Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**MAC and PHY MIB for WirelessMAN and WirelessHUMAN BS and SS**
Date Submitted	**2004-03-12**

Source(s)	Joey Chou Intel Corporation	[mailto:joey.chou@intel.com]
	Russ Reynolds Alex Raji Proxim Corporation	[mailto:RReynolds@proxim.com] [mailto:adr@telecominnovations.com]
	Vladimir Yanover Alvarion	[mailto: Vladimir.Yanover@alvarion.com]
	Shlomi Eini Airspan	[mailto:seini@Airspan.com]
	Radu Selea Redline Communications	[mailto:radu@redlinecommunications.com]

Re:	
Abstract	IEEE 802.16 working group defines WirelessMAN and WirelessHUMAN air interface specifications for the development of standard based Base Station (BS) and Subscriber Station (SS) to provide broadband wireless services to Metropolitan Area Networks (MANs). This contribution defines the 802.16 MIB for MAC and PHY layers to achieve management interoperability and provide the remote management capability that are urgently needed for massive WirelessMAN and WirelessHUMAN deployment by carriers.
Purpose	Adoption
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	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. Introduction *5*
1.1 Scope **5**
1.2 References **5**
2. BWA Network Management Reference Model *6*
3. Relationship with Interface MIB *7*
3.1 MIB-2 Integration **7**
3.2 Usage of MIB-II Tables **7**
4. Service Flow Creation *9*
5. 802.16 MIB Structure *10*
5.1 wmanIfBsObjects **11**
5.1.1 wmanIfBsSystem **11**
5.1.1.1 wmanIfBsInventoryInfoTable **11**
5.1.1.2 wmanIfBsRegisteredSsTable **11**
5.1.2 wmanIfBsPacketCs **11**
5.1.2.1 wmanIfBsPreProvisionedSfTable **11**
5.1.2.2 wmanIfBsServiceClassTable **12**
5.1.3 wmanIfBsCps **12**
5.1.3.1 wmanIfBsConfigurationTable **12**
5.1.3.2 wmanIfBsStatisticsCountersTable **13**
5.1.4 wmanIfBsPkm **13**
5.1.4.1 wmanIfBsPkmBaselineTable **13**
5.1.4.2 wmanIfBsAuthTable **13**
5.1.4.3 wmanIfBsTekTable **13**
5.1.5 wmanIfBsTraps **13**
5.2 wmanIfSsObjects **13**
5.2.1 wmanSsSystem **13**
5.2.1.1 wmanSsInventoryInfoTable **13**
5.2.2 wmanIfSsCps **13**
5.2.2.1 wmanIfSsConfigurationTable **13**
5.2.2.2 wmanIfSsStatisticsCountersTable **14**
5.2.3 wmanIfSsPkm **14**
5.2.3.1 wmanIfSsPkmAuthTable **14**
5.2.3.2 wmanIfSsPkmTekTable **14**
5.2.3.3 wmanIfSsPkmCertificatesTable **14**

5.2.4 wmanIfSsTraps **14**
5.3 wmanIfCommonObjects **14**
5.3.1 wmanIfCmnPacketCs **14**
5.3.1.1 wmanIfCmnClassifierRuleTable **14**
5.3.2 wmanIfCmnCps **14**
5.3.2.1 wmanIfCmnSystemParameters **14**
5.3.2.2 wmanIfCmnSsCapabilitiesTable (index = ifIndex) **14**
5.3.2.3 wmanIfCmnServiceFlowTable **15**
5.3.3 wmanIfCmnPrivacy **15**
5.3.3.1 wmanIfCmnCryptoSuiteTable **15**
5.3.4 wmanIfCmnOfdmPhy **15**
5.3.4.1 wmanIfOfdmUplinkChannelTable **15**
5.3.4.2 wmanIfOfdmDownlinkChannelTable **15**
5.3.4.3 wmanIfOfdmUcdBurstProfileTable **16**
5.3.4.4 wmanIfOfdmDcdBurstProfileTable **16**
6. ASN.1 Definition of 802.16 MIB ***17***

1. Introduction

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

1.1 Scope

The scope of this contribution is to define the 802.16 MAC and PHY MIB for SS and BS, based on IEEE 802.16REVd/D3 specification [3]. The definition of managed objects in this MIB is based on SNMPv2 Structure of Management Information (SMI) [4] and Textual Conventions [5]. Therefore, 802.16 MIB is compliant to SNMPV2, but is backward compatible to SNMPv1 through appropriate translation.

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

1.2 References

[1] IEEE 802.16-2001, “IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Wireless Access Systems”.

[2] IEEE 802.16a-2003, “IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Wireless Access Systems – Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz.

[3] IEEE 802.16REVd/D3-2004, “Draft IEEE Standard for Local and Metropolitan area networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

[4] RFC1902, “Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)”, January 1996

[5] RFC1903, “Textual Convention for Version 2 of the Simple Network Management Protocol (SNMPv2)”, January 1996

[6] RFC 1213, " Management Information Base for Network Management of TCP/IP-based internets: MIB-II", IETF, March 1991

[7] RFC2863, “The Interfaces Group MIB”, June, 2000

[8] RFC2515, “Definitions of Managed Objects for ATM Management”, February, 1999

2. BWA Network Management Reference Model

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

The management information between SS and BS will be carried over Second Management CID for managed SS. If it is an unmanaged SS where 2nd management CID does not exist, the SNMP messages shall go through another interface in the customer premise. However, the definition of 802.16 MIB should be independent of managed / unmanaged SS, or proxy / direct mode.

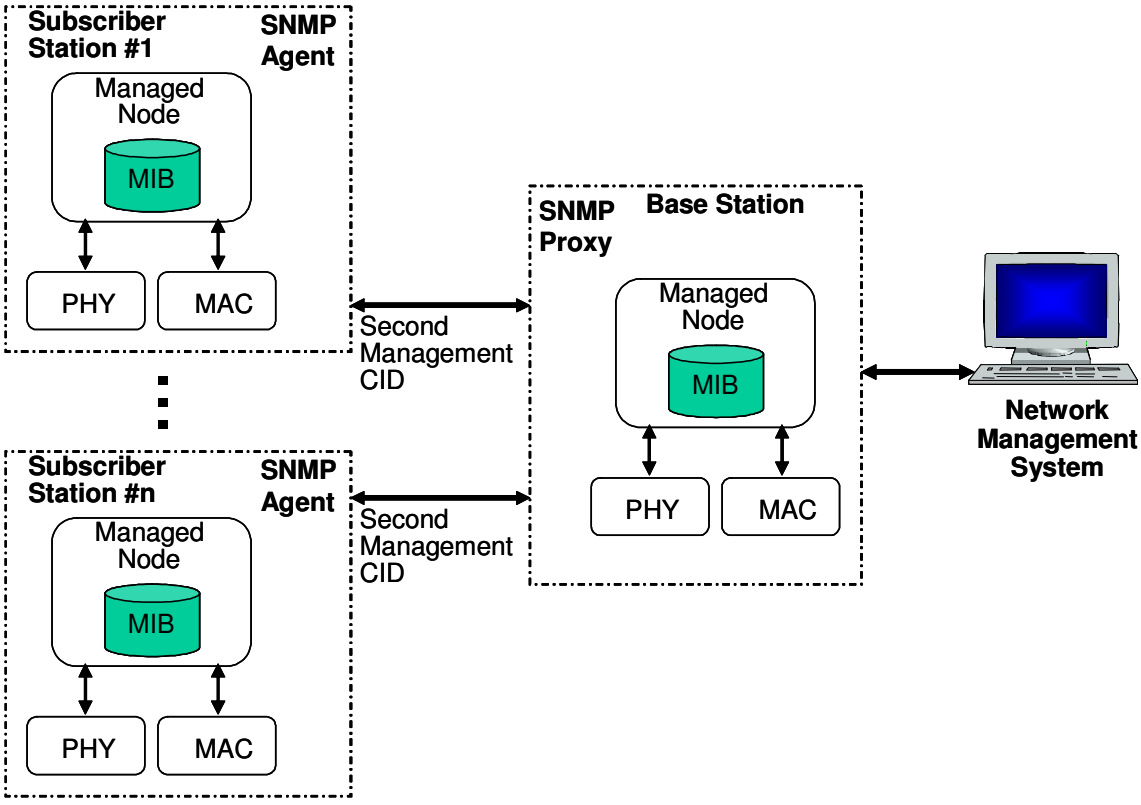

Figure 1 - BWA Network Management Reference Model

3. Relationship with Interface MIB

This section describes the integration with MIB-II [6] under Interface Group MIB defined in RFC2863, as 802.16 MIB will need to be integrated in the MIB tree. It describes where 802.16 MIB 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.

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

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

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

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

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

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

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

Or

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

If propBWAp2Mp has been designed for other BWA interface, it will be required to ask IANA to assign another IANAifType for 802.16. In addition, two IANAifType should be assigned by IANA to represent 802.16 downstream and upstream interfaces. The flowing shows the structure of IANAifType that should be assigned by IANA.

```
IANAifType ::= TEXTUAL-COVENTION
SYNTAX INTEGER {
      wmanMAC (?)        -- wireless MAN MAC Layer
      wmanDownstream (?) -- wirelesss MAN downstream interface
      wmanUpstream (?)   -- wireless MAN upstream interface
 }
```

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 wmanIfMib, and provides an example on how ifIndex will be assigned for each sub-layer.

Three ifEntry.will be created for each BS or radio channel installed in the system. One ifIndex is for BS MAC sub-layer, while two ifIndex will be created for BS upstream and downstream PHY sub-layers, respectively. When the SS has entered the network, three ifEntry.will be created as well , including one ifIndex for BS MAC sub-layer, and two ifIndex for upstream and downstream PHY sub-layers, respectively.

ifTable	*ifIndex*	*ifType (IANA)*	*ifSpeed*	*ifPhysAddress*	*ifAdminStatus*	*ifOperStatus*
BS MAC	An ifEntry per BS (1)	wmanMAC (?)	Null	MAC address	Administration Status	Operational Status
BS Down-Stream	An ifEntry per BS Downstream Channel (2)	wmanDownstream (?)	Speed of downstream channel in Mbps/s	Null	Administration Status	Operational Status
BS Upstream	An ifEntry per BS Upstream Channel (3)	wmanUpstream (?)	Speed of upstream channel in Mbps/s	Null	Administration Status	Operational Status
SS MAC	An ifEntry per SS (4)	wmanMAC (?)	Null	MAC address	Administration Status	Operational Status
SS Down-stream	An ifEntry per SS Downstream Channel (5)	wmanDownstream (?)	Speed of downstream channel in Mbps/s	Null	Administration Status	Operational Status
SS upstream	An ifEntry per SS Upstream Channel (6)	wmanUpstream (?)	Speed of upstream channel in Mbps/s	Null	Administration Status	Operational Status

Table 1 – Usage of ifTable objects

ifStackTable must be used to define the association of MAC, downstream, and upstream sub-layers in SS and BS. Table 2 shows the example of ifStackTable for SS and BS sub-layers as illustrated in Table 1.

ifStackTable	*ifIndex*									
ifStackHigherLayer	0	1	1	2	3	0	4	4	5	6
ifStackLowerLayer	1	2	3	0	0	4	5	6	0	0

Table 2 – Usage of ifStackTable

4. Service Flow Creation

Section 6.4.13.7 of IEEE 802.16REVd/D3 standard [3] specifies that there are two types of service flow creation, which are pre-provisioning and dynamic service flow creation, similar to PVC (Permanent Virtual Circuit) and SVC (Switched Virtual Circuit) for ATM. In most cases, BWA will provide subscribed services; therefore, pre-provisioning shall be required. 802.16 MIB defines pre-provisioned service flow table to support pre-provisioning service flow creation. In the mobile environment, where a mobile SS requests Internet access in the hotspot, it may be necessary to use dynamic service flow creation.

Section 6.4.13.6 of IEEE 802.16REVd/D3 standard [3] specifies that there are three types of service flows, which are Provisioned, Admitted, and Active service flows. This section describes the usage scenario of the 802.16 MIB related to the transition of service flow types.

- Provisioned Service Flow: When the NMS pre-provisions a service flow, an entry in `wmanIfBsPreProvisionedSfTable` in the BS MIB is created. It is assigned a Provisioned Service Flow state, but is not active yet. An attribute—`wmanIfBsSfState`–in `wmanIfBsPreProvisionedSfTable` determines how the service flow shall be transitioned to the next state, when a trigger is generated. When `wmanIfBsSfState` is in one of the following state:
  - Admitted or Active state: The pre-provisioned service flow will be transitioned to the Admitted or Active state, as soon as the SS passes the network entry procedure, and connection admission control. An entry will be created in the SS and BS service flow tables.
  - Provisioned state: After SS enters the network; the pre-provisioned service flow will remain in the Provisioned state until NMS set it different state. An entry will be created in the SS and BS service flow tables.
- Admitted Service Flow: Network resource will be reserved when the service flow is in the Admitted service flow state. The service flow can be transitioned to the active service flow state by application triggers, such as off-hook for UGS service flows.
- Active Service Flow: Active Service Flow state indicates the data packets are sent upstream and downstream by UL-MAP and DL-MAP. The service flow can be transitioned directly from Provision service flow state to the Active service flow state.

5. 802.16 MIB Structure

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

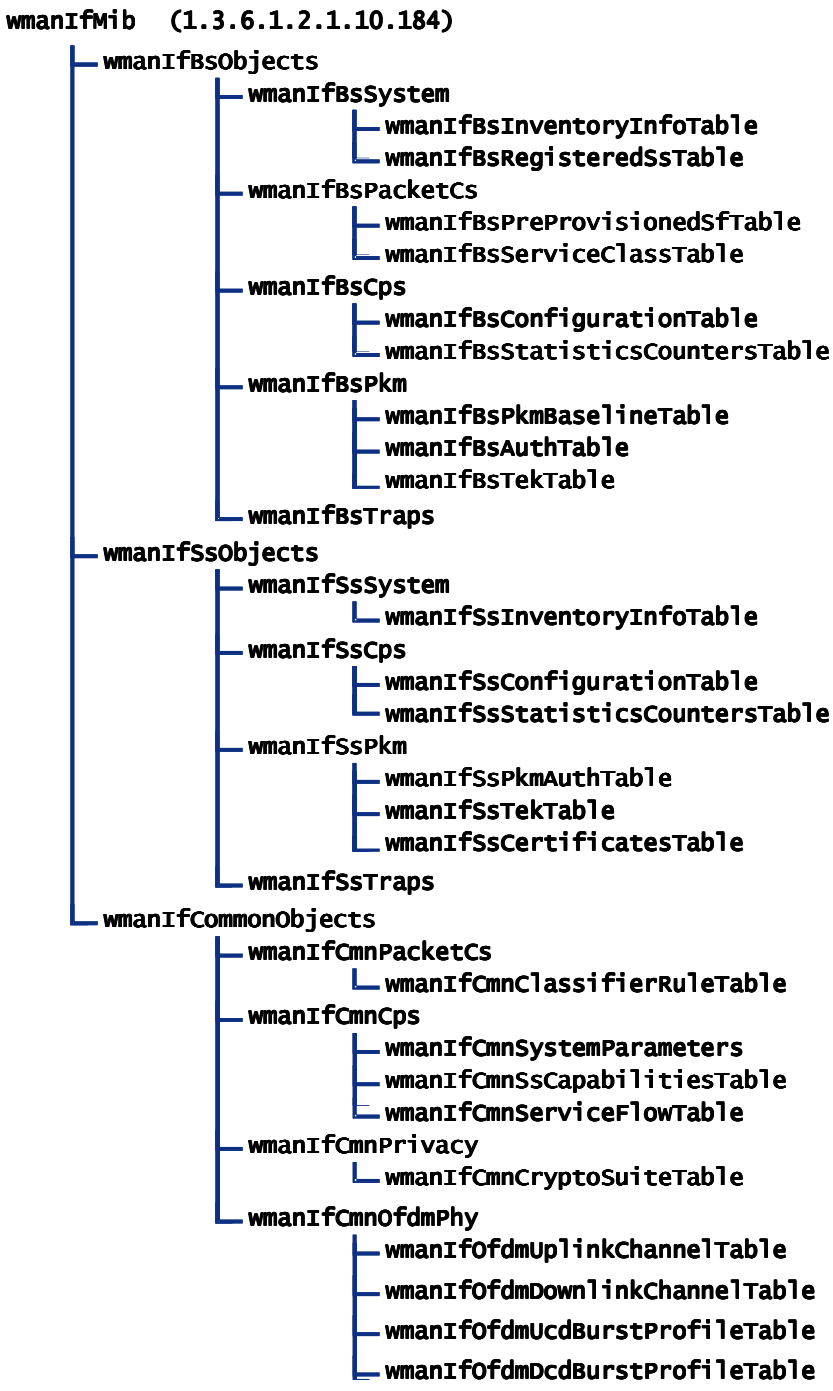

Figure 2 – WirelessMan MIB Structure

wmanIfMib is composed of three groups:

- wmanIfBsObjects: This group contains managed objects to be implemented in the SNMP agent in BS.
- wmanIfSsObjects: This group contains managed objects to be implemented in the SNMP agent in SS.
- wmanIfCommonObjects: This group contains common managed objects to be implemented in the SNMP agent in BS and SS.

