChMeasurementEntry OBJECT-TYPE
47             SYNTAX      wmanIfCmnSsChMeasurementEntry
48             MAX-ACCESS  not-accessible
49             STATUS      current
50             DESCRIPTION
51                 "Each entry in the table contains RSSI and CINR
52                 signal quality measurement taken from the SS. The primary
53                 index is teh ifIndex with ifType propBWA2Mp identifying
54                 the BS sector. The primary index is the ifIndex with ifType

```

```

1         of propBWAp2Mp identifying the BS sector. wmanIfCmnSsidIndex
2         identifies the SS where the measurement taking place.
3         wmanIfCmnHistogramIndex is the index to histogram samples.
4         Since there is no time stamp in the table,
5         wmanIfCmnHistogramIndex should be increased monotonically,
6         and wraps around when it reaches the limit.
7         be maintained as FIFO to store measurement samples that
8         can be used to create RSSI and CINR histogram report.
9         When the measurement entry for a SS reaches the limit,
10        the oldest entry shall be deleted as the new entry is
11        added to the table."
12        INDEX        { ifIndex, wmanIfCmnSsidIndex,
13                    wmanIfCmnHistogramIndex }
14        ::= { wmanIfCmnSschMeasurementTable 1 }
15
16        wmanIfCmnSschMeasurementEntry ::= SEQUENCE {
17            wmanIfCmnSsidIndex            Unsigned32,
18            wmanIfCmnHistogramIndex       Unsigned32,
19            wmanIfCmnChannelNumber        INTEGER,
20            wmanIfCmnStartFrame           INTEGER,
21            wmanIfCmnDuration              INTEGER,
22            wmanIfCmnBasicReport           BITS,
23            wmanIfCmnMeanCinrReport        INTEGER,
24            wmanIfCmnStdDeviationCinrReport INTEGER,
25            wmanIfCmnMeanRssiReport        INTEGER,
26            wmanIfCmnStdDeviationRssiReport INTEGER
27        }
28
29        wmanIfCmnSsidIndex OBJECT-TYPE
30            SYNTAX      Unsigned32 (1 .. 4294967295)
31            MAX-ACCESS  read-only
32            STATUS      current
33            DESCRIPTION
34                "wmanIfCmnSsidIndex identifies the SS providing the
35                channel measurement."
36            REFERENCE
37                "Section 6.4.2.3.5 in IEEE 802.16REVd/D5-2004"
38            ::= { wmanIfCmnSschMeasurementEntry 1 }
39
40        wmanIfCmnHistogramIndex OBJECT-TYPE
41            SYNTAX      Unsigned32 (1 .. 4294967295)
42            MAX-ACCESS  read-only
43            STATUS      current
44            DESCRIPTION
45                "wmanIfBsHistogramIndex identifies the histogram samples
46                in the table for each subscriber station."
47            ::= { wmanIfCmnSschMeasurementEntry 2 }
48
49        wmanIfCmnChannelNumber OBJECT-TYPE
50            SYNTAX      INTEGER
51            MAX-ACCESS  read-only
52            STATUS      current
53            DESCRIPTION
54                "Physical channel number to be reported on."

```



```

1      REFERENCE
2          "Section 8.5.1 in IEEE 802.16REvD/D5-2004"
3      ::= { wmanIfCmnSsSchMeasurementEntry 3 }
4
5      wmanIfCmnStartFrame OBJECT-TYPE
6          SYNTAX      INTEGER
7          MAX-ACCESS  read-only
8          STATUS      current
9          DESCRIPTION
10             "Frame number in which measurement for this channel
11             started."
12         REFERENCE
13             "Section 11.12 in IEEE 802.16REvD/D5-2004"
14         ::= { wmanIfCmnSsSchMeasurementEntry 4 }
15
16         wmanIfCmnDuration OBJECT-TYPE
17             SYNTAX      INTEGER
18             MAX-ACCESS  read-only
19             STATUS      current
20             DESCRIPTION
21                 "Cumulative measurement duration on the channel in
22                 multiples of Ts. For any value exceeding 0xFFFFFFFF,
23                 report 0xFFFFFFFF."
24             REFERENCE
25                 "Section 11.12 in IEEE 802.16REvD/D5-2004"
26             ::= { wmanIfCmnSsSchMeasurementEntry 5 }
27
28         wmanIfCmnBasicReport OBJECT-TYPE
29             SYNTAX      BITS {wirelessHuman(0),
30                             unknownTransmission(1),
31                             primaryUser(2),
32                             channegNotMeasured(3)}
33             MAX-ACCESS  read-only
34             STATUS      current
35             DESCRIPTION
36                 "Bit #0: WirelessHUMAN detected on the channel
37                 Bit #1: Unknown transmissions detected on the channel
38                 Bit #2: Primary User detected on the channel
39                 Bit #3: Unmeasured. Channel not measured"
40             REFERENCE
41                 "Section 11.12 in IEEE 802.16REvD/D5-2004"
42             ::= { wmanIfCmnSsSchMeasurementEntry 6 }
43
44         wmanIfCmnMeanCinrReport OBJECT-TYPE
45             SYNTAX      INTEGER
46             MAX-ACCESS  read-only
47             STATUS      current
48             DESCRIPTION
49                 "Mean CINR report."
50             REFERENCE
51                 "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
52                 802.16REvD/D5-2004"
53             ::= { wmanIfCmnSsSchMeasurementEntry 7 }
54

```