5.1 wmanIfBsObjects

5.1.1 wmanIfBsSystem

wmanIfBsSystem group contains system level BS managed objects.

5.1.1.1 wmanIfBsInventoryInfoTable

This table is indexed by BS ifIndex and contains inventory information about the Base Station such as manufacturer, hardware model, serial number, and software or firmware revision.

5.1.1.2 wmanIfBsRegisteredSsTable

This table is indexed by SS ifIndex and contains SS information obtained from REG-REQ message as defined in section 6.4.3.2.7 in [2]. Each entry in the table may contain the following objects.

- Basic CID
- Primary management CID
- Secondary Management CID
- HMAC tuple
- Uplink CID support
- SS management support
- SS capability
- IP version
- CS sublayer capabilities

5.1.2 wmanIfBsPacketCs

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

5.1.2.1 wmanIfBsPreProvisionedSfTable

This table is doubly indexed (SS MAC address, SF ID) and contains pre-provisioned service flow profiles, Per SS. These connection parameters shall be provisioned for the SS using DSA messages, as specified in [3], 6.4.13. Admittance and activation of provisioned service flow may be postponed.

5.1.2.2 wmanIfBsServiceClassTable

This table is provisioned and is indexed by Service Class ID. Each entry of the table contains corresponding service flow characteristic attributes (e.g. QoS parameter set) as defined in section 6.4.13.4 in [3].

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

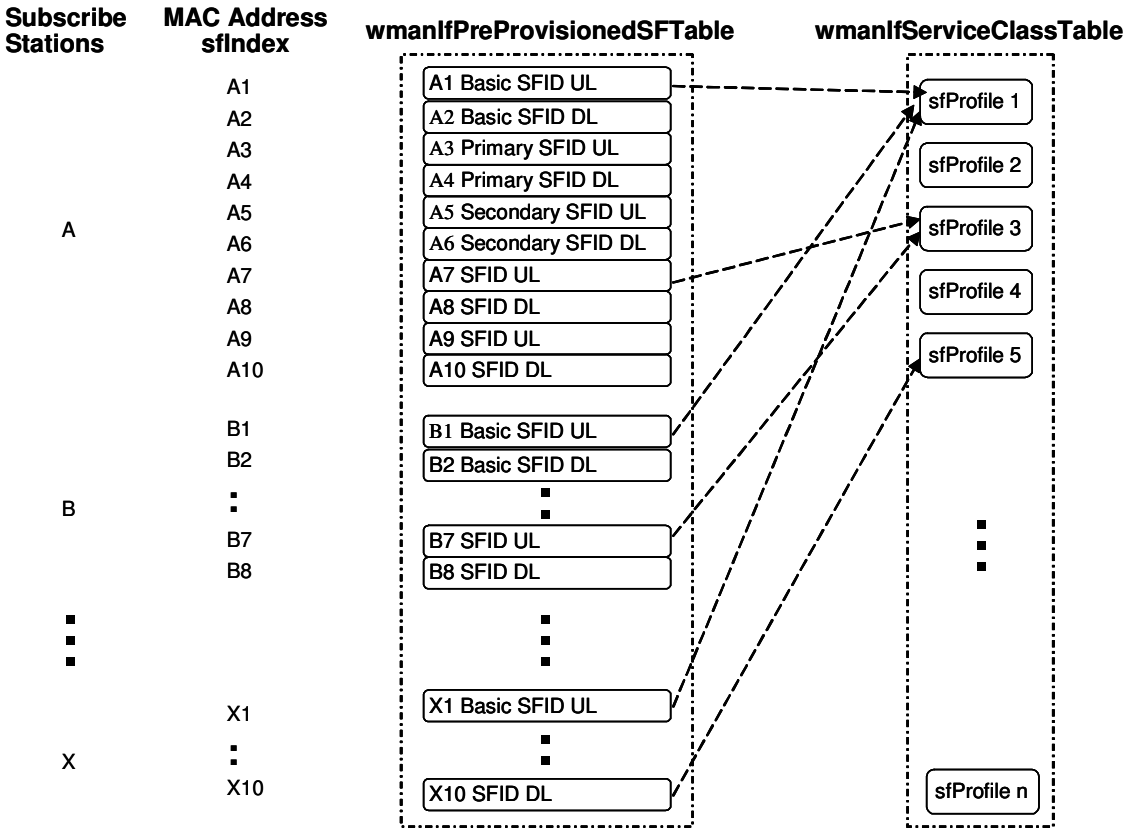

Figure 3 – Service Classes – Service Flows Mapping

5.1.3 wmanIfBsCps

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

5.1.3.1 wmanIfBsConfigurationTable

This table contains objects for BS system parameters and constants as defined in section 10.1, Table 264 of [3].

- Base station ID

5.1.3.2 wmanIfBsStatisticsCountersTable

This table is indexed by BS ifIndex and contains statistics about the packet count and various error conditions related to the operation of the BS.

5.1.4 **wmanIfBsPkm**

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

5.1.4.1 wmanIfBsPkmBaselineTable

This table is indexed by BS ifIndex and contains base station PKM operational parameters described in section 10.2 and table 270 of [3].

5.1.4.2 wmanIfBsAuthTable

This table is double indexed by ifIndex and SsMacAddress and contains runtime subscriber station authentication and authorization parameters for each base station.

5.1.4.3 wmanIfBsTekTable

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

5.1.5 **wmanIfBsTraps**

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

5.2 wmanIfSsObjects

5.2.1 **wmanSsSystem**

wmanIfSsSystem group contains subscriber station system level objects.

5.2.1.1 wmanSsInventoryInfoTable

This table is indexed by SS MAC address and contains inventory information about the subscriber station such as manufacturer, hardware model, serial number, and software or firmware revision.

5.2.2 **wmanIfSsCps**

wmanIfSsCpsParameters group contains subscriber station manageable objects relating to the MAC CPS management entity layer in figure 1 of [3].

5.2.2.1 wmanIfSsConfigurationTable

This table is indexed by SS MAC address and contains objects for SS system parameters and constants as defined in section 10.1, Table 264 of [3].

5.2.2.2 wmanIfSsStatisticsCountersTable

This object contains the performance monitoring data for SS.

5.2.3 **wmanIfSsPkm**

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

5.2.3.1 wmanIfSsPkmAuthTable

This table is indexed by SS MAC address and contains subscriber station authentication and authorization parameters including those described in section 10.2 and table 270 of [3].

5.2.3.2 wmanIfSsPkmTekTable

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

5.2.3.3 wmanIfSsPkmCertificatesTable

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

5.2.4 **wmanIfSsTraps**

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

5.3 wmanIfCommonObjects

5.3.1 **wmanIfCmnPacketCs**

5.3.1.1 wmanIfCmnClassifierRuleTable

wmanIfClassifierRuleTable is indexed by service flow ID and contains runtime classifier rules screening criteria for each service flow as described in section 11.13.22.3.4 of [3].

5.3.2 **wmanIfCmnCps**

5.3.2.1 wmanIfCmnSystemParameters

wmanIfCommonSystemParameters contain objects for common BS and SS system parameters and constants as defined in section 10.1, Table 264 of [3].

5.3.2.2 wmanIfCmnSsCapabilitiesTable (index = ifIndex)

This table is indexed by SS MAC address and negotiated SS capability parameters and constants as defined in sections 11.4.2.2 and 11.4.2.6 in [3]. In the BS, this table contains SS capability parameters for all SSs, while in the BS, this table should have an entry indicating the capability of such SS. In the BS, NMS has read / write access to this table. The SS capability parameters are provided to SS via SBC-RSP message during the network entry; hence NMS has only read access to SS.

5.3.2.3 wmanIfCmnServiceFlowTable

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

A Service Flow is represented by parameters, such as

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

5.3.3 **wmanIfCmnPrivacy**

5.3.3.1 wmanIfCmnCryptoSuiteTable

This table is doubly indexed by ifIndex and wmanIfCryptoSuiteIndex and contains supported crypto suites for the particular SS and other crypto parameters such as key lifetimes. See sections 11.2.14 and 11.2.15 of [3].

5.3.4 **wmanIfCmnOfdmPhy**

`wmanIfofdmPhy` is a group containing objects specific to OFDM PHY.

5.3.4.1 wmanIfOfdmUplinkChannelTable

This table contains the uplink channels that the BS is able to receive. In the SS, this table should have an entry indicating the uplink channel that the SS can transmit. Each entry contains the parameters needed to describe uplink channel descriptor as defined in section 11, Table 268 of [3], and include the following objects.

- Uplink center frequency (KHz)
- Subchannelization REQ Region-Full Parameters
- Bandwidth request opportunity size
- Ranging request opportunity size

5.3.4.2 wmanIfOfdmDownlinkChannelTable

This table contains the downlink channels that the BS is able to transmit. In the SS, this table should have an entry indicating the downlink channel that the SS can receive. Each entry contains the parameters needed to describe downlink channel descriptor as defined in section 11, Table 273 of [3], and including the following.

- channel number (for license exempt operation only)
- Frequency (downlink center frequency (kHz)
- BS EIRP

- TTG
- RTG
- MAC Version

5.3.4.3 wmanIfOfdmUcdBurstProfileTable

Each entry in this table contains the parameters needed for the UCD burst profile as defined in section 11, Table 271 of [3].

5.3.4.4 wmanIfOfdmDcdBurstProfileTable

`wmanIfDcdBurstProfileTable` – Each entry in this table contains the parameters needed for the UCD burst profile as defined in section 11, Table 276 of [3].