```

1  wmanIfCmnStdDeviationCinrReport OBJECT-TYPE
2      SYNTAX      INTEGER
3      MAX-ACCESS  read-only
4      STATUS      current
5      DESCRIPTION
6          "Standard deviation CINR report."
7      REFERENCE
8          "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
9              802.16REVD/D5-2004"
10     ::= { wmanIfCmnSsSchMeasurementEntry 8 }
11
12  wmanIfCmnMeanRssiReport OBJECT-TYPE
13      SYNTAX      INTEGER
14      MAX-ACCESS  read-only
15      STATUS      current
16      DESCRIPTION
17          "Mean RSSI report."
18      REFERENCE
19          "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
20              802.16REVD/D5-2004"
21     ::= { wmanIfCmnSsSchMeasurementEntry 9 }
22
23  wmanIfCmnStdDeviationRssiReport OBJECT-TYPE
24      SYNTAX      INTEGER
25      MAX-ACCESS  read-only
26      STATUS      current
27      DESCRIPTION
28          "Standard deviation RSSI report."
29      REFERENCE
30          "Section 8.2.2, 8.3.8, 8.4.1, 11.12 in IEEE
31              802.16REVD/D5-2004"
32     ::= { wmanIfCmnSsSchMeasurementEntry 10 }
33
34  -- Common PKM group
35  -- wmanIfCmnPkmObjects contain the Privacy Sublayer objects that are
36  -- common to both Base Station and Subscriber Station
37  wmanIfCmnPkmObjects OBJECT IDENTIFIER ::= { wmanIfCommonObjects 3 }
38
39  --
40  -- Table wmanIfCmnCryptoSuiteTable
41  --
42  wmanIfCmnCryptoSuiteTable OBJECT-TYPE
43      SYNTAX      SEQUENCE OF wmanIfCmnCryptoSuiteEntry
44      MAX-ACCESS  not-accessible
45      STATUS      current
46      DESCRIPTION
47          "This table describes the PKM cryptographic suite
48              capabilities for each SS or BS wireless interface."
49     ::= { wmanIfCmnPkmObjects 1 }
50
51  wmanIfCmnCryptoSuiteEntry OBJECT-TYPE
52      SYNTAX      wmanIfCmnCryptoSuiteEntry
53      MAX-ACCESS  not-accessible
54      STATUS      current

```

```

1      DESCRIPTION
2          "Each entry contains the cryptographic suite pair that SS
3          or BS supports."
4      INDEX      { ifIndex, wmanIfCmnCryptoSuiteIndex }
5      ::= { wmanIfCmnCryptoSuiteTable 1 }
6
7      wmanIfCmnCryptoSuiteEntry ::= SEQUENCE {
8          wmanIfCmnCryptoSuiteIndex      Integer32,
9          wmanIfCmnCryptoSuiteDataEncryptAlg  INTEGER,
10         wmanIfCmnCryptoSuiteDataAuthentAlg  INTEGER,
11         wmanIfCmnCryptoSuiteTEKEncryptAlg  INTEGER
12     }
13
14     wmanIfCmnCryptoSuiteIndex OBJECT-TYPE
15         SYNTAX      Integer32 (1 .. 1000)
16         MAX-ACCESS  not-accessible
17         STATUS      current
18         DESCRIPTION
19             "The index for a cryptographic suite row."
20         ::= { wmanIfCmnCryptoSuiteEntry 1 }
21
22     wmanIfCmnCryptoSuiteDataEncryptAlg OBJECT-TYPE
23         SYNTAX      INTEGER { none(0),
24                         des56CbcMode(1),
25                         aesCcmMode(2) }
26         MAX-ACCESS  read-only
27         STATUS      current
28         DESCRIPTION
29             "The value of this object is the data encryption algorithm
30             for this cryptographic suite capability."
31         REFERENCE
32             "IEEE 802.16 standard; Table 373"
33         ::= { wmanIfCmnCryptoSuiteEntry 2 }
34
35     wmanIfCmnCryptoSuiteDataAuthentAlg OBJECT-TYPE
36         SYNTAX      INTEGER { none(0) }
37         MAX-ACCESS  read-only
38         STATUS      current
39         DESCRIPTION
40             "The value of this object is the data authentication
41             algorithm for this cryptographic suite capability."
42         REFERENCE
43             "IEEE 802.16 standard; Table 302"
44         ::= { wmanIfCmnCryptoSuiteEntry 3 }
45
46     wmanIfCmnCryptoSuiteTEKEncryptAlg OBJECT-TYPE
47         SYNTAX      INTEGER {tripleDES128Key(1),
48                         rsa1024Key(2),
49                         aes128Key(3) }
50         MAX-ACCESS  read-only
51         STATUS      current
52         DESCRIPTION
53             "The value of this object is the TEK key encryption
54             algorithm for this cryptographic suite capability."

```