6. ASN.1 Definition of 802.16 MIB

```
WMAN-IF-MIB DEFINITIONS ::= BEGIN

  IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        Unsigned32,
        Integer32,
        Counter32,
        Counter64,
        TimeTicks,
        IpAddress,
        transmission
                FROM SNMPv2-SMI
        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB
        TEXTUAL-CONVENTION,
        MacAddress,
        RowStatus,
        TruthValue,
        DateAndTime,
        DisplayString,
        TimeInterval,
        TimeStamp
                FROM SNMPv2-TC
        InetAddressType, InetAddress
                FROM INET-ADDRESS-MIB
        OBJECT-GROUP,

        MODULE-COMPLIANCE
                FROM SNMPv2-CONF
        ifIndex, InterfaceIndexOrZero
                FROM IF-MIB;

wmanIfMib MODULE-IDENTITY
        LAST-UPDATED    "0403110000Z" -- March 11, 2004
        ORGANIZATION    "IETF IPCDN Working Group"
        CONTACT-INFO
            "        Joey Chou
             Postal: Intel Corporation
                     5000 W. Chandler Blvd, Chandler, AZ 85227, USA
             E-mail: joey.chou@intel.com

                     Russ Reynolds
             Postal: Proxim Corporation
                     935 Stewart Drive, Sunnyvale, CA 94085, USA
             E-mail: RReynolds@proxim.com

                     Shlomi Eini
             Postal: Airspan Networks
                         Airport city 70100,Israel
```

```
                E-mail: seini@airspan.com"

        DESCRIPTION
            "This MIB Module defines managed objects for 802.16 based
            Subscriber Station and Base Station."
        ::= { transmission 184 }

-- Textual Conventions

wmanIfMibObjects  OBJECT IDENTIFIER ::= { wmanIfMib 1 }
wmanIfBsObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 1 }
wmanIfSsObjects   OBJECT IDENTIFIER ::= { wmanIfMibObjects 2 }
wmanIfCommonObjects OBJECT IDENTIFIER ::= { wmanIfMibObjects 3 }

--
-- BS object group - containing tables and objects to be implemented in
-- the Base station
--
-- wmanIfBsSystem contain the Base Station system objects
wmanIfBsSystem OBJECT IDENTIFIER ::= { wmanIfBsObjects 1 }

wmanIfBsRegisteredSsTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF WmanIfBsRegisteredSsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table contains entries of SSs that have been
            registered through REG-REQ message"
        REFERENCE
            "Section 6.4.3.2.7 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsSystem 1 }

wmanIfBsRegisteredSsEntry OBJECT-TYPE
        SYNTAX      WmanIfBsRegisteredSsEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
              "This table provides one row for each SS that has been
              registered in the BS, and is indexed by SS ifIndex. An
              entry in this table exists for each ifEntry of SS with an
              ifType of wmanMac."
        INDEX { ifIndex }
        ::= { wmanIfBsRegisteredSsTable 1 }

WmanIfBsRegisteredSsEntry ::= SEQUENCE {
        wmanIfBsSsMacAddress            MacAddress,
        wmanIfBsSsBaseStationIndex      INTEGER,
        wmanIfBsSsBasicCid              INTEGER,
        wmanIfBsSsPrimaryCid            INTEGER,
        wmanIfBsSsSecondaryCid          INTEGER,
        wmanIfBsHmacTuple               OCTET STRING,
        wmanIfBsUlCidSupport            INTEGER,
        wmanIfBsSsManagementSupport     INTEGER,
        wmanIfBsSsArqSupport            INTEGER,
```

```
        wmanIfBsSsDsxFlowControl        INTEGER,
        wmanIfBsSsMacCrcSupport         INTEGER,
        wmanIfBsSsMcaFlowControl        INTEGER,
        wmanIfBsSsMcpGroupCidSupport    INTEGER,
        wmanIfBsSsPkmFlowControl        INTEGER,
        wmanIfBsIpVersion               INTEGER,
        wmanIfBsSSMacCsSupportBitMap    BITS,
        wmanIfBsSSMaxNumOfClassifier    INTEGER,
        wmanIfBsSSPhsSupport            INTEGER
        }

wmanIfBsSsMacAddress OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The MAC address of SS is received from the RNG-REQ
             message. This MAC address can be used as the
             index to find out the BS and its associated SSs."
        REFERENCE
            "Section 6.4.2.3.5 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 1 }


wmanIfBsSsBaseStationIndex OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "wmanIfBsSsBaseStationIndex identifies with which BS the
             SS is associated."
        REFERENCE
            "Section 6.4.2.3.5 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 2 }

wmanIfBsSsBasicCid OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object indicates the SS's basic CID
             that was sent in the RNG-RSP message."
        REFERENCE
            "Section 6.4.9.5 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 3 }

wmanIfBsSsPrimaryCid OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object indicates the basic CID of the
             SS received from the RNG-RSP message."
        REFERENCE
```

```
                "Section 6.4.9.5 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 4 }

    wmanIfBsSsSecondaryCid OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object indicates the secondary
                 management CID present in the REG-REQ message."
            REFERENCE
                "Section 6.4.2.3.8 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 5 }

    wmanIfBsHmacTuple OBJECT-TYPE
            SYNTAX      OCTET STRING
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This parameter contains the HMAC Key Sequence Number
                 concatenated with an HMAC-Digest used for message
                 authentication. The HMAC Key Sequence Number is stored
                 in the four least significant bits of the first byte of
                 the HMAC Tuple, and the most significant four bits
                 are reserved."
            REFERENCE
                "Section 11.1.2 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 6 }

    wmanIfBsUlCidSupport OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object shows the number of Uplink CIDs the SS can
                 support."
            REFERENCE
                "Section 11.7.4 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 7 }

    wmanIfBsSsManagementSupport OBJECT-TYPE
            SYNTAX      INTEGER {unmanagedSs(0),
                                 managedSs(1)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object indicates whether or not the SS is managed."
            REFERENCE
                "Section 11.7.1.1 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 8 }

    wmanIfBsSsArqSupport OBJECT-TYPE
            SYNTAX      INTEGER {arqOn(0),
                                 arqOff(1)}
```

```
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object indicates whether the SS support ARQ."
            REFERENCE
                "Section 11.7.6.1 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 9 }

    wmanIfBsSsDsxFlowControl OBJECT-TYPE
            SYNTAX      INTEGER (0..255)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object specifies the maximum number of concurrent
                 DSA, DSC, or DSD transactions that may be outstanding."
            REFERENCE
                "Section 11.7.6.2 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 10 }

    wmanIfBsSsMacCrcSupport OBJECT-TYPE
            SYNTAX      INTEGER {noMacCrcSupport(0),
                                 macCrcSupport(1)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object indicates whether or not the SS supports MAC
                 level CRC."
            REFERENCE
                "Section 11.7.6.3 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 11 }

    wmanIfBsSsMcaFlowControl OBJECT-TYPE
            SYNTAX      INTEGER (0..255)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object specifies the maximum number of concurrent
                 MCA transactions that may be outstanding."
            REFERENCE
                "Section 11.7.6.4 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 12 }

    wmanIfBsSsMcpGroupCidSupport OBJECT-TYPE
            SYNTAX      INTEGER (0..255)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "This object indicates the maximum number of
                 simultaneous Multicast Polling Groups the SS is
                 capable of belonging to."
            REFERENCE
                "Section 11.7.6.5 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsRegisteredSsEntry 13 }
```

```
wmanIfBsSsPkmFlowControl OBJECT-TYPE
        SYNTAX      INTEGER (0..255)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "This object specifies the maximum number of concurrent PKM
             transactions that may be outstanding."
        REFERENCE
            "Section 11.7.6.6 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 14 }

wmanIfBsIpVersion OBJECT-TYPE
        SYNTAX      INTEGER {ipv4(1),
                             ipv6(2)}
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "This object indicates the version of IP used on the
             Secondary Management Connection."
        REFERENCE
            "Section 11.7.2.1 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 15 }

wmanIfBsSSMacCsSupportBitMap OBJECT-TYPE
        SYNTAX      BITS {atm(0),
                          packetIpv4(1),
                          packetIpv6(2),
                          packet802-3(3),
                          packet802-1Q(4),
                          packetIpv4Over802-3(5),
                          packetIpv6Over802-3(6),
                          packetIpv4Over802-1Q(7),
                          packetIpv6Over802-1Q(8)}
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "This object indicates the set of MAC convergence
             sublayer support. When a bit is set, it indicates
             the corresponding CS feature is supported."
        REFERENCE
            "Section 11.7.5.1 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 16 }

wmanIfBsSSMaxNumOfClassifier OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "This object indicates the maximum number of admitted
             Classifiers that the SS is allowed to have."
        REFERENCE
            "Section 11.7.5.2 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsRegisteredSsEntry 17 }
```

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

```
                "This table provides one row for each service flow been
                pre-provisioned by NMS."
        INDEX { wmanIfBsSsProvMacAddress, wmanIfBsSfId}
        ::= { wmanIfBsPreProvisionedSfTable 1 }

WmanIfBsPreProvisionedSfEntry ::= SEQUENCE {
        wmanIfBsSfId                      INTEGER,
        wmanIfBsSfDirection               INTEGER,
        wmanIfBsServiceClassIndex         INTEGER,
        wmanIfBsSsProvMacAddress          MacAddress,
        wmanIfBsSfState                   INTEGER,
        wmanIfBsSfCreateTime              TimeStamp,
        wmanIfBsPreProvisionedSfRowStatus RowStatus
        }

wmanIfBsSfId OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "A 32 bit quantity that uniquely identifies a service flow
            to both the subscriber station and base station (BS)."
        ::= { wmanIfBsPreProvisionedSfEntry 1 }

wmanIfBsSfDirection OBJECT-TYPE
        SYNTAX      INTEGER {downstream(1),
                             upstream(2)}
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An attribute indiating the service flow is downstream or
            upstream."
        ::= { wmanIfBsPreProvisionedSfEntry 2 }

wmanIfBsServiceClassIndex OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The index in wmanIfBsServiceClassTable describing the
             service class or QoS parameters for such service flow.
             If no associated entry in wmanIfBsServiceClassTable
             exists, this object returns a value of zero."
        ::= { wmanIfBsPreProvisionedSfEntry 3 }

wmanIfBsSsProvMacAddress OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The MAC address of the SS, where the service flow resides.
             It can be used as the index to associate service flows
             with the SS."
        ::= { wmanIfBsPreProvisionedSfEntry 4 }
```

```
1
2    wmanIfBsSfState OBJECT-TYPE
3            SYNTAX      INTEGER {provisionedState(1),
4                                 admittedState(2),
5                                 activeState(3)}
6            MAX-ACCESS  read-write
7            STATUS      current
8            DESCRIPTION
9                "wmanIfBsSfState determines how the service flow will be
10               transitioned to the Admitted or Active state.
11               Admitted or Active state: The pre-provisioned service flow
12               will be transitioned to the Admitted or Active state, as
13               soon as the SS passes the network entry procedure, and
14               connection admission control. An entry will be created
15               in the SS and BS service flow tables.
16               Provisioned state: After SS enters the network; the
17               pre-provisioned service flow will remain in the Provisioned
18               state until NMS set it different state. An entry will be
19               created in the SS and BS service flow tables"
20           REFERENCE
21               "Section 6.4.13.6, in IEEE 802.16REVd/D3-2004"
22           ::= { wmanIfBsPreProvisionedSfEntry 5 }
23
24   wmanIfBsSfCreateTime OBJECT-TYPE
25           SYNTAX      TimeStamp
26           MAX-ACCESS  read-write
27           STATUS      current
28           DESCRIPTION
29               "Indicates the data and time when the service flow is
30               created."
31           ::= { wmanIfBsPreProvisionedSfEntry 6 }
32
33   wmanIfBsPreProvisionedSfRowStatus OBJECT-TYPE
34           SYNTAX          RowStatus
35           MAX-ACCESS      read-create
36           STATUS          current
37           DESCRIPTION
38               "This object is used to create a new row or modify or
39               delete an existing row in this table.
40
41               If the implementator of this MIB has choosen not
42               to implement 'dynamic assignment' of profiles, this
43               object is not useful and should return noSuchName
44               upon SNMP request."
45           ::= { wmanIfBsPreProvisionedSfEntry 7 }
46
47   wmanIfBsServiceClassTable OBJECT-TYPE
48           SYNTAX      SEQUENCE OF WmanIfBsServiceClassEntry
49           MAX-ACCESS  not-accessible
50           STATUS      current
51           DESCRIPTION
52               "This table is provisioned and is indexed by Service Class
53               ID. Each entry of the table contains corresponding service
54               flow characteristic attributes (e.g. QoS parameter set)"
```

```
        REFERENCE
            "Section 6.4.13.4 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsPacketCs 2 }

wmanIfBsServiceClassEntry OBJECT-TYPE
        SYNTAX      WmanIfBsServiceClassEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table provides one row for each service class"
        INDEX { wmanIfBsQoSProfileIndex }
        ::= { wmanIfBsServiceClassTable 1 }

WmanIfBsServiceClassEntry ::= SEQUENCE {
        wmanIfBsQoSProfileIndex           INTEGER,
        wmanIfBsQosServiceClassName       DisplayString,
        wmanIfBsQoSTrafficPriority        INTEGER,
        wmanIfBsQoSMaxSustainedRate       INTEGER,
        wmanIfBsQoSMaxTrafficBurst        INTEGER,
        wmanIfBsQoSMinReservedRate        INTEGER,
        wmanIfBsQoSToleratedJitter        INTEGER,
        wmanIfBsQoSMaxLatency             INTEGER,
        wmanIfBsQosScSchedulingType              WmanUlSchedulingType,
        wmanIfBsQosScArqEnable            TruthValue,
        wmanIfBsQosScArqWindowSize        INTEGER,
        wmanIfBsQosScArqFragmentLifetime  INTEGER,
        wmanIfBsQosScArqSyncLossTimeout   INTEGER,
        wmanIfBsQosScArqDeliverInOrder    TruthValue,
        wmanIfBsQosScArqRxPurgeTimeout    INTEGER,
        wmanIfBsQosScFragmentLen          INTEGER,
        wmanIfBsQosSCMinRsvdTolerableRate INTEGER,
        wmanIfBsQoSServiceClassRowStatus  RowStatus
        }

wmanIfBsQoSProfileIndex OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index value which uniquely identifies an entry
            in the wmanIfBsServiceClassTable"
        ::= { wmanIfBsServiceClassEntry 1 }

wmanIfBsQosServiceClassName  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION    "Refers to the Service Class Name"
        REFERENCE      "802.16 Chapter 11.4.8"
        ::= { wmanIfBsServiceClassEntry 2 }

wmanIfBsQoSTrafficPriority OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-create
```

```
            STATUS      current
            DESCRIPTION
                "The value of this parameter specifies the priority
                assigned to a service flow. For uplink service flows,
                the BS should use this parameter when determining
                precedence in request service and grant generation,
                and the SS shall preferentially select contention
                Request opportunities for Priority Request CIDs
                based on this priority"
            REFERENCE
                "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsServiceClassEntry 3 }

    wmanIfBsQoSMaxSustainedRate OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This parameter defines the peak information rate
                of the service. The rate is expressed in bits per
                second and pertains to the SDUs at the input to
                the system."
            REFERENCE
                "Section 11.13.8 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsServiceClassEntry 4 }

    wmanIfBsQoSMaxTrafficBurst OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This parameter defines the maximum burst size that
                must be accommodated for the service."
            REFERENCE
                "Section 11.13.9 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsServiceClassEntry 5 }

    wmanIfBsQoSMinReservedRate OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This parameter specifies the minimum rate reserved
                for this service flow."
            REFERENCE
                "Section 11.13.10 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfBsServiceClassEntry 6 }

    wmanIfBsQoSToleratedJitter OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This parameter defines the Maximum delay
```

```
            variation (jitter) for the connection."
        REFERENCE
            "Section 11.13.15 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsServiceClassEntry 7 }

wmanIfBsQoSMaxLatency OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The value of this parameter specifies the maximum
            latency between the reception of a packet by the BS
            or SS on its network interface and the forwarding
            of the packet to its RF Interface."
        REFERENCE
            "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfBsServiceClassEntry 8 }

wmanIfBsQosScSchedulingType OBJECT-TYPE
        SYNTAX      WmanUlSchedulingType
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Specifies the upstream scheduling service used for
            upstream service flow. If the referenced parameter
            is not present in the corresponding 802.16 QOS
            Parameter Set of an upstream service flow, the
            default value of this object is bestEffort(2)."
        REFERENCE      "802.16 Chapter 11.4.8"
        DEFVAL         {2}
        ::= { wmanIfBsServiceClassEntry 9}

wmanIfBsQosScArqEnable OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "True(1) ARQ enabling is requested for the connection."
        ::= { wmanIfBsServiceClassEntry 10}

wmanIfBsQosScArqWindowSize    OBJECT-TYPE
        SYNTAX      INTEGER (1 .. 255)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Indicates the maximum number of unacknowledged
              fragments at any time."
        ::= { wmanIfBsServiceClassEntry 11}

wmanIfBsQosScArqFragmentLifetime OBJECT-TYPE
        SYNTAX      INTEGER (0 .. 65535)
        UNITS       "10 us"
        MAX-ACCESS  read-create
        STATUS      current
```

```
            DESCRIPTION
                "The maximum time interval an ARQ fragment will be
                    managed by the transmitter ARQ machine, once
                    initial transmission of the fragment has occured.
                    If transmission or retransmission of the fragment
                    is not acknowledged by the receiver before the
                    time limit is reached, the fragmnet is discarded.
                    A value of 0 means Infinite."
            DEFVAL        {0}
            ::= { wmanIfBsServiceClassEntry 12 }

    wmanIfBsQosScArqSyncLossTimeout OBJECT-TYPE
            SYNTAX      INTEGER (0 .. 65535 )
            UNITS       "10 us"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The maximum interval before declaring a loss
                  of synchronization of the sender and receiver
                  state machines. A value of 0 means Infinite."
            DEFVAL      {0}
            ::= { wmanIfBsServiceClassEntry 13}

    wmanIfBsQosScArqDeliverInOrder  OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Indicates whether or not data is to be delivered
                  by the receiving MAC to its client application
                  in the order in which data was handed off to the
                  originating MAC."
            ::= { wmanIfBsServiceClassEntry 14 }

    wmanIfBsQosScArqRxPurgeTimeout  OBJECT-TYPE
            SYNTAX      INTEGER (0 .. 65535)
            UNITS       "10 us"
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "Indicates the time interval the ARQ window is advanced
                  after  a  fragment  is  received.  A  value  of  0  means
    Infinite."
            DEFVAL      {0}
            ::= { wmanIfBsServiceClassEntry 15}

    wmanIfBsQosScFragmentLen OBJECT-TYPE
            SYNTAX      INTEGER (32 .. 2040)
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "The maximum size fragment a transmitter shall form
                or a receiver shall expect to receive."
            ::= { wmanIfBsServiceClassEntry 16 }
```

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

```
        wmanIfBsClassifierRuleIndex             Unsigned32,
        wmanIfBsClassifierRuleServiceFlowId     Unsigned32,
        wmanIfBsClassifierRulePriority          INTEGER,
        wmanIfBsClassifierRuleIpTosLow          OCTET STRING,
        wmanIfBsClassifierRuleIpTosHigh         OCTET STRING,
        wmanIfBsClassifierRuleIpTosMask         OCTET STRING,
        wmanIfBsClassifierRuleIpProtocol        Integer32,
        wmanIfBsClassifierRuleInetAddressType   InetAddressType,
        wmanIfBsClassifierRuleInetSourceAddr    InetAddress,
        wmanIfBsClassifierRuleInetSourceMask    InetAddress,
        wmanIfBsClassifierRuleInetDestAddr      InetAddress,
        wmanIfBsClassifierRuleInetDestMask      InetAddress,
        wmanIfBsClassifierRuleSourcePortStart   Integer32,
        wmanIfBsClassifierRuleSourcePortEnd     Integer32,
        wmanIfBsClassifierRuleDestPortStart     Integer32,
        wmanIfBsClassifierRuleDestPortEnd       Integer32,
        wmanIfBsClassifierRuleDestMacAddr       MacAddress,
        wmanIfBsClassifierRuleDestMacMask       MacAddress,
        wmanIfBsClassifierRuleSourceMacAddr     MacAddress,
        wmanIfBsClassifierRuleSourceMacMask     MacAddress,
        wmanIfBsClassifierRuleEnetProtocolType  INTEGER,
        wmanIfBsClassifierRuleEnetProtocol      Integer32,
        wmanIfBsClassifierRuleUserPriLow        Integer32,
        wmanIfBsClassifierRuleUserPriHigh       Integer32,
        wmanIfBsClassifierRuleVlanId            Integer32,
        wmanIfBsClassifierRuleState             INTEGER,
        wmanIfBsClassifierRulePkts              Counter64,
        wmanIfBsClassifierRuleRowStatus         RowStatus
        }

wmanIfBsClassifierRuleIndex  OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "An index is assigned to a classifier in BS classifiers
           table"
        REFERENCE       ""
        ::= { wmanIfBsClassifierRuleEntry 1 }


wmanIfBsClassifierRuleServiceFlowId OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An index assigned to a service flow by SC (SFID)."
        REFERENCE       "802.16 Chapter 11.4.8"
        ::= { wmanIfBsClassifierRuleEntry 2 }

wmanIfBsClassifierRulePriority OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
```

```
            DESCRIPTION
                "The value specifies the order of evaluation of the
                classifiers. The higher the value the higher the
                priority. The value of 0 is used as default in
                provisioned service flows classifiers. The default
                value of 64 is used for dynamic service flow classifiers.
                If the referenced parameter is not present in a classifier,
                this object reports the default value as defined above"
        ::= { wmanIfBsClassifierRuleEntry 3 }

    wmanIfBsClassifierRuleIpTosLow OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The low value of a range of TOS byte values. If the
                referenced parameter is not present in a classifier, this
                object reports the value of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 4 }

    wmanIfBsClassifierRuleIpTosHigh OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The 8-bit high value of a range of TOS byte values.
                If the referenced parameter is not present in a classifier,
                this object reports the value of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 5 }

    wmanIfBsClassifierRuleIpTosMask OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The mask value is bitwise ANDed with TOS byte in an IP
                packet and this value is used check range checking of
                TosLow and TosHigh. If the referenced parameter is not
                present in a classifier, this object reports the value
                of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 6 }

    wmanIfBsClassifierRuleIpProtocol OBJECT-TYPE
            SYNTAX      Integer32 (0..255)
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object indicates the value of the IP Protocol field
                required for IP packets to match this rule. If the
                referenced parameter is not present in a classifier, this
                object reports the value of 0."
```

```
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 7 }

wmanIfBsClassifierRuleInetAddressType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The type of the internet address for
            wmanIfBsClassifierRuleInetSourceAddr,
            wmanIfBsClassifierRuleInetSourceMask,
            wmanIfBsClassifierRuleInetDestAddr, and
            wmanIfBsClassifierRuleInetDestMask.
            If the referenced parameter is not present in a classifier,
            this object reports the value of ipv4(1)."
        REFERENCE       ""
        ::= { wmanIfBsClassifierRuleEntry 8 }

wmanIfBsClassifierRuleInetSourceAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the value of the IP Source Address
            required for packets to match this rule. An IP packet
            matches the rule when the packet ip source address bitwise
            ANDed with the wmanIfBsClassifierRuleInetSourceMask value
            equals the wmanIfBsClassifierRuleInetSourceAddr value.
            If the referenced parameter is not present n a classifier,
            this object reports the value of 0.0.0.0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 9 }

wmanIfBsClassifierRuleInetSourceMask OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies which bits of a packet's IP Source
            Address that are compared to match this rule. An IP packet
            matches the rule when the packet source address bitwise
            ANDed with the
            wmanIfBsClassifierRuleInetSourceMask value equals the
            wmanIfBsClassifierRuleInetSourceAddr value.
            If the referenced parameter is not present in a classifier,
            this object reports the value of 0.0.0.0."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 10 }


wmanIfBsClassifierRuleInetDestAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-write
        STATUS      current
```

```
        DESCRIPTION
            "This object specifies the value of the IP Destination
            Address required for packets to match this rule. An IP
            packet matches the rule when the packet IP destination
            address bitwise ANDed with the
            wmanIfBsClassifierRuleInetDestMask value equals the
            wmanIfBsClassifierRuleInetDestAddr value.
            If the referenced parameter is not present in a
            classifier, this object reports the value of 0.0.0.0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 11 }

wmanIfBsClassifierRuleInetDestMask OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies which bits of a packet's IP
            Destination Address that are compared to match this rule.
            An IP packet matches the rule when the packet destination
            address bitwise ANDed with the
            wmanIfBsClassifierRuleInetDestMask value equals the
            wmanIfBsClassifierRuleInetDestAddr value.
            If the referenced parameter is not present in a classifier
            , this object reports the value of 0.0.0.0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 12}

wmanIfBsClassifierRuleSourcePortStart OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
             "This object specifies the low end inclusive range of
             TCP/UDP source port numbers to which a packet is compared
             . This object is irrelevant for non-TCP/UDP IP packets.
             If the referenced parameter is not present in a
             classifier, this object reports the value of 0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 13 }

wmanIfBsClassifierRuleSourcePortEnd OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the high end inclusive range of
            TCP/UDP source port numbers to which a packet is compared.
            This object is irrelevant for non-TCP/UDP IP packets.
            If the referenced parameter is not present in a classifier,
            this object reports the value of 65535."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 14 }
```

```
wmanIfBsClassifierRuleDestPortStart OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the low end inclusive range of
            TCP/UDP destination port numbers to which a packet is
            compared. If the referenced parameter is not present
            in a classifier, this object reports the value of 0."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 15 }

wmanIfBsClassifierRuleDestPortEnd OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the high end inclusive range of
            TCP/UDP destination port numbers to which a packet is
            compared. If the referenced parameter is not present
            in a classifier, this object reports the value of
            65535."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 16 }

wmanIfBsClassifierRuleDestMacAddr OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches an entry when its destination
            MAC address bitwise ANDed with
            wmanIfBsClassifierRuleDestMacMask equals the value of
            wmanIfBsClassifierRuleDestMacAddr. If the referenced
            parameter is not present in a classifier, this object
            reports the value of '000000000000'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 17 }

wmanIfBsClassifierRuleDestMacMask OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches an entry when its destination
            MAC address bitwise ANDed with
            wmanIfBsClassifierRuleDestMacMask equals the value of
            wmanIfBsClassifierRuleDestMacAddr. If the referenced
            parameter is not present in a classifier, this object
            reports the value of '000000000000'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 18 }

wmanIfBsClassifierRuleSourceMacAddr OBJECT-TYPE
```

```
            SYNTAX      MacAddress
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "An Ethernet packet matches this entry when its source
                MAC address  bitwise ANDed with
                wmanIfBsClassifierRuleSourceMacMask equals the value
                of wmanIfBsClassifierRuleSourceMacAddr. If the
                referenced parameter is not present in a classifier,
                this object reports the value of 'FFFFFFFFFFFF'H."
            REFERENCE      "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 19 }

    wmanIfBsClassifierRuleSourceMacMask OBJECT-TYPE
            SYNTAX      MacAddress
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "An Ethernet packet matches an entry when its destination
                MAC address bitwise ANDed with
                wmanIfBsClassifierRuleSourceMacMask equals the value of
                wmanIfBsClassifierRuleSourceMacAddr. If the referenced
                parameter is not present in a classifier, this object
                reports the value of '000000000000'H."
            REFERENCE      "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 20 }

    wmanIfBsClassifierRuleEnetProtocolType OBJECT-TYPE
            SYNTAX      INTEGER {none(0),
                                 ethertype(1),
                                 dsap(2)}
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object indicates the format of the layer 3 protocol
                id in the Ethernet packet. A value of none(0) means that
                the rule does not use the layer 3 protocol type as a
                matching criteria. A value of ethertype(1) means that the
                rule applies only to frames which contains an EtherType
                value. Ethertype values are contained in packets using
                the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
                Sub-Network Access Protocol (SNAP) encapsulation formats.
                A value of dsap(2) means that the rule applies only to
                frames using the IEEE802.3 encapsulation format with a
                Destination Service Access Point (DSAP) other than 0xAA
                (which is reserved for SNAP). If the Ethernet frame
                contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
                this object applies to the embedded EtherType field within
                the 802.1P/Q header. If the referenced parameter is not
                present in a classifier, this object reports the value of
                0."
            REFERENCE   "802.16 Chapter 11.4.9"
            ::= { wmanIfBsClassifierRuleEntry 21 }
```

```
wmanIfBsClassifierRuleEnetProtocol OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "If wmanIfBsClassifierRuleEnetProtocolType is none(0),
            this object is ignored when considering whether a packet
            matches the current rule.
            If wmanIfBsClassifierRuleEnetProtocolType is ethertype(1),
            this object gives the 16-bit value of the EtherType that
            the packet must match in order to match the rule.
            If wmanIfBsClassifierRuleEnetProtocolType is dsap(2), the
            lower 8 bits of this object's value must match the DSAP
            byte of the packet in order to match the rule.
            If wmanIfBsClassifierRuleEnetProtocolType is mac(3), the
            lower 8 bits of this object value represent a lower bound
            (inclusive) of MAC management message type codes matched,
            and the upper 8 bits of this object value represent the
            upper bound (inclusive) of matched MAC message type codes.
            Certain message type codes are excluded from matching, as
            specified in the reference.
            If the Ethernet frame contains an 802.1P/Q Tag header
            (i.e. EtherType 0x8100), this object applies to the
            embedded EtherType field within the 802.1P/Q header.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 0."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 22 }

wmanIfBsClassifierRuleUserPriLow OBJECT-TYPE
        SYNTAX      Integer32 (0..7)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object applies only to Ethernet frames using the
            802.1P/Q tag header (indicated with EtherType 0x8100).
            Such frames include a 16-bit Tag that contains a 3 bit
            Priority field and a 12 bit VLAN number.
            Tagged Ethernet packets must have a 3-bit Priority field
            within the range of wmanIfBsClassifierRulePriLow and
            wmanIfBsClassifierRulePriHigh in order to match this
            rule.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 0."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 23 }

wmanIfBsClassifierRuleUserPriHigh OBJECT-TYPE
        SYNTAX      Integer32 (0..7)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object applies only to Ethernet frames using the
            802.1P/Q tag header (indicated with EtherType 0x8100).
```

```
            Such frames include a 16-bit Tag that contains a 3 bit
            Priority field and a 12 bit VLAN number.
            Tagged Ethernet packets must have a 3-bit Priority
            field within the range of wmanIfBsClassifierRulePriLow
            and wmanIfBsClassifierRulePriHigh in order to match
            this rule.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 7."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 24 }

wmanIfBsClassifierRuleVlanId OBJECT-TYPE
        SYNTAX      Integer32 (0..4095)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object applies only to Ethernet frames using the
            802.1P/Q tag header.
            If this object's value is nonzero, tagged packets must
            have a VLAN Identifier that matches the value in order
            to match the rule.
            Only the least significant 12 bits of this object's
            value are valid.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 0."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 25 }

wmanIfBsClassifierRuleState OBJECT-TYPE
        SYNTAX      INTEGER {active(1),
                             inactive(2)}
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object indicates whether or not the classifier is
            enabled to classify packets to a Service Flow.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported
            as active(1)."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfBsClassifierRuleEntry 26 }

wmanIfBsClassifierRulePkts OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object counts the number of packets that have
            been classified using this entry."
        ::= { wmanIfBsClassifierRuleEntry 27 }

wmanIfBsClassifierRuleRowStatus OBJECT-TYPE
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
```

```
            STATUS      current
            DESCRIPTION
                "This object is used to create a new row or modify or
                delete an existing row in this table.

                If the implementator of this MIB has choosen not
                to implement 'dynamic assignment' of profiles, this
                object is not useful and should return noSuchName
                upon SNMP request."
            ::= { wmanIfBsClassifierRuleEntry 28 }

    --
    -- wmanIfBsCps contain the Base Station Common Part Sublayer objects
    wmanIfBsCps OBJECT IDENTIFIER ::= { wmanIfBsObjects 3 }


    --
    -- Base station PKM group
    -- wmanIfBsPkmObjects contain the Base Station Privacy Sublayer objects
    wmanIfBsPkmObjects OBJECT IDENTIFIER ::= { wmanIfBsObjects  4 }

    --
    -- Table wmanIfBsPkmBaseTable
    --
    wmanIfBsPkmBaseTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF   WmanIfBsPkmBaseEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "This table describes the basic PKM attributes of each Base
                 Station wireless interface."
      ::= { wmanIfBsPkmObjects  1 }

      wmanIfBsPkmBaseEntry OBJECT-TYPE
            SYNTAX      WmanIfBsPkmBaseEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "Each entry contains objects describing attributes of one
                 BS wireless interface."
            INDEX       { ifIndex }
      ::= { wmanIfBsPkmBaseTable  1 }

       WmanIfBsPkmBaseEntry ::= SEQUENCE {
            wmanIfBsPkmDefaultAuthLifetime                Integer32,
            wmanIfBsPkmDefaultTEKLifetime                 Integer32,
            wmanIfBsPkmDefaultSelfSignedManufCertTrust    INTEGER,
            wmanIfBsPkmCheckCertValidityPeriods           TruthValue,
            wmanIfBsPkmAuthentInfos                       Counter64,
            wmanIfBsPkmAuthRequests                       Counter64,
            wmanIfBsPkmAuthReplies                        Counter64,
            wmanIfBsPkmAuthRejects                        Counter64,
            wmanIfBsPkmAuthInvalids                       Counter64
            }
```

```
wmanIfBsPkmDefaultAuthLifetime OBJECT-TYPE
      SYNTAX      Integer32 (86400..6048000)
      UNITS       "seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
           "The value of this object is the default lifetime, in
           seconds, the BS assigns to a new authorization key."
      REFERENCE
           "IEEE 802.16 standard; Table 270"
      DEFVAL         { 604800 }
      ::= { wmanIfBsPkmBaseEntry 1 }


wmanIfBsPkmDefaultTEKLifetime OBJECT-TYPE
      SYNTAX      Integer32 (1800..604800)
      UNITS       "seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
           "The value of this object is the default lifetime, in
           seconds, the BS assigns to a new Traffic Encryption
           Key(TEK)."
      REFERENCE
           "IEEE 802.16 standard; Table 270"
      DEFVAL         { 43200 }
      ::= { wmanIfBsPkmBaseEntry 2 }


wmanIfBsPkmDefaultSelfSignedManufCertTrust OBJECT-TYPE
      SYNTAX      INTEGER { trusted (1),
                            untrusted (2) }
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
           "This object determines the default trust of all (new)
           self-signed manufacturer certificates obtained after
           setting the object."
      ::= { wmanIfBsPkmBaseEntry 3 }


wmanIfBsPkmCheckCertValidityPeriods OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
           "Setting this object to TRUE causes all certificates
           received? thereafter to have their validity periods (and
           their chain's validity periods) checked against the current
           time of day. A FALSE setting will cause all certificates
           received? Thereafter to not have their validity periods
           (nor their chain's validity periods) checked against the
```

```
                current time of day."
          ::= { wmanIfBsPkmBaseEntry 4 }


     wmanIfBsPkmAuthentInfos OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                received an Authentication Information message from any
                SS."
          ::= { wmanIfBsPkmBaseEntry 5 }


     wmanIfBsPkmAuthRequests OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                received an Authorization Request message from any SS"
          ::= { wmanIfBsPkmBaseEntry 6 }


     wmanIfBsPkmAuthReplies OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                transmitted an Authorization Reply message to any SS."
          ::= { wmanIfBsPkmBaseEntry 7 }


     wmanIfBsPkmAuthRejects OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                transmitted an Authorization Reject message to any SS."
          ::= { wmanIfBsPkmBaseEntry 8 }


     wmanIfBsPkmAuthInvalids OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                transmitted an Authorization Invalid message to any SS."
          ::= { wmanIfBsPkmBaseEntry 9 }
```

```
--
-- Table wmanIfBsPkmAuthTable
--

wmanIfBsPkmAuthTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF   WmanIfBsPkmAuthEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
           "This table describes the attributes of each SS
           authorization association. The BS maintains one
           authorization association with each Baseline
           Privacy-enabled SS on each BS wireless interface."
::= { wmanIfBsPkmObjects  2 }


wmanIfBsPkmAuthEntry OBJECT-TYPE
      SYNTAX      WmanIfBsPkmAuthEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
           "Each entry contains objects describing attributes of one
           authorization association. The BS MUST create one entry per
           SS per wireless interface, based on the receipt of an
           Authorization Request message, and MUST not delete the
           entry before the SS authorization permanently expires."
      INDEX       { ifIndex, wmanIfBsPkmAuthSsMacAddress }
::= { wmanIfBsPkmAuthTable  1 }


 WmanIfBsPkmAuthEntry ::= SEQUENCE {
      wmanIfBsPkmAuthSsMacAddress                  MacAddress,
      wmanIfBsPkmAuthSsPublicKey                   OCTET STRING,
      wmanIfBsPkmAuthSsKeySequenceNumber           Integer32,
      wmanIfBsPkmAuthSsExpiresOld                  DateAndTime,
      wmanIfBsPkmAuthSsExpiresNew                  DateAndTime,
      wmanIfBsPkmAuthSsLifetime                    Integer32,
      wmanIfBsPkmAuthSsGraceTime                   Integer32,
      wmanIfBsPkmAuthSsReset                       INTEGER,
      wmanIfBsPkmAuthSsInfos                       Counter64,
      wmanIfBsPkmAuthSsRequests                    Counter64,
      wmanIfBsPkmAuthSsReplies                     Counter64,
      wmanIfBsPkmAuthSsRejects                     Counter64,
      wmanIfBsPkmAuthSsInvalids                    Counter64,
      wmanIfBsPkmAuthRejectErrorCode               INTEGER,
      wmanIfBsPkmAuthRejectErrorString             SnmpAdminString,
      wmanIfBsPkmAuthInvalidErrorCode              INTEGER,
      wmanIfBsPkmAuthInvalidErrorString            SnmpAdminString,
      wmanIfBsPkmAuthPrimarySAId                   Integer32,
      wmanIfBsPkmAuthBpkmSsCertValid               INTEGER,
      wmanIfBsPkmAuthBpkmSsCert                    OCTET STRING
      }
```