```

1      REFERENCE
2          "IEEE 802.16 standard; Table 375"
3      ::= { wmanIfCmnCryptoSuiteEntry 4 }
4
5      --
6      -- wmanIfCmnOfdmPhy contain the OFDM PHY objects that are common to both
7      -- Base Station and Subscriber Station. When the objects are implemented
8      -- in the BS, they should have the read-write access. When the objects
9      -- are implemented the SS, they should have the read-only access.
10     --
11     wmanIfCmnOfdmPhy OBJECT IDENTIFIER ::= { wmanIfCommonObjects 4 }
12
13     wmanIfCmnOfdmUplinkChannelTable OBJECT-TYPE
14         SYNTAX      SEQUENCE OF wmanIfCmnOfdmUplinkChannelEntry
15         MAX-ACCESS  not-accessible
16         STATUS      current
17         DESCRIPTION
18             "This table contains UCD channel attributes, defining the
19             transmission characteristics of uplink channels"
20         REFERENCE
21             "Section 11.3.1, table 276 and 279, in IEEE
22             802.16REVd/D5-2004"
23         ::= { wmanIfCmnOfdmPhy 1 }
24
25     wmanIfCmnOfdmUplinkChannelEntry OBJECT-TYPE
26         SYNTAX      wmanIfCmnOfdmUplinkChannelEntry
27         MAX-ACCESS  not-accessible
28         STATUS      current
29         DESCRIPTION
30             "This table provides one row for each uplink channel of
31             multi-sector BS, and is indexed by BS ifIndex. An entry
32             in this table exists for each ifEntry of BS with an
33             ifType of propBWA2Mp.
34             The objects in each entry will be implemented as
35             read-create in BS and read-only in SS."
36         INDEX { ifIndex }
37         ::= { wmanIfCmnOfdmUplinkChannelTable 1 }
38
39     wmanIfCmnOfdmUplinkChannelEntry ::= SEQUENCE {
40         wmanIfCmnOfdmCtBasedResvTimeout      INTEGER,
41         wmanIfCmnOfdmBwReqOppSize           INTEGER,
42         wmanIfCmnOfdmRangReqOppSize         INTEGER,
43         wmanIfCmnOfdmUplinkCenterFreq      INTEGER,
44         wmanIfCmnOfdmSubChReqRegionFull    INTEGER,
45         wmanIfCmnOfdmSubChFocusCtCode      INTEGER,
46         wmanIfCmnOfdmUplinkChannelRowStatus  RowStatus
47     }
48
49     wmanIfCmnOfdmCtBasedResvTimeout OBJECT-TYPE
50         SYNTAX      INTEGER (1..255)
51         MAX-ACCESS  read-only
52         STATUS      current
53         DESCRIPTION
54             "The number of UL-MAPS to receive before contention-based

```