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

```
                "The value of this object is the actual clock time for
                expiration of the most recent authorization key for this
                FSM"
          ::= { wmanIfBsPkmAuthEntry 5 }


     wmanIfBsPkmAuthSsLifetime OBJECT-TYPE
          SYNTAX      Integer32 (86400..6048000)
          UNITS       "seconds"
          MAX-ACCESS  read-write
          STATUS      current
          DESCRIPTION
                "The vaue of this object is the lifetime, in seconds, the
                BS assigns to an authorization key for this SS."
          REFERENCE
                "IEEE 802.16 standard; Table 270"
          DEFVAL         { 604800 }
          ::= { wmanIfBsPkmAuthEntry 6 }


     wmanIfBsPkmAuthSsGraceTime OBJECT-TYPE
          SYNTAX      Integer32 (300..3024000)
          UNITS       "seconds"
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the grace time for the
                authorization key in seconds.  The SS is expected to start
                trying to get a new authorization key beginning
                AuthGraceTime seconds before the authorization key actually
                expires."
          REFERENCE
                "IEEE 802.16 standard; Table 270"
          DEFVAL         { 600 }
          ::= { wmanIfBsPkmAuthEntry 7 }


     wmanIfBsPkmAuthSsReset OBJECT-TYPE
          SYNTAX      INTEGER { noResetRequested(1),
                                invalidateAuth(2),
                                sendAuthInvalid(3),
                                invalidateTeks(4) }
          MAX-ACCESS  read-write
          STATUS      current
          DESCRIPTION
                "Setting this object to invalidateAuth(2) causes the BS to
                invalidate the current SS authorization key(s), but not to
                transmit an Authorization Invalid message nor to invalidate
                unicast TEKs. Setting this object to sendAuthInvalid(3)
                causes the BS to invalidate the current SS authorization
                key(s), and to transmit an Authorization Invalid message to
                the SS, but not to invalidate unicast TEKs. Setting this
                object to invalidateTeks(4) causes the BS to invalidate the
                current SS authorization key(s), to transmit an
```

```
                Authorization Invalid message to the SS, and to
                invalidate all unicast TEKs associated with this SS
                authorization. Reading this object returns the
                most-recently-set value of this object, or returns
                noResetRequested(1) if the object has not been set since
                the last BS reboot."
          ::= { wmanIfBsPkmAuthEntry 8 }


     wmanIfBsPkmAuthSsInfos OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                received an Authentication Information message from this
                SS."
          ::= { wmanIfBsPkmAuthEntry 9 }


     wmanIfBsPkmAuthSsRequests OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                received an Authorization Request message from this SS."
          ::= { wmanIfBsPkmAuthEntry 10 }


     wmanIfBsPkmAuthSsReplies OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                transmitted an Authorization Reply message to this SS."
          ::= { wmanIfBsPkmAuthEntry 11 }


     wmanIfBsPkmAuthSsRejects OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
                "The value of this object is the count of times the BS has
                transmitted an Authorization Reject message to this SS."
          ::= { wmanIfBsPkmAuthEntry 12 }


     wmanIfBsPkmAuthSsInvalids OBJECT-TYPE
          SYNTAX      Counter64
          MAX-ACCESS  read-only
          STATUS      current
```

```
            DESCRIPTION
                 "The value of this object is the count of times the BS has
                 transmitted an Authorization Invalid message to this SS."
            ::= { wmanIfBsPkmAuthEntry 13 }


      wmanIfBsPkmAuthRejectErrorCode OBJECT-TYPE
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 unauthorizedSs(3),
                                 unauthorizedSaid(4),
                                 permanentAuthorizationFailure(8),
                                 timeOfDayNotAcquired(11)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the enumerated description of
                 the Error-Code in most recent Authorization Reject message
                 transmitted to the SS.  This has value unknown(2) if the
                 last Error-Code value was 0, and none(1) if no
                 Authorization Reject message has been transmitted to the
                 SS."
            ::= { wmanIfBsPkmAuthEntry 14 }


      wmanIfBsPkmAuthRejectErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Display-String in most
                 recent Authorization Reject message transmitted to the SS.
                 This is a zero length string if no Authorization Reject
                 message has been transmitted to the SS."
            ::= { wmanIfBsPkmAuthEntry 15 }


      wmanIfBsPkmAuthInvalidErrorCode OBJECT-TYPE
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 unauthorizedSs(3),
                                 unsolicited(5),
                                 invalidKeySequence(6),
                                 keyRequestAuthenticationFailure(7)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The value of this object is the enumerated description of
                 the Error-Code in most recent Authorization Invalid message
                 transmitted to the SS.  This has value unknown(2) if the
                 last Error-Code value was 0, and none(1) if no
                 Authorization Invalid message has been transmitted to the
                 SS."
            ::= { wmanIfBsPkmAuthEntry 16 }
```

```
1
2
3       wmanIfBsPkmAuthInvalidErrorString OBJECT-TYPE
4             SYNTAX      SnmpAdminString (SIZE (0..128))
5             MAX-ACCESS  read-only
6             STATUS      current
7             DESCRIPTION
8                   "The value of this object is the Display-String in most
9                   recent Authorization Invalid message transmitted to the SS.
10                  This is a zero length string if no Authorization Invalid
11                  message has been transmitted to the SS."
12            ::= { wmanIfBsPkmAuthEntry 17 }
13
14
15      wmanIfBsPkmAuthPrimarySAId OBJECT-TYPE
16            SYNTAX      Integer32 (0..16383)
17            MAX-ACCESS  read-only
18            STATUS      current
19            DESCRIPTION
20                  "The value of this object is the Primary Security
21                  Association identifier."
22            ::= { wmanIfBsPkmAuthEntry 18 }
23
24
25      wmanIfBsPkmAuthBpkmSsCertValid OBJECT-TYPE
26            SYNTAX      INTEGER {unknown (0),
27                                 validSsChained (1),
28                                 validSsTrusted (2),
29                                 invalidSsUntrusted (3),
30                                 invalidCAUntrusted (4),
31                                 invalidSsOther (5),
32                                 invalidCAOther (6) }
33            MAX-ACCESS  read-only
34            STATUS      current
35            DESCRIPTION
36                  "Contains the reason why a SS's certificate is deemed valid
37                  or invalid. Return unknown if the SS is running PKM mode.
38                  ValidSsChained means the certificate is valid because it
39                  chains to a valid certificate. ValidSsTrusted means the
40                  certificate is valid because it has been provisioned to be
41                  trusted. InvalidSsUntrusted means the certificate is
42                  invalid because it has been provisioned to be untrusted.
43                  InvalidCAUntrusted means the certificate is invalid
44                  because   it chains to an untrusted certificate.
45                  InvalidSsOther and InvalidCAOther refer to errors in
46                  parsing, validity periods, etc, which are attributable to
47                  the SS certificate or its chain respectively."
48            ::= { wmanIfBsPkmAuthEntry 19 }
49
50
51      wmanIfBsPkmAuthBpkmSsCert OBJECT-TYPE
52            SYNTAX      OCTET STRING
53            MAX-ACCESS  read-only
54            STATUS      current
```

```
            DESCRIPTION
                 "The X509 SS Certificate sent as part of a PKM
                 Authorization Request."
            ::= { wmanIfBsPkmAuthEntry 20 }


      --
      -- Table wmanIfBsPkmTEKTable
      --

      wmanIfBsPkmTEKTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF   WmanIfBsPkmTEKEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "This table describes the attributes of each Traffic
                 Encryption Key (TEK) association. The BS maintains one TEK
                 association per SAID on each BS wireless interface."
      ::= { wmanIfBsPkmObjects  3 }


      wmanIfBsPkmTEKEntry OBJECT-TYPE
            SYNTAX      WmanIfBsPkmTEKEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "Each entry contains objects describing attributes of one
                 TEK association on a particular BS wireless interface. The
                 BS MUST create one entry per SAID per wireless interface,
                 based on the receipt of a Key Request message, and MUST not
                 delete the entry before the SS authorization for the SAID
                 permanently expires."
            INDEX       { ifIndex, wmanIfBsPkmTEKSAId }
      ::= { wmanIfBsPkmTEKTable  1 }


       WmanIfBsPkmTEKEntry ::= SEQUENCE {
            wmanIfBsPkmTEKSAId                          Integer32,
            wmanIfBsPkmTEKSAType                        INTEGER,
            wmanIfBsPkmTEKDataEncryptAlg                INTEGER,
            wmanIfBsPkmTEKDataAuthentAlg                INTEGER,
            wmanIfBsPkmTEKEncryptAlg                    INTEGER,
            wmanIfBsPkmTEKLifetime                      Integer32,
            wmanIfBsPkmTEKGraceTime                     Integer32,
            wmanIfBsPkmTEKKeySequenceNumber             Integer32,
            wmanIfBsPkmTEKExpiresOld                    DateAndTime,
            wmanIfBsPkmTEKExpiresNew                    DateAndTime,
            wmanIfBsPkmTEKReset                         TruthValue,
            wmanIfBsPkmKeyRequests                      Counter64,
            wmanIfBsPkmKeyReplies                       Counter64,
            wmanIfBsPkmKeyRejects                       Counter64,
            wmanIfBsPkmTEKInvalids                      Counter64,
            wmanIfBsPkmKeyRejectErrorCode               INTEGER,
            wmanIfBsPkmKeyRejectErrorString             SnmpAdminString,
```

```
        wmanIfBsPkmTEKInvalidErrorCode                INTEGER,
        wmanIfBsPkmTEKInvalidErrorString              SnmpAdminString
        }


    wmanIfBsPkmTEKSAId OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
             "The value of this object is the WiMAX Security Association
             ID (SAID)."
        ::= { wmanIfBsPkmTEKEntry 1 }


    wmanIfBsPkmTEKSAType OBJECT-TYPE
        SYNTAX      INTEGER {none(0),
                             primary(1),
                             static(2),
                             dynamic(3) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the type of security
             association. Dynamic does not apply to SSs running in PKM
             mode."
        ::= { wmanIfBsPkmTEKEntry 2 }


    wmanIfBsPkmTEKDataEncryptAlg OBJECT-TYPE
        SYNTAX      INTEGER { none(0),
                             des56CbcMode(1) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the data encryption algorithm
             being utilized."
        REFERENCE
             "IEEE 802.16 standard; Table 301"
        ::= { wmanIfBsPkmTEKEntry 3 }


    wmanIfBsPkmTEKDataAuthentAlg OBJECT-TYPE
        SYNTAX      INTEGER { none(0) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the data authentication
             algorithm being utilized."
        REFERENCE
             "IEEE 802.16 standard; Table 302"
        ::= { wmanIfBsPkmTEKEntry 4 }
```

```
wmanIfBsPkmTEKEncryptAlg OBJECT-TYPE
      SYNTAX      INTEGER { tripleDES(0),
                            rsa1024(1) }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
           "The value of this object is the TEK key encryption
           algorithm being utilized."
      REFERENCE
           "IEEE 802.16 standard; Table 303"
      ::= { wmanIfBsPkmTEKEntry 5 }


wmanIfBsPkmTEKLifetime OBJECT-TYPE
      SYNTAX      Integer32 (1800..604800)
      UNITS       "seconds"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
           "The value of this object is the lifetime, in seconds, the
           BS assigns to keys for this TEK association."
      REFERENCE
           "IEEE 802.16 standard; Table 270"
      DEFVAL         { 43200 }
      ::= { wmanIfBsPkmTEKEntry 6 }


wmanIfBsPkmTEKGraceTime OBJECT-TYPE
      SYNTAX      Integer32 (300..302399)
      UNITS       "seconds"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
           "The value of this object is the grace time for the TEK in
           seconds. The SS is expected to start trying to acquire a
           new TEK beginning TEK GraceTime seconds before the TEK
           actually expires."
      REFERENCE
           "IEEE 802.16 standard; Table 270"
      DEFVAL         { 3600 }
      ::= { wmanIfBsPkmTEKEntry 7 }


wmanIfBsPkmTEKKeySequenceNumber OBJECT-TYPE
      SYNTAX      Integer32 (0..15)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The value of this object is the most recent TEK key
           sequence number for this SAID."
      ::= { wmanIfBsPkmTEKEntry 8 }


wmanIfBsPkmTEKExpiresOld OBJECT-TYPE
```

```
            SYNTAX      DateAndTime
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the actual clock time for
                 expiration of the immediate predecessor of the most recent
                 TEK for this FSM. If this FSM has only one TEK, then the
                 value is the time of activation of this FSM."
            ::= { wmanIfBsPkmTEKEntry 9 }


      wmanIfBsPkmTEKExpiresNew OBJECT-TYPE
            SYNTAX      DateAndTime
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the actual clock time for
                 expiration of the most recent TEK for this FSM."
            ::= { wmanIfBsPkmTEKEntry 10 }


      wmanIfBsPkmTEKReset OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                 "Setting this object to TRUE causes the BS to invalidate
                 the current active TEK(s) (plural due to key transition
                 periods), and to generate a new TEK for the associated
                 SAID; the BS MAY also generate an unsolicited TEK Invalid
                 message, to optimize the TEK synchronization between the BS
                 and the SS. Reading this object always returns   FALSE."
            ::= { wmanIfBsPkmTEKEntry 11 }


      wmanIfBsPkmKeyRequests OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The value of this object is the count of times the BS has
                 received a Key Request message."
            ::= { wmanIfBsPkmTEKEntry 12 }


      wmanIfBsPkmKeyReplies OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the BS has
                 transmitted a Key Reply message."
            ::= { wmanIfBsPkmTEKEntry 13 }
```

```
1
2       wmanIfBsPkmKeyRejects OBJECT-TYPE
3             SYNTAX      Counter64
4             MAX-ACCESS  read-only
5             STATUS      current
6             DESCRIPTION
7                   "The value of this object is the count of times the BS has
8                   transmitted a Key Reject message."
9             ::= { wmanIfBsPkmTEKEntry 14 }
10
11
12      wmanIfBsPkmTEKInvalids OBJECT-TYPE
13            SYNTAX      Counter64
14            MAX-ACCESS  read-only
15            STATUS      current
16            DESCRIPTION
17                  "The value of this object is the count of times the BS has
18                  transmitted a TEK Invalid message."
19            ::= { wmanIfBsPkmTEKEntry 15 }
20
21
22      wmanIfBsPkmKeyRejectErrorCode OBJECT-TYPE
23            SYNTAX      INTEGER {none(1),
24                                 unknown(2),
25                                 unauthorizedSaid(4)}
26            MAX-ACCESS  read-only
27            STATUS      current
28            DESCRIPTION
29                  "The value of this object is the enumerated; description of
30                  the Error-Code in the most recent Key Reject message sent
31                  in response to a Key Request for this SAID. This has value
32                  unknown(2) if the last Error-Code value was 0, and none(1)
33                  if no Key Reject message has been received since reboot."
34            ::= { wmanIfBsPkmTEKEntry 16 }
35
36
37      wmanIfBsPkmKeyRejectErrorString OBJECT-TYPE
38            SYNTAX      SnmpAdminString (SIZE (0..128))
39            MAX-ACCESS  read-only
40            STATUS      current
41            DESCRIPTION
42                  "The value of this object is the Display-String in the most
43                  recent Key Reject message sent in response to a Key Request
44                  for this SAID. This is a zero length string if no Key
45                  Reject message has been received since reboot."
46            ::= { wmanIfBsPkmTEKEntry 17 }
47
48
49      wmanIfBsPkmTEKInvalidErrorCode OBJECT-TYPE
50            SYNTAX      INTEGER  {none(1),
51                                 unknown(2),
52                                 invalidKeySequence(6)}
53            MAX-ACCESS  read-only
54            STATUS      current
```

```
            DESCRIPTION
                "The value of this object is the enumerated description of
                 the Error-Code in the most recent TEK Invalid message sent
                 in association with this SAID.  This has value unknown(2)
                 if the last Error-Code value was 0, and none(1) if no TEK
                 Invalid message has been received since reboot."
            ::= { wmanIfBsPkmTEKEntry 18 }


      wmanIfBsPkmTEKInvalidErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Display-String in the most
                 recent TEK Invalid message sent in association with this
                 SAID.  This is a zero length string if no TEK Invalid
                 message has been received since reboot."
            ::= { wmanIfBsPkmTEKEntry 19 }

    --
    -- SS object group - containing tables and objects to be implemented in
    -- the Subscriber station
    --
    -- wmanIfSsSystem contain the Subscriber Station System objects
    wmanIfSsSystem OBJECT IDENTIFIER ::= { wmanIfSsObjects 1 }

    wmanIfSsConfigFileEncodingTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF WmanIfSsConfigFileEncodingEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "This table contains configuration file encoding
                information of the SS."
            REFERENCE
                "Section 11.2 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfSsSystem 1 }

    wmanIfSsConfigFileEncodingEntry OBJECT-TYPE
            SYNTAX      WmanIfSsConfigFileEncodingEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                  "This table provides one row for each SS, and is indexed
                  by BS ifIndex. An entry in this table exists for each
                  ifEntry of SS with an ifType of wmanMac"
            INDEX { ifIndex }
            ::= { wmanIfSsConfigFileEncodingTable 1 }

    WmanIfSsConfigFileEncodingEntry ::= SEQUENCE {
            wmanIfSsMicConfigSetting        OCTET STRING,
            wmanIfSsVendorId                OCTET STRING,
            wmanIfSsHwId                    OCTET STRING,
            wmanIfSsSwVersion               OCTET STRING,
```

```
        wmanIfSsUpgradeFileName         OCTET STRING,
        wmanIfSsSwUpgradeTftpServer     InetAddress,
        wmanIfSsTftpServerTimeStamp     DateAndTime
        }

wmanIfSsMicConfigSetting OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(16))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value field contains the SS MIC code. This is used
            to detect unauthorized modification or corruption of
            the configuration file."
        ::= { wmanIfSsConfigFileEncodingEntry 1 }

wmanIfSsVendorId OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE(3))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This value identifies the managed SS vendor to which the
            software upgrade is to be applied."
        ::= { wmanIfSsConfigFileEncodingEntry 2 }

wmanIfSsHwId OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This value identifies the hardware version to which the
            software upgrade is to be applied."
        ::= { wmanIfSsConfigFileEncodingEntry 3 }

wmanIfSsSwVersion OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value field contains the SS MIC code. This is used
            to detect unauthorized modification or corruption of
            the configuration file."
        ::= { wmanIfSsConfigFileEncodingEntry 4 }

wmanIfSsUpgradeFileName OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The filename is a fully qualified directory path
            name which is in a format appropriate to the server."
        ::= { wmanIfSsConfigFileEncodingEntry 5 }

wmanIfSsSwUpgradeTftpServer OBJECT-TYPE
        SYNTAX      InetAddress
```

```
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "This object is the IP address of the TFTP server on
                which the software upgrade file for the SS resides."
            ::= { wmanIfSsConfigFileEncodingEntry 6 }

    wmanIfSsTftpServerTimeStamp OBJECT-TYPE
            SYNTAX      DateAndTime
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "This is the sending time of the configuration file in
    seconds.
                The definition of time is as in IETF RFC 868."
            ::= { wmanIfSsConfigFileEncodingEntry 7 }

    --
    -- wmanIfSsCps contain the Base Station Common Part Sublayer objects
    wmanIfSsCps OBJECT IDENTIFIER ::= { wmanIfSsObjects 2 }



    -- Subscriber station PKM  group
    -- wmanIfSsPkmObjects contain the Subscriber Station Privacy Sublayer
    -- objects
    wmanIfSsPkmObjects OBJECT IDENTIFIER ::= { wmanIfSsObjects  3 }


      --
      -- Table wmanIfSsPkmBaseTable
      --

      wmanIfSsPkmBaseTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF   WmanIfSsPkmBaseEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "This table describes the basic and authorization related
                 PKM attributes of each SS wireless interface."
      ::= { wmanIfSsPkmObjects  1 }


      wmanIfSsPkmBaseEntry OBJECT-TYPE
            SYNTAX      WmanIfSsPkmBaseEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "Each entry contains objects describing attributes of one
                 SS wireless interface."
            INDEX       { ifIndex }
      ::= { wmanIfSsPkmBaseTable  1 }
```

```
 1       WmanIfSsPkmBaseEntry ::= SEQUENCE {
 2            wmanIfSsPkmPrivacyEnable                    TruthValue,
 3            wmanIfSsPkmPublicKey                        OCTET STRING,
 4            wmanIfSsPkmAuthState                        INTEGER,
 5            wmanIfSsPkmAuthKeySequenceNumber            Integer32,
 6            wmanIfSsPkmAuthExpiresOld                   DateAndTime,
 7            wmanIfSsPkmAuthExpiresNew                   DateAndTime,
 8            wmanIfSsPkmAuthReset                        TruthValue,
 9            wmanIfSsPkmAuthGraceTime                    Integer32,
10            wmanIfSsPkmTEKGraceTime                     Integer32,
11            wmanIfSsPkmAuthWaitTimeout                  Integer32,
12            wmanIfSsPkmReauthWaitTimeout                Integer32,
13            wmanIfSsPkmOpWaitTimeout                    Integer32,
14            wmanIfSsPkmRekeyWaitTimeout                 Integer32,
15            wmanIfSsPkmAuthRejectWaitTimeout            Integer32,
16            wmanIfSsPkmAuthentInfos                     Counter64,
17            wmanIfSsPkmAuthRequests                     Counter64,
18            wmanIfSsPkmAuthReplies                      Counter64,
19            wmanIfSsPkmAuthRejects                      Counter64,
20            wmanIfSsPkmAuthInvalids                     Counter64,
21            wmanIfSsPkmAuthRejectErrorCode              INTEGER,
22            wmanIfSsPkmAuthRejectErrorString            SnmpAdminString,
23            wmanIfSsPkmAuthInvalidErrorCode             INTEGER,
24            wmanIfSsPkmAuthInvalidErrorString           SnmpAdminString
25            }
26
27
28       wmanIfSsPkmPrivacyEnable OBJECT-TYPE
29            SYNTAX      TruthValue
30            MAX-ACCESS  read-only
31            STATUS      current
32            DESCRIPTION
33                 "This object identifies whether this SS is provisioned to
34                 run Baseline Privacy Plus."
35            ::= { wmanIfSsPkmBaseEntry 1 }
36
37
38       wmanIfSsPkmPublicKey OBJECT-TYPE
39            SYNTAX      OCTET STRING (SIZE (140))
40            MAX-ACCESS  read-only
41            STATUS      current
42            DESCRIPTION
43                 "The value of this object is a DER-encoded RSAPublicKey
44                 ASN.1 type string, as defined in the RSA Encryption
45                 Standard (PKCS#1) [10], corresponding to the public key of
46                 the SS. The 74, 106, 140, 204, and 270 byte key encoding
47                 lengths correspond to 512 bit, 768 bit, 1024 bit, 1536 bit,
48                 and 2048 public moduli respectively."
49            ::= { wmanIfSsPkmBaseEntry 2 }
50
51
52       wmanIfSsPkmAuthState OBJECT-TYPE
53            SYNTAX      INTEGER {start(1),
54                                 authWait(2),
```

```
                                authorized(3),
                                reauthWait(4),
                                authRejectWait(5),
                                silent(6)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the state of the SS
                 authorization FSM.  The start state indicates that FSM is
                 in its initial state."
            ::= { wmanIfSsPkmBaseEntry 3 }


      wmanIfSsPkmAuthKeySequenceNumber OBJECT-TYPE
            SYNTAX      Integer32 (0..15)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the most recent authorization
                 key sequence number for this FSM."
            ::= { wmanIfSsPkmBaseEntry 4 }


      wmanIfSsPkmAuthExpiresOld OBJECT-TYPE
            SYNTAX      DateAndTime
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the actual clock time for
                 expiration of the immediate predecessor of the most recent
                 authorization key for this FSM.  If this FSM has only one
                 authorization key, then the value is the time of activation
                 of this FSM."
            ::= { wmanIfSsPkmBaseEntry 5 }


      wmanIfSsPkmAuthExpiresNew OBJECT-TYPE
            SYNTAX      DateAndTime
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the actual clock time for
                 expiration of the most recent authorization key for this
                 FSM."
            ::= { wmanIfSsPkmBaseEntry 6 }


      wmanIfSsPkmAuthReset OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Setting this object to TRUE generates a Reauthorize event
                 in the authorization FSM. Reading this object always
```

```
                returns FALSE."
          ::= { wmanIfSsPkmBaseEntry 7 }


     wmanIfSsPkmAuthGraceTime OBJECT-TYPE
          SYNTAX      Integer32 (300..3024000)
          UNITS       "seconds"
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
               "The value of this object is the grace time for an
               authorization key.  A SS is expected to start trying to get
               a new authorization key beginning AuthGraceTime seconds
               before the authorization key actually expires."
          REFERENCE
               "IEEE 802.16 standard; Table 270"
          DEFVAL         { 600 }
          ::= { wmanIfSsPkmBaseEntry 8 }


     wmanIfSsPkmTEKGraceTime OBJECT-TYPE
          SYNTAX      Integer32 (300..3024000)
          UNITS       "seconds"
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
               "The value of this object is the grace time for the TEK in
               seconds. The SS is expected to start trying to acquire a
               new TEK beginning TEK GraceTime seconds before the
               expiration of the most recent TEK."
          REFERENCE
               "IEEE 802.16 standard; Table 270"
          DEFVAL         { 3600 }
          ::= { wmanIfSsPkmBaseEntry 9 }


     wmanIfSsPkmAuthWaitTimeout OBJECT-TYPE
          SYNTAX      Integer32 (2..30)
          UNITS       "seconds"
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
               "The value of this object is the Authorize Wait Timeout."
          REFERENCE
               "IEEE 802.16 standard; Table 270"
          DEFVAL         { 10 }
          ::= { wmanIfSsPkmBaseEntry 10 }


     wmanIfSsPkmReauthWaitTimeout OBJECT-TYPE
          SYNTAX      Integer32 (2..30)
          UNITS       "seconds"
          MAX-ACCESS  read-only
          STATUS      current
```

```
            DESCRIPTION
                 "The value of this object is the Reauthorize Wait Timeout
                 in seconds."
            REFERENCE
                 "IEEE 802.16 standard; Table 270"
            DEFVAL         { 10 }
            ::= { wmanIfSsPkmBaseEntry 11 }


      wmanIfSsPkmOpWaitTimeout OBJECT-TYPE
            SYNTAX      Integer32 (1..10)
            UNITS       "seconds"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Operational Wait Timeout
                 in seconds."
            REFERENCE
                 "IEEE 802.16 standard; Table 270"
            DEFVAL         { 1 }
            ::= { wmanIfSsPkmBaseEntry 12 }


      wmanIfSsPkmRekeyWaitTimeout OBJECT-TYPE
            SYNTAX      Integer32 (1..10)
            UNITS       "seconds"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Rekey Wait Timeout in
                 seconds."
            REFERENCE
                 "IEEE 802.16 standard; Table 270"
            DEFVAL         { 1 }
            ::= { wmanIfSsPkmBaseEntry 13 }


      wmanIfSsPkmAuthRejectWaitTimeout OBJECT-TYPE
            SYNTAX      Integer32 (10..600)
            UNITS       "seconds"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Authorization Reject Wait
                 Timeout in seconds."
            REFERENCE
                 "IEEE 802.16 standard; Table 270"
            DEFVAL         { 60 }
            ::= { wmanIfSsPkmBaseEntry 14 }


      wmanIfSsPkmAuthentInfos OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
```

```
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the SS has
                 transmitted an Authentication Information message."
            ::= { wmanIfSsPkmBaseEntry 15 }


      wmanIfSsPkmAuthRequests OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the SS has
                 transmitted an Authorization Request message."
            ::= { wmanIfSsPkmBaseEntry 16 }


      wmanIfSsPkmAuthReplies OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the SS has
                 received an Authorization Reply message."
            ::= { wmanIfSsPkmBaseEntry 17 }


      wmanIfSsPkmAuthRejects OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the SS has
                 received an Authorization Reject message."
            ::= { wmanIfSsPkmBaseEntry 18 }


      wmanIfSsPkmAuthInvalids OBJECT-TYPE
            SYNTAX      Counter64
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the count of times the SS has
                 received an Authorization Invalid message."
            ::= { wmanIfSsPkmBaseEntry 19 }


      wmanIfSsPkmAuthRejectErrorCode OBJECT-TYPE
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 unauthorizedSs(3),
                                 unauthorizedSaid(4),
                                 permanentAuthorizationFailure(8),
                                 timeOfDayNotAcquired(11)}
```

```
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the enumerated description of
                 the Error-Code in most recent Authorization Reject message
                 received by the SS.  This has value unknown(2)if the last
                 Error-Code value was 0, and none(1) if no Authorization
                 Reject message has been received since reboot."
            ::= { wmanIfSsPkmBaseEntry 20 }


      wmanIfSsPkmAuthRejectErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The value of this object is the Display-String in most
                 recent Authorization Reject message received by the SS.
                 This is a zero length string if no Authorization Reject
                 message has been received since reboot."
            ::= { wmanIfSsPkmBaseEntry 21 }


      wmanIfSsPkmAuthInvalidErrorCode OBJECT-TYPE
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 unauthorizedSs(3),
                                 unsolicited(5),
                                 invalidKeySequence(6),
                                 keyRequestAuthenticationFailure(7)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the enumerated description of