```

1         reservation is attempted again for the same connection."
2     REFERENCE
3         "Section 11.3.1, table 276, in IEEE 802.16REVd/D5-2004"
4     ::= { wmanIfCmnOfdmUplinkChannelEntry 1 }
5
6     wmanIfCmnOfdmBwReqOppSize OBJECT-TYPE
7         SYNTAX      INTEGER (1..65535)
8         UNITS       "PS"
9         MAX-ACCESS  read-only
10        STATUS      current
11        DESCRIPTION
12            " Size (in units of PS) of PHY payload that SS may use to
13            format and transmit a bandwidth request message in a
14            contention request opportunity. The value includes all
15            PHY overhead as well as allowance for the MAC data the
16            message may hold."
17        REFERENCE
18            "Section 11.3.1, table 276, in IEEE 802.16REVd/D5-2004"
19        ::= { wmanIfCmnOfdmUplinkChannelEntry 2 }
20
21        wmanIfCmnOfdmRangReqOppSize OBJECT-TYPE
22            SYNTAX      INTEGER (1..65535)
23            UNITS       "PS"
24            MAX-ACCESS  read-only
25            STATUS      current
26            DESCRIPTION
27                " Size (in units of PS) of PHY payload that SS may use to
28                format and transmit a RNG-REQ message in a contention
29                request opportunity. The value includes all PHY overhead
30                as well as allowance for the MAC data the message may
31                hold and the maximum SS/BS roundtrip propagation delay."
32            REFERENCE
33                "Section 11.3.1, table 276, in IEEE 802.16REVd/D5-2004"
34            ::= { wmanIfCmnOfdmUplinkChannelEntry 3 }
35
36        wmanIfCmnOfdmUplinkCenterFreq OBJECT-TYPE
37            SYNTAX      INTEGER
38            UNITS       "KHZ"
39            MAX-ACCESS  read-only
40            STATUS      current
41            DESCRIPTION
42                " Uplink center frequency (KHz)"
43            REFERENCE
44                "Section 11.3.1, table 276, in IEEE 802.16REVd/D5-2004"
45            ::= { wmanIfCmnOfdmUplinkChannelEntry 4 }
46
47        wmanIfCmnOfdmSubChReqRegionFull OBJECT-TYPE
48            SYNTAX      INTEGER {oneSubchannel(0),
49                            twoSubchannels(1),
50                            fourSubchannels(2),
51                            eightSubchannels(3),
52                            sixteenSubchannels(4)}
53            MAX-ACCESS  read-only
54            STATUS      current

```

```

1      DESCRIPTION
2          "Bits 0 - 2 Number of subchannels used by each transmit
3          opportunity when REQ Region-Full is allocated in
4          subchannelization region, per the following enumeration:
5              0: 1 Subchannel.
6              1: 2 Subchannels.
7              2: 4 Subchannels.
8              3: 8 Subchannels.
9              4: 16 Subchannels.
10             5-7: Shall not be used.
11             Bits 3 - 7: Number of OFDM symbols used by each transmit
12             opportunity when REQ Region-Full is allocated in
13             subchannelization region.
14      REFERENCE
15          Section 11.3.1, table 279, in IEEE 802.16REVd/D5-2004"
16      ::= { wmanIfCmnOfdmUplinkChannelEntry 5 }
17
18      wmanIfCmnOfdmSubChFocusCtCode OBJECT-TYPE
19          SYNTAX      INTEGER (0..8)
20          MAX-ACCESS  read-only
21          STATUS      current
22          DESCRIPTION
23              "Number of contention codes (CSE) that shall only be used to
24              request a subchannelized allocation. Default value 0.
25              Allowed values 0-8."
26          REFERENCE
27              "Section 11.3.1, table 279, in IEEE 802.16REVd/D5-2004"
28          DEFVAL      { 0 }
29      ::= { wmanIfCmnOfdmUplinkChannelEntry 6 }
30
31      wmanIfCmnOfdmUplinkChannelRowStatus OBJECT-TYPE
32          SYNTAX      RowStatus
33          MAX-ACCESS  read-only
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      ::= { wmanIfCmnOfdmUplinkChannelEntry 7 }
44
45      wmanIfCmnOfdmDownlinkChannelTable OBJECT-TYPE
46          SYNTAX      SEQUENCE OF wmanIfCmnOfdmDownlinkChannelEntry
47          MAX-ACCESS  not-accessible
48          STATUS      current
49          DESCRIPTION
50              "This table contains DCD channel attributes, defining the
51              transmission characteristics of downlink channels"
52          REFERENCE
53              "Section 11.4.1, Table 286, in IEEE 802.16REVd/D5-2004"
54      ::= { wmanIfCmnOfdmPhy 2 }

```

```

1
2  wmanIfCmnOfdmDownlinkChannelEntry OBJECT-TYPE
3      SYNTAX      wmanIfCmnOfdmDownlinkChannelEntry
4      MAX-ACCESS  not-accessible
5      STATUS      current
6      DESCRIPTION
7          "This table provides one row for each downlink channel of
8          multi-sector BS, and is indexed by BS ifIndex. An entry
9          in this table exists for each ifEntry of BS with an
10         iftype of propBWA2Mp.
11         The objects in each entry will be implemented as
12         read-create in BS and read-only in SS."
13     INDEX { ifIndex }
14     ::= { wmanIfCmnOfdmDownlinkChannelTable 1 }
15
16  wmanIfCmnOfdmDownlinkChannelEntry ::= SEQUENCE {
17      wmanIfCmnOfdmBSEIRP          INTEGER,
18      wmanIfCmnOfdmChannelNumber  INTEGER,
19      wmanIfCmnOfdmTTG             INTEGER,
20      wmanIfCmnOfdmRTG            INTEGER,
21      wmanIfCmnOfdmInitRngMaxRSS  INTEGER,
22      wmanIfCmnOfdmChSwitchFrameNmr  INTEGER,
23      wmanIfCmnOfdmDownlinkCenterFreq  INTEGER,
24      wmanIfCmnOfdmBsId           OCTET STRING,
25      wmanIfCmnOfdmMacVersion      INTEGER,
26      wmanIfCmnOfdmFrameDurationCode  INTEGER,
27      wmanIfCmnOfdmFrameNumber     INTEGER,
28      wmanIfCmnOfdmDownlinkChannelRowStatus  RowStatus
29  }
30
31  wmanIfCmnOfdmBSEIRP OBJECT-TYPE
32      SYNTAX      INTEGER (0..65535)
33      UNITS       "dbm"
34      MAX-ACCESS  read-only
35      STATUS      current
36      DESCRIPTION
37          " Signed in units of 1 dBm."
38      REFERENCE
39          "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
40      ::= { wmanIfCmnOfdmDownlinkChannelEntry 1 }
41
42  wmanIfCmnOfdmChannelNumber OBJECT-TYPE
43      SYNTAX      INTEGER (1..255)
44      MAX-ACCESS  read-only
45      STATUS      current
46      DESCRIPTION
47          " Downlink channel number as defined in 8.5.
48          Used for license-exempt operation only."
49      REFERENCE
50          "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
51      ::= { wmanIfCmnOfdmDownlinkChannelEntry 2 }
52
53  wmanIfCmnOfdmTTG OBJECT-TYPE
54      SYNTAX      INTEGER (0..255)

```

```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          " Transmit / Receive Transition Gap."
5      REFERENCE
6          "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
7      ::= { wmanIfCmnOfdmDownlinkChannelEntry 3 }
8
9      wmanIfCmnOfdmRTG OBJECT-TYPE
10     SYNTAX      INTEGER (0..255)
11     MAX-ACCESS  read-only
12     STATUS      current
13     DESCRIPTION
14         " Receive / Transmit Transition Gap."
15     REFERENCE
16         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
17     ::= { wmanIfCmnOfdmDownlinkChannelEntry 4 }
18
19     wmanIfCmnOfdmInitRngMaxRSS OBJECT-TYPE
20     SYNTAX      INTEGER (0..65535)
21     UNITS       "dbm"
22     MAX-ACCESS  read-only
23     STATUS      current
24     DESCRIPTION
25         " Initial Ranging Max. Received Signal Strength at BS
26         Signed in units of 1 dBm."
27     REFERENCE
28         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
29     ::= { wmanIfCmnOfdmDownlinkChannelEntry 5 }
30
31     wmanIfCmnOfdmChSwitchFrameNmr OBJECT-TYPE
32     SYNTAX      INTEGER (0..16777215)
33     MAX-ACCESS  read-only
34     STATUS      current
35     DESCRIPTION
36         " Channel switch frame number as defined in 6.4.14.7,
37         Used for license-exempt operation only."
38     REFERENCE
39         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
40     ::= { wmanIfCmnOfdmDownlinkChannelEntry 6 }
41
42     wmanIfCmnOfdmDownlinkCenterFreq OBJECT-TYPE
43     SYNTAX      INTEGER
44     UNITS       "KHZ"
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         " Downlink center frequency (kHz)."
49     REFERENCE
50         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
51     ::= { wmanIfCmnOfdmDownlinkChannelEntry 7 }
52
53     wmanIfCmnOfdmBsId OBJECT-TYPE
54     SYNTAX      OCTET STRING (SIZE(6))

```