                 the Error-Code in most recent Authorization Invald message
                 received by the SS.  This has value unknown(2) if the last
                 Error-Code value was 0, and none(1) if no Authorization
                 Invalid message has been received since reboot."
            ::= { wmanIfSsPkmBaseEntry 22 }


      wmanIfSsPkmAuthInvalidErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Display-String in most
                 recent Authorization Invalid message received by the SS.
                 This is a zero length string if no Authorization Invalid
                 message has been received since reboot."
            ::= { wmanIfSsPkmBaseEntry 23 }


      --
```

```
-- Table wmanIfSsPkmTEKTable
--

wmanIfSsPkmTEKTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF   WmanIfSsPkmTEKEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
           "This table describes the attributes of each SS Traffic
           Encryption Key(TEK) association. The SS maintains (no more
           than) one TEK association per SAID per SS wireless
           interface."
::= { wmanIfSsPkmObjects  2 }


wmanIfSsPkmTEKEntry OBJECT-TYPE
      SYNTAX      WmanIfSsPkmTEKEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
           "Each entry contains objects describing the TEK association
           attributes of one SAID. The SS MUST create one entry per
           SAID, regardless of whether the SAID was obtained from a
            Registration Response message, from an Authorization Reply
            message, or from any dynamic SAID establishment
            mechanisms."
      INDEX       { ifIndex, wmanIfSsPkmTEKSAId }
::= { wmanIfSsPkmTEKTable  1 }


 WmanIfSsPkmTEKEntry ::= SEQUENCE {
      wmanIfSsPkmTEKSAId                          Integer32,
      wmanIfSsPkmTEKSAType                        INTEGER,
      wmanIfSsPkmTEKDataEncryptAlg                INTEGER,
      wmanIfSsPkmTEKDataAuthentAlg                INTEGER,
      wmanIfSsPkmTEKEncryptAlg                    INTEGER,
      wmanIfSsPkmTEKState                         INTEGER,
      wmanIfSsPkmTEKKeySequenceNumber             Integer32,
      wmanIfSsPkmTEKExpiresOld                    DateAndTime,
      wmanIfSsPkmTEKExpiresNew                    DateAndTime,
      wmanIfSsPkmTEKKeyRequests                   Counter64,
      wmanIfSsPkmTEKKeyReplies                    Counter64,
      wmanIfSsPkmTEKKeyRejects                    Counter64,
      wmanIfSsPkmTEKInvalids                      Counter64,
      wmanIfSsPkmTEKAuthPends                     Counter64,
      wmanIfSsPkmTEKKeyRejectErrorCode            INTEGER,
      wmanIfSsPkmTEKKeyRejectErrorString          SnmpAdminString,
      wmanIfSsPkmTEKInvalidErrorCode              INTEGER,
      wmanIfSsPkmTEKInvalidErrorString            SnmpAdminString
      }


wmanIfSsPkmTEKSAId OBJECT-TYPE
      SYNTAX      Integer32 (1..16383)
```

```
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "The value of this object is the WiMAX Security Association
                 ID (SAID)."
            ::= { wmanIfSsPkmTEKEntry 1 }


      wmanIfSsPkmTEKSAType OBJECT-TYPE
            SYNTAX      INTEGER {none(0),
                                 primary(1),
                                 static(2),
                                 dynamic(3)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the type of security
                 association."
            ::= { wmanIfSsPkmTEKEntry 2 }


      wmanIfSsPkmTEKDataEncryptAlg OBJECT-TYPE
            SYNTAX      INTEGER { none(0),
                                  des56CbcMode(1) }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the data encryption algorithm
                 being utilized."
            REFERENCE
                 "IEEE 802.16 standard; Table 301"
            ::= { wmanIfSsPkmTEKEntry 3 }


      wmanIfSsPkmTEKDataAuthentAlg OBJECT-TYPE
            SYNTAX      INTEGER { none(0) }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the data authentication
                 algorithm being utilized."
            REFERENCE
                 "IEEE 802.16 standard; Table 302"
            ::= { wmanIfSsPkmTEKEntry 4 }


      wmanIfSsPkmTEKEncryptAlg OBJECT-TYPE
            SYNTAX      INTEGER { tripleDES(0),
                                  rsa1024(1) }
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the TEK key encryption
                 algorithm for this cryptographic suite capability."
```

```
        REFERENCE
             "IEEE 802.16 standard; Table 303"
        ::= { wmanIfSsPkmTEKEntry 5 }


    wmanIfSsPkmTEKState OBJECT-TYPE
        SYNTAX      INTEGER { start(1),
                              opWait(2),
                              opReauthWait(3),
                              operational(4),
                              rekeyWait(5),
                              rekeyReauthWait(6) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the state of the indicated TEK
             FSM. The start(1) state indicates that FSM is in its
             initial state."
        ::= { wmanIfSsPkmTEKEntry 6 }


    wmanIfSsPkmTEKKeySequenceNumber OBJECT-TYPE
        SYNTAX      Integer32 (0..15)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the most recent TEK key
             sequence number for this TEK FSM."
        ::= { wmanIfSsPkmTEKEntry 7 }


    wmanIfSsPkmTEKExpiresOld OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of this object is the actual clock time for
             expiration of the immediate predecessor of the most recent
             TEK for this FSM.  If this FSM has only one TEK, then the
             value is the time of activation of this FSM."
        ::= { wmanIfSsPkmTEKEntry 8 }


    wmanIfSsPkmTEKExpiresNew OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the actual clock time for
             expiration of the most recent TEK for this FSM."
        ::= { wmanIfSsPkmTEKEntry 9 }


    wmanIfSsPkmTEKKeyRequests OBJECT-TYPE
```

```
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the count of times the SS has
             transmitted a Key Request message."
        ::= { wmanIfSsPkmTEKEntry 10 }


    wmanIfSsPkmTEKKeyReplies OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of this object is the count of times the SS has
             received a Key Reply message, including a message whose
             authentication failed."
        ::= { wmanIfSsPkmTEKEntry 11 }


    wmanIfSsPkmTEKKeyRejects OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the count of times the SS has
             received a Key Reject message, including a message whose
             authentication failed."
        ::= { wmanIfSsPkmTEKEntry 12 }


    wmanIfSsPkmTEKInvalids OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the count of times the SS has
             received a TEK Invalid message, including a message whose
             authentication failed."
        ::= { wmanIfSsPkmTEKEntry 13 }


    wmanIfSsPkmTEKAuthPends OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the count of times an
             Authorization Pending (Auth Pend) event occurred in this
             FSM."
        ::= { wmanIfSsPkmTEKEntry 14 }


    wmanIfSsPkmTEKKeyRejectErrorCode OBJECT-TYPE
```

```
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 unauthorizedSaid(4)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the enumerated description of
                 the Error-Code in most recent Key Reject message received
                 by the SS. This has value unknown(2) if the last Error-Code
                 value was 0, and none(1) if no Key Reject message has been
                 received since reboot."
            ::= { wmanIfSsPkmTEKEntry 15 }


      wmanIfSsPkmTEKKeyRejectErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the Display-String in most
                 recent Key Reject message received by the SS. This is a
                 zero length string if no Key Reject message has been
                 received since reboot."
            ::= { wmanIfSsPkmTEKEntry 16 }


      wmanIfSsPkmTEKInvalidErrorCode OBJECT-TYPE
            SYNTAX      INTEGER {none(1),
                                 unknown(2),
                                 invalidKeySequence(6)}
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                 "The value of this object is the enumerated description of
                 the Error-Code in most recent TEK Invalid message received
                 by the SS.  This has value unknown(2) if the last
                 Error-Code value was 0, and  none(1) if no TEK Invalid
                 message has been received since reboot."
            ::= { wmanIfSsPkmTEKEntry 17 }


      wmanIfSsPkmTEKInvalidErrorString OBJECT-TYPE
            SYNTAX      SnmpAdminString (SIZE (0..128))
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The value of this object is the Display-String in most
                 recent TEK Invalid message received by the SS. This is a
                 zero length string if no TEK Invalid message has been
                 received since reboot."
            ::= { wmanIfSsPkmTEKEntry 18 }


      --
```

```
    -- Table wmanIfSsDeviceCertTable
    --

    wmanIfSsDeviceCertTable OBJECT-TYPE
          SYNTAX      SEQUENCE OF   WmanIfSsDeviceCertEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
               "This table describes the PKM device certificates for each
               SS wireless interface."
    ::= { wmanIfSsPkmObjects  3 }


    wmanIfSsDeviceCertEntry OBJECT-TYPE
          SYNTAX      WmanIfSsDeviceCertEntry
          MAX-ACCESS  not-accessible
          STATUS      current
          DESCRIPTION
               "Each entry contains the device certificate of one SS."
          INDEX       { ifIndex }
    ::= { wmanIfSsDeviceCertTable  1 }


     WmanIfSsDeviceCertEntry ::= SEQUENCE {
          wmanIfSsDeviceCert           OCTET STRING,
          wmanIfSsDeviceManufCert      OCTET STRING
          }


    wmanIfSsDeviceCert OBJECT-TYPE
          SYNTAX      OCTET STRING
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
               "The X509 DER-encoded subscriber station certificate."
          ::= { wmanIfSsDeviceCertEntry 1 }


    wmanIfSsDeviceManufCert OBJECT-TYPE
          SYNTAX      OCTET STRING
          MAX-ACCESS  read-only
          STATUS      current
          DESCRIPTION
               "The X509 DER-encoded manufacturer certificate which is
               signed by the CA root authority certificate."
          ::= { wmanIfSsDeviceCertEntry 2 }



  --
  -- Common object group - containing common tables and objects to be
  -- implemented in both Base Station and Subscriber Station
  --
```

```
-- wmanIfCmnPacketCs contain the Packet Convergence Sublayer objects
-- that are common to both Base Station and Subscriber Station
wmanIfCmnPacketCs OBJECT IDENTIFIER ::= { wmanIfCommonObjects 1 }


--
-- wmanIfCmnCps contain the Common Part Sublayer objects that are common
-- to both Base Station and Subscriber Station
wmanIfCmnCps OBJECT IDENTIFIER ::= { wmanIfCommonObjects 2 }

wmanIfCmnCpsServiceFlowTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF WmanIfCmnCpsServiceFlowEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table contains Service Flows that are created in both
            BS and SS."
        ::= { wmanIfCmnCps 1 }

wmanIfCmnCpsServiceFlowEntry OBJECT-TYPE
        SYNTAX      WmanIfCmnCpsServiceFlowEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table provides one row for each service flow, and is
            indexed by ifIndex and wmanIfCmnCpsServiceFlowId. The
            ifIndex is provided by a ifEntry for BS or SS with ifType
            wmanMac. ifIndex is used to identify the SS or BS, while
            wmanIfCmnCpsServiceFlowId is ised to ifentify the service
            flow."
        INDEX     { ifIndex, wmanIfCmnCpsServiceFlowId }
        ::= { wmanIfCmnCpsServiceFlowTable 1 }

WmanIfCmnCpsServiceFlowEntry::= SEQUENCE {
        wmanIfCpsSfId                    Integer32,
        wmanIfCpsSfCid                   INTEGER,
        wmanIfCpsSfDirection             INTEGER,
        wmanIfCpsServiceClassIndex       INTEGER,
        wmanIfCpsSfState                 INTEGER,
        wmanIfCpsServiceClassName        DisplayString,
        wmanIfCpsTrafficPriority         INTEGER,
        wmanIfCpsMaxSustainedRate        INTEGER,
        wmanIfCpsMaxTrafficBurst         INTEGER,
        wmanIfCpsMinReservedRate         INTEGER,
        wmanIfCpsToleratedJitter         INTEGER,
        wmanIfCpsMaxLatency              INTEGER,
        wmanIfCpsScSchedulingType         WmanUlSchedulingType,
        wmanIfCpsScArqEnable             TruthValue,
        wmanIfCpsScArqWindowSize         INTEGER,
        wmanIfCpsScArqFragmentLifetime   INTEGER,
        wmanIfCpsScArqSyncLossTimeout    INTEGER,
        wmanIfCpsScArqDeliverInOrder     TruthValue,
        wmanIfCpsScArqRxPurgeTimeout     INTEGER,
        wmanIfCpsScFragmentLen           INTEGER,
```

```
        wmanIfCpsSCMinRsvdTolerableRate   INTEGER
        }

wmanIfCpsSfId OBJECT-TYPE
        SYNTAX      Integer32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "A 32 bit quantity that uniquely identifies a service flow
            to both the subscriber station and base station (BS)."
        ::= { wmanIfCmnCpsServiceFlowEntry 1 }

wmanIfCpsSfCid OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "A 16 bit channel identifier to identify the connection
            being created by DSA."
        ::= { wmanIfCmnCpsServiceFlowEntry 2 }

wmanIfCpsSfDirection OBJECT-TYPE
        SYNTAX      INTEGER {downstream(1),
                             upstream(2)}
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "An attribute indiating the service flow is downstream or
            upstream."
        ::= { wmanIfCmnCpsServiceFlowEntry 3 }

wmanIfCpsServiceClassIndex OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The index in wmanIfCpsServiceFlowTable describing the
             service flow and associated QoS parameters.
             If no associated entry in wmanIfCpsServiceFlowTable
             exists, this object returns a value of zero."
        ::= { wmanIfCmnCpsServiceFlowEntry 4 }

wmanICpsSfState OBJECT-TYPE
        SYNTAX      INTEGER {provisionedState(1),
                             admittedState(2),
                             activeState(3)}
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "wmanIfCpsSfState indicates the service flow state:
            Provisioned, AdmittedState(2), and Active service flow
            state."
        REFERENCE
            "Section 6.4.13.6, in IEEE 802.16REVd/D3-2004"
```

```
        ::= { wmanIfCmnCpsServiceFlowEntry 5 }

wmanIfCpsServiceClassName  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Refers to the Service Class Name"
        REFERENCE      "802.16 Chapter 11.4.8"
        ::= { wmanIfCmnCpsServiceFlowEntry 6 }

wmanIfCpsTrafficPriority OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of this parameter specifies the priority
            assigned to a service flow. For uplink service flows,
            the BS should use this parameter when determining
            precedence in request service and grant generation,
            and the SS shall preferentially select contention
            Request opportunities for Priority Request CIDs
            based on this priority"
        REFERENCE
            "Section 11.13.7 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfCmnCpsServiceFlowEntry 7 }

wmanIfCpsMaxSustainedRate OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This parameter defines the peak information rate
            of the service. The rate is expressed in bits per
            second and pertains to the SDUs at the input to
            the system."
        REFERENCE
            "Section 11.13.8 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfCmnCpsServiceFlowEntry 8 }

wmanIfCpsMaxTrafficBurst OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This parameter defines the maximum burst size that
            must be accommodated for the service."
        REFERENCE
            "Section 11.13.9 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfCmnCpsServiceFlowEntry 9 }

wmanIfCpsMinReservedRate OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
```

```
            STATUS      current
            DESCRIPTION
                "This parameter specifies the minimum rate reserved
                for this service flow."
            REFERENCE
                "Section 11.13.10 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfCmnCpsServiceFlowEntry 10 }

    wmanIfCpsToleratedJitter OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "This parameter defines the Maximum delay
                variation (jitter) for the connection."