```

1      MAX-ACCESS  read-only
2      STATUS      current
3      DESCRIPTION
4          " Base station ID."
5      REFERENCE
6          "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
7      ::= { wmanIfCmnOfdmDownlinkChannelEntry 8 }
8
9  wmanIfCmnOfdmMacVersion OBJECT-TYPE
10     SYNTAX      INTEGER {ieee802Dot16-2001(1),
11                 ieee802Dot16c-2002(2),
12                 ieee802Dot16a-2003(3),
13                 ieee802Dot16-2004(4)}
14     MAX-ACCESS  read-only
15     STATUS      current
16     DESCRIPTION
17         " This parameter specifies the version of 802.16 to which
18         the message originator conforms."
19     REFERENCE
20         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
21     ::= { wmanIfCmnOfdmDownlinkChannelEntry 9 }
22
23  wmanIfCmnOfdmFrameDurationCode OBJECT-TYPE
24     SYNTAX      INTEGER (0..6)
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28         " The duration of the frame. The frame duration code
29         values are specified in Table 230."
30     REFERENCE
31         "Section 11.4.1, table 230, in IEEE 802.16/2004"
32     ::= { wmanIfCmnOfdmDownlinkChannelEntry 10 }
33
34  wmanIfCmnOfdmFrameNumber OBJECT-TYPE
35     SYNTAX      INTEGER (0..16777215)
36     MAX-ACCESS  read-only
37     STATUS      current
38     DESCRIPTION
39         " The number of frame containing the DCD message."
40     REFERENCE
41         "Section 11.4.1, table 286, in IEEE 802.16REVd/D5-2004"
42     ::= { wmanIfCmnOfdmDownlinkChannelEntry 11 }
43
44  wmanIfCmnOfdmDownlinkChannelRowStatus OBJECT-TYPE
45     SYNTAX      RowStatus
46     MAX-ACCESS  read-only
47     STATUS      current
48     DESCRIPTION
49         "This object is used to create a new row or modify or
50         delete an existing row in this table.
51
52         If the implementator of this MIB has chosen not
53         to implement 'dynamic assignment' of profiles, this
54         object is not useful and should return noSuchName

```

```

1      upon SNMP request."
2      ::= { wmanIfCmnOfdmDownlinkChannelEntry 12 }
3
4  wmanIfCmnOfdmUcdBurstProfileTable OBJECT-TYPE
5      SYNTAX      SEQUENCE OF wmanIfCmnOfdmUcdBurstProfileEntry
6      MAX-ACCESS  not-accessible
7      STATUS      current
8      DESCRIPTION
9          "This table contains UCD burst profiles for each uplink
10         channel"
11     REFERENCE
12         "Section 11.3.1.1, table 281 and 284, in IEEE
13         802.16REVd/D5-2004"
14     ::= { wmanIfCmnOfdmPhy 3 }
15
16  wmanIfCmnOfdmUcdBurstProfileEntry OBJECT-TYPE
17     SYNTAX      wmanIfCmnOfdmUcdBurstProfileEntry
18     MAX-ACCESS  not-accessible
19     STATUS      current
20     DESCRIPTION
21         "This table provides one row for each UCD burst profile.
22         This table is double indexed. The primary index is an
23         ifIndex with an ifType of propBWAp2Mp. The secondary index
24         is wmanIfCmnOfdmOfdmUcdBurstProfIndex.
25         The objects in each entry will be implemented as
26         read-create in BS and read-only in SS."
27     INDEX { ifIndex, wmanIfCmnOfdmOfdmUcdBurstProfIndex }
28     ::= { wmanIfCmnOfdmUcdBurstProfileTable 1 }
29
30  wmanIfCmnOfdmUcdBurstProfileEntry ::= SEQUENCE {
31      wmanIfCmnOfdmOfdmUcdBurstProfIndex      INTEGER,
32      wmanIfCmnOfdmUiucValue                  INTEGER,
33      wmanIfCmnOfdmUplinkFrequency            INTEGER,
34      wmanIfCmnOfdmUcdFecCodeType             INTEGER,
35      wmanIfCmnOfdmFocusCtPowerBoost          INTEGER,
36      wmanIfCmnOfdmUcdBurstProfileRowStatus   RowStatus
37  }
38
39  wmanIfCmnOfdmOfdmUcdBurstProfIndex OBJECT-TYPE
40     SYNTAX      INTEGER (5 .. 12)
41     MAX-ACCESS  not-accessible
42     STATUS      current
43     DESCRIPTION
44         "ifIndex and wmanIfCmnOfdmOfdmUcdBurstProfIndex uniquely
45         identify an entry in the wmanIfCmnOfdmUcdBurstProfileTable."
46     ::= { wmanIfCmnOfdmUcdBurstProfileEntry 1 }
47
48  wmanIfCmnOfdmUiucValue OBJECT-TYPE
49     SYNTAX      INTEGER (5..12)
50     MAX-ACCESS  read-only
51     STATUS      current
52     DESCRIPTION
53         "The Uplink Interval Usage Code indicates the uplink burst
54         profile in the UCD message."

```

```

1      REFERENCE
2          "Section 8.3.6.3.1, in IEEE 802.16/2004"
3      ::= { wmanIfCmnOfdmUcdBurstProfileEntry 2 }
4
5      wmanIfCmnOfdmUplinkFrequency OBJECT-TYPE
6          SYNTAX      INTEGER
7          UNITS        "KHZ"
8          MAX-ACCESS  read-only
9          STATUS       current
10         DESCRIPTION
11             "Uplink Frequency (kHz)."

```

```

1           17 = 64-QAM (CTC) 2/3
2           18 = 64-QAM (CTC) 3/4
3           19 - 255 Reserved."
4     REFERENCE
5         "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D5-2004"
6     ::= { wmanIfCmnOfdmUcdBurstProfileEntry 4 }
7
8     wmanIfCmnOfdmFocusCtPowerBoost OBJECT-TYPE
9         SYNTAX      INTEGER
10        MAX-ACCESS  read-only
11        STATUS      current
12        DESCRIPTION
13            "The power boost in dB of focused contention carriers, as
14            described in 8.3.6.3.3."
15        REFERENCE
16            "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D5-2004"
17        ::= { wmanIfCmnOfdmUcdBurstProfileEntry 5 }
18
19     wmanIfCmnOfdmUcdBurstProfileRowStatus OBJECT-TYPE
20        SYNTAX      RowStatus
21        MAX-ACCESS  read-only
22        STATUS      current
23        DESCRIPTION
24            "This object is used to create a new row or modify or
25            delete an existing row in this table.
26
27            If the implementator of this MIB has chosen not
28            to implement 'dynamic assignment' of profiles, this
29            object is not useful and should return noSuchName
30            upon SNMP request."
31        ::= { wmanIfCmnOfdmUcdBurstProfileEntry 6 }
32
33     wmanIfCmnOfdmDcdBurstProfileTable OBJECT-TYPE
34        SYNTAX      SEQUENCE OF WmanIfOfdmDcdBurstProfileEntry
35        MAX-ACCESS  not-accessible
36        STATUS      current
37        DESCRIPTION
38            "This table provides one row for each DCD burst profile.
39            This table is double indexed. The primary index is an
40            ifIndex with an ifType of propBWA2Mp. The secondary
41            index is wmanIfCmnOfdmOfdmDcdBurstProfIndex"
42        ::= { wmanIfCmnOfdmPhy 4 }
43
44
45     wmanIfCmnOfdmDcdBurstProfileEntry OBJECT-TYPE
46        SYNTAX      WmanIfOfdmDcdBurstProfileEntry
47        MAX-ACCESS  not-accessible
48        STATUS      current
49        DESCRIPTION
50            "This table provides one row for each DCD burst profile.
51            This table is double indexed. The primary index is an
52            ifIndex with an ifType of propBWA2Mp. The secondary index
53            is wmanIfCmnOfdmDcdBurstProfIndex.
54            The objects in each entry will be implemented as

```

```

1         read-create in BS and read-only in SS."
2     INDEX { ifIndex, wmanIfCmnOfdmDcdBurstProfIndex }
3     ::= { wmanIfCmnOfdmDcdBurstProfileTable 1 }
4
5     wmanIfOfdmDcdBurstProfileEntry ::= SEQUENCE {
6         wmanIfCmnOfdmDcdBurstProfIndex    INTEGER,
7         wmanIfCmnOfdmDiucValue            INTEGER,
8         wmanIfCmnOfdmDownlinkFrequency    INTEGER,
9         wmanIfCmnOfdmDcdFecCodeType       INTEGER,
10        wmanIfCmnOfdmDiucMandatoryExitThresh    INTEGER,
11        wmanIfCmnOfdmDiucMinEntryThresh    INTEGER,
12        wmanIfCmnOfdmTcsEnable             INTEGER,
13        wmanIfCmnOfdmDcdBurstProfileRowStatus    RowStatus
14    }
15
16    wmanIfCmnOfdmDcdBurstProfIndex OBJECT-TYPE
17        SYNTAX      INTEGER (1 .. 11)
18        MAX-ACCESS  not-accessible
19        STATUS      current
20        DESCRIPTION
21            "ifIndex and wmanIfCmnOfdmDcdBurstProfIndex uniquely
22             identify an entry in the wmanIfCmnOfdmDcdBurstProfileTable."
23        ::= { wmanIfCmnOfdmDcdBurstProfileEntry 1 }
24
25    wmanIfCmnOfdmDiucValue OBJECT-TYPE
26        SYNTAX      INTEGER (1..11)
27        MAX-ACCESS  read-only
28        STATUS      current
29        DESCRIPTION
30            "The Downlink Interval Usage Code indicates the downlink
31             burst profile in the UCD message."
32        REFERENCE
33            "Section 8.3.6.3.1, in IEEE 802.16/2004"
34        ::= { wmanIfCmnOfdmDcdBurstProfileEntry 2 }
35
36    wmanIfCmnOfdmDownlinkFrequency OBJECT-TYPE
37        SYNTAX      INTEGER
38        UNITS       "KHZ"
39        MAX-ACCESS  read-only
40        STATUS      current
41        DESCRIPTION
42            "Downlink Frequency (kHz)."
43        REFERENCE
44            "Section 11.4.1, table 287, in IEEE 802.16REVd/D5-2004"
45        ::= { wmanIfCmnOfdmDcdBurstProfileEntry 3 }
46
47    wmanIfCmnOfdmDcdFecCodeType OBJECT-TYPE
48        SYNTAX      INTEGER {qpskRsCc1-2(0),
49                            qpskRsCc3-4(1),
50                            sixteenQamRsCc1-2(2),
51                            sixteenQamRsCc3-4(3),
52                            sixtyFourQamRsCc2-3(4),
53                            sixtyFourQamRsCc3-4(5),
54                            qpskBtc1-2(6),

```

```

1          qpskBtc3-4(7),
2          sixteenQamBtc3-4(8),
3          sixteenQamBtc4-5(9),
4          sixtyFourQamBtc2-3or5-8(10),
5          sixtyFourQamBtc5-6or4-5(11),
6          qpskCtc1-2(12),
7          qpskCtc2-3(13),
8          qpskCtc3-4(14),
9          sixteenQamCtc1-2(16),
10         sixteenQamCtc3-4(17),
11         sixtyFourQamCtc3-4(18)}
12     MAX-ACCESS    read-only
13     STATUS        current
14     DESCRIPTION
15         " 0= QPSK (RS+CC) 1/2
16         1= QPSK (RS+CC) 3/4
17         2= 16-QAM (RS+CC) 1/2
18         3= 16-QAM (RS+CC) 3/4
19         4= 64-QAM (RS+CC) 2/3
20         5= 64-QAM (RS+CC) 3/4
21         6= QPSK (BTC) 1/2
22         7= QPSK (BTC) 3/4
23         8= 16-QAM (BTC) 3/5
24         9= 16-QAM (BTC) 4/5
25         10 = 64-QAM (BTC) 2/3 or 5/8
26         11 = 64-QAM (BTC) 5/6 or 4/5
27         12 = QPSK (CTC) 1/2
28         13 = QPSK (CTC) 2/3
29         14 = QPSK (CTC) 3/4
30         15 = 16-QAM (CTC) 1/2
31         16 = 16-QAM (CTC) 3/4
32         17 = 64-QAM (CTC) 2/3
33         18 = 64-QAM (CTC) 3/4
34         19 - 255 Reserved."
35     REFERENCE
36         "Section 11.4.1, table 290, in IEEE 802.16REVd/D5-2004"
37     ::= { wmanIfCmnOfdmDcdBurstProfileEntry 4 }
38
39     wmanIfCmnOfdmDiucMandatoryExitThresh OBJECT-TYPE
40         SYNTAX          INTEGER (0..255)
41         MAX-ACCESS      read-only
42         STATUS          current
43         DESCRIPTION
44             "DIUC mandatory exit threshold: 0 - 63.75 dB CINR at or
45             below where this DIUC can no longer be used and where this
46             change to a more robust DIUC is required, in 0.25 dB units."
47         REFERENCE
48             "Section 11.4.1, table 290, in IEEE 802.16REVd/D5-2004"
49         ::= { wmanIfCmnOfdmDcdBurstProfileEntry 5 }
50
51     wmanIfCmnOfdmDiucMinEntryThresh OBJECT-TYPE
52         SYNTAX          INTEGER (0..255)
53         MAX-ACCESS      read-only
54         STATUS          current

```

```
1      DESCRIPTION
2          "DIUC minimum entry threshold: 0 - 63.75 dB The minimum CINR
3          required to start using this DIUC when changing from a more
4          robust DIUC is required, in 0.25 dB units."
5      REFERENCE
6          "Section 11.4.1, table 290, in IEEE 802.16REVd/D5-2004"
7      ::= { wmanIfCmnOfdmDcdBurstProfileEntry 6 }
8
9      wmanIfCmnOfdmTcsEnable OBJECT-TYPE
10     SYNTAX      INTEGER {tcsDisabled (0),
11                 tcsEnabled (1)}
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "Indicates whether Transmission Convergence Sublayer
16         is enabled or disabled."
17     REFERENCE
18         "Section 11.4.1, table 360, in IEEE 802.16/2004"
19     ::= { wmanIfCmnOfdmDcdBurstProfileEntry 7 }
20
21     wmanIfCmnOfdmDcdBurstProfileRowStatus OBJECT-TYPE
22     SYNTAX      RowStatus
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "This object is used to create a new row or modify or
27         delete an existing row in this table.
28
29         If the implementator of this MIB has chosen not
30         to implement 'dynamic assignment' of profiles, this
31         object is not useful and should return noSuchName
32         upon SNMP request."
33     ::= { wmanIfCmnOfdmDcdBurstProfileEntry 8 }
34
35
36     END
37
38
39
```