            REFERENCE
                "Section 11.13.15 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfCmnCpsServiceFlowEntry 11 }

    wmanIfCpsMaxLatency OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The value of this parameter specifies the maximum
                latency between the reception of a packet by the BS
                or SS on its network interface and the forwarding
                of the packet to its RF Interface."
            REFERENCE
                "Section 11.13.16 in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfCmnCpsServiceFlowEntry 12 }

    wmanIfCpsScSchedulingType OBJECT-TYPE
            SYNTAX      WmanUlSchedulingType
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Specifies the upstream scheduling service used for
                upstream service flow. If the referenced parameter
                is not present in the corresponding 802.16 QOS
                Parameter Set of an upstream service flow, the
                default value of this object is bestEffort(2)."
            REFERENCE      "802.16 Chapter 11.4.8"
            ::= { wmanIfCmnCpsServiceFlowEntry 13 }

    wmanIfCpsScArqEnable OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "True(1) ARQ enabling is requested for the connection."
            ::= { wmanIfCmnCpsServiceFlowEntry 14 }

    wmanIfCpsScArqWindowSize      OBJECT-TYPE
```

```
            SYNTAX      INTEGER (1 .. 255)
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the maximum number of unacknowledged
                  fragments at any time."
            ::= { wmanIfCmnCpsServiceFlowEntry 15 }

    wmanIfCpsScArqFragmentLifetime OBJECT-TYPE
            SYNTAX      INTEGER (0 .. 65535)
            UNITS       "10 us"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The maximum time interval an ARQ fragment will be
                    managed by the transmitter ARQ machine, once
                    initial transmission of the fragment has occured.
                    If transmission or retransmission of the fragment
                    is not acknowledged by the receiver before the
                    time limit is reached, the fragmnet is discarded.
                    A value of 0 means Infinite."
            ::= { wmanIfCmnCpsServiceFlowEntry 16 }

    wmanIfCpsScArqSyncLossTimeout OBJECT-TYPE
            SYNTAX      INTEGER (0 .. 65535 )
            UNITS       "10 us"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "The maximum interval before declaring a loss
                  of synchronization of the sender and receiver
                  state machines. A value of 0 means Infinite."
            ::= { wmanIfCmnCpsServiceFlowEntry 17}

    wmanIfCpsScArqDeliverInOrder  OBJECT-TYPE
            SYNTAX      TruthValue
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates whether or not data is to be delivered
                  by the receiving MAC to its client application
                  in the order in which data was handed off to the
                  originating MAC."
            ::= { wmanIfCmnCpsServiceFlowEntry 18 }

    wmanIfCpsScArqRxPurgeTimeout  OBJECT-TYPE
            SYNTAX      INTEGER (0 .. 65535)
            UNITS       "10 us"
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION
                "Indicates the time interval the ARQ window is advanced
                  after  a  fragment  is  received.  A  value  of  0  means
    Infinite."
```

```
        ::= { wmanIfCmnCpsServiceFlowEntry 19}

wmanIfCpsScFragmentLen OBJECT-TYPE
        SYNTAX      INTEGER (32 .. 2040)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The maximum size fragment a transmitter shall form
            or a receiver shall expect to receive."
        ::= { wmanIfCmnCpsServiceFlowEntry 20 }

wmanIfCpsSCMinRsvdTolerableRate OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Minimum Tolerable Traffic Rate = R (bits/sec) with
            time base T(sec) means the following. Let S denote
            additional demand accumulated at the MAC SAP of the
            transmitter during an arbitrary time interval of the
            length T. Then the amount of data forwarded at the
            receiver to CS (in bits) during this interval should
            be not less than min {S, R * T}."
        REFERENCE   "Section 11.13.11 in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfCmnCpsServiceFlowEntry 17 }

wmanIfCmnClassifierRuleTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF WmanIfCmnClassifierRuleEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table contains packet classifier rules associated
            with service flows."
        ::= { wmanIfCmnPacketCs 3 }

wmanIfCmnClassifierRuleEntry OBJECT-TYPE
        SYNTAX      WmanIfCmnClassifierRuleEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "This table provides one row for each packet classifier
            rule, and is indexed by ifIndex, wmanIfCmnSfId and
            wmanIfCmnClassifierRuleIndex. ifIndex identifies the SS.
            wmanIfCmnSfId identifies the service flow, and
            wmanIfCmnClassifierRuleIndexAn identifies the packet
            classifier rule."
        INDEX { ifIndex, wmanIfCpsSfId, wmanIfCmnClassifierRuleIndex }
        ::= { wmanIfCmnClassifierRuleTable 1 }

WmanIfCmnClassifierRuleEntry::= SEQUENCE {
        wmanIfCmnClassifierRuleIndex            Unsigned32,
        wmanIfCmnClassifierRuleServiceFlowId    Unsigned32,
        wmanIfCmnClassifierRulePriority         INTEGER,
        wmanIfCmnClassifierRuleIpTosLow         OCTET STRING,
```

```
        wmanIfCmnClassifierRuleIpTosHigh         OCTET STRING,
        wmanIfCmnClassifierRuleIpTosMask         OCTET STRING,
        wmanIfCmnClassifierRuleIpProtocol        Integer32,
        wmanIfCmnClassifierRuleInetAddressType   InetAddressType,
        wmanIfCmnClassifierRuleInetSourceAddr    InetAddress,
        wmanIfCmnClassifierRuleInetSourceMask    InetAddress,
        wmanIfCmnClassifierRuleInetDestAddr      InetAddress,
        wmanIfCmnClassifierRuleInetDestMask      InetAddress,
        wmanIfCmnClassifierRuleSourcePortStart   Integer32,
        wmanIfCmnClassifierRuleSourcePortEnd     Integer32,
        wmanIfCmnClassifierRuleDestPortStart     Integer32,
        wmanIfCmnClassifierRuleDestPortEnd       Integer32,
        wmanIfCmnClassifierRuleDestMacAddr       MacAddress,
        wmanIfCmnClassifierRuleDestMacMask       MacAddress,
        wmanIfCmnClassifierRuleSourceMacAddr     MacAddress,
        wmanIfCmnClassifierRuleSourceMacMask     MacAddress,
        wmanIfCmnClassifierRuleEnetProtocolType  INTEGER,
        wmanIfCmnClassifierRuleEnetProtocol      Integer32,
        wmanIfCmnClassifierRuleUserPriLow        Integer32,
        wmanIfCmnClassifierRuleUserPriHigh       Integer32,
        wmanIfCmnClassifierRuleVlanId            Integer32,
        wmanIfCmnClassifierRuleState             INTEGER,
        wmanIfCmnClassifierRulePkts              Counter64,
        wmanIfCmnClassifierRuleRowStatus         RowStatus
        }

wmanIfCmnClassifierRuleIndex  OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "An index is assigned to each classifier in the classifiers
           table"
        REFERENCE      ""
        ::= { wmanIfCmnClassifierRuleEntry 1 }

wmanIfCmnClassifierRuleServiceFlowId OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An index assigned to a service flow by SC (SFID)."
        REFERENCE      "802.16 Chapter 11.4.8"
        ::= { wmanIfCmnClassifierRuleEntry 2 }

wmanIfCmnClassifierRulePriority OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The value specifies the order of evaluation of the
            classifiers. The higher the value the higher the
            priority. The value of 0 is used as default in
            provisioned service flows classifiers. The default
```

```
                value of 64 is used for dynamic service flow classifiers.
                If the referenced parameter is not present in a classifier,
                this object reports the default value as defined above"
        ::= { wmanIfCmnClassifierRuleEntry 3 }

    wmanIfCmnClassifierRuleIpTosLow OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The low value of a range of TOS byte values. If the
                referenced parameter is not present in a classifier, this
                object reports the value of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 4 }

    wmanIfCmnClassifierRuleIpTosHigh OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The 8-bit high value of a range of TOS byte values.
                If the referenced parameter is not present in a classifier,
                this object reports the value of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 5 }

    wmanIfCmnClassifierRuleIpTosMask OBJECT-TYPE
            SYNTAX      OCTET STRING (SIZE(1))
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The mask value is bitwise ANDed with TOS byte in an IP
                packet and this value is used check range checking of
                TosLow and TosHigh. If the referenced parameter is not
                present in a classifier, this object reports the value
                of 0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 6 }

    wmanIfCmnClassifierRuleIpProtocol OBJECT-TYPE
            SYNTAX      Integer32 (0..255)
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object indicates the value of the IP Protocol field
                required for IP packets to match this rule. If the
                referenced parameter is not present in a classifier, this
                object reports the value of 0."
            REFERENCE      "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 7 }

    wmanIfCmnClassifierRuleInetAddressType OBJECT-TYPE
            SYNTAX      InetAddressType
```

```
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The type of the internet address for
                wmanIfCmnClassifierRuleInetSourceAddr,
                wmanIfCmnClassifierRuleInetSourceMask,
                wmanIfCmnClassifierRuleInetDestAddr, and
                wmanIfCmnClassifierRuleInetDestMask.
                If the referenced parameter is not present in a classifier,
                this object reports the value of ipv4(1)."
            REFERENCE       ""
            ::= { wmanIfCmnClassifierRuleEntry 8 }

    wmanIfCmnClassifierRuleInetSourceAddr OBJECT-TYPE
            SYNTAX      InetAddress
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object specifies the value of the IP Source Address
                required for packets to match this rule. An IP packet
                matches the rule when the packet ip source address bitwise
                ANDed with the wmanIfCmnClassifierRuleInetSourceMask value
                equals the wmanIfCmnClassifierRuleInetSourceAddr value.
                If the referenced parameter is not present n a classifier,
                this object reports the value of 0.0.0.0."
            REFERENCE       "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 9 }

    wmanIfCmnClassifierRuleInetSourceMask OBJECT-TYPE
            SYNTAX      InetAddress
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object specifies which bits of a packet's IP Source
                Address that are compared to match this rule. An IP packet
                matches the rule when the packet source address bitwise
                ANDed with the
                wmanIfCmnClassifierRuleInetSourceMask value equals the
                wmanIfCmnClassifierRuleInetSourceAddr value.
                If the referenced parameter is not present in a classifier,
                this object reports the value of 0.0.0.0."
            REFERENCE      "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 10 }

    wmanIfCmnClassifierRuleInetDestAddr OBJECT-TYPE
            SYNTAX      InetAddress
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object specifies the value of the IP Destination
                Address required for packets to match this rule. An IP
                packet matches the rule when the packet IP destination
                address bitwise ANDed with the
                wmanIfCmnClassifierRuleInetDestMask value equals the
```

```
            wmanIfCmnClassifierRuleInetDestAddr value.
            If the referenced parameter is not present in a
            classifier, this object reports the value of 0.0.0.0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 11 }

wmanIfCmnClassifierRuleInetDestMask OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies which bits of a packet's IP
            Destination Address that are compared to match this rule.
            An IP packet matches the rule when the packet destination
            address bitwise ANDed with the
            wmanIfCmnClassifierRuleInetDestMask value equals the
            wmanIfCmnClassifierRuleInetDestAddr value.
            If the referenced parameter is not present in a classifier
            , this object reports the value of 0.0.0.0."
        REFERENCE       "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 12}

wmanIfCmnClassifierRuleSourcePortStart OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
             "This object specifies the low end inclusive range of
             TCP/UDP source port numbers to which a packet is compared
             . This object is irrelevant for non-TCP/UDP IP packets.
             If the referenced parameter is not present in a
             classifier, this object reports the value of 0."
        REFERENCE        "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 13 }

wmanIfCmnClassifierRuleSourcePortEnd OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the high end inclusive range of
            TCP/UDP source port numbers to which a packet is compared.
            This object is irrelevant for non-TCP/UDP IP packets.
            If the referenced parameter is not present in a classifier,
            this object reports the value of 65535."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 14 }

wmanIfCmnClassifierRuleDestPortStart OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the low end inclusive range of
```

```
            TCP/UDP destination port numbers to which a packet is
            compared. If the referenced parameter is not present
            in a classifier, this object reports the value of 0."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 15 }

wmanIfCmnClassifierRuleDestPortEnd OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object specifies the high end inclusive range of
            TCP/UDP destination port numbers to which a packet is
            compared. If the referenced parameter is not present
            in a classifier, this object reports the value of
            65535."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 16 }

wmanIfCmnClassifierRuleDestMacAddr OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches an entry when its destination
            MAC address bitwise ANDed with
            wmanIfCmnClassifierRuleDestMacMask equals the value of
            wmanIfCmnClassifierRuleDestMacAddr. If the referenced
            parameter is not present in a classifier, this object
            reports the value of '000000000000'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 17 }

wmanIfCmnClassifierRuleDestMacMask OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches an entry when its destination
            MAC address bitwise ANDed with
            wmanIfCmnClassifierRuleDestMacMask equals the value of
            wmanIfCmnClassifierRuleDestMacAddr. If the referenced
            parameter is not present in a classifier, this object
            reports the value of '000000000000'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 18 }

wmanIfCmnClassifierRuleSourceMacAddr OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches this entry when its source
            MAC address  bitwise ANDed with
```

```
            wmanIfCmnClassifierRuleSourceMacMask equals the value
            of wmanIfCmnClassifierRuleSourceMacAddr. If the
            referenced parameter is not present in a classifier,
            this object reports the value of 'FFFFFFFFFFFF'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 19 }

wmanIfCmnClassifierRuleSourceMacMask OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "An Ethernet packet matches an entry when its destination
            MAC address bitwise ANDed with
            wmanIfCmnClassifierRuleSourceMacMask equals the value of
            wmanIfCmnClassifierRuleSourceMacAddr. If the referenced
            parameter is not present in a classifier, this object
            reports the value of '000000000000'H."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 20 }

wmanIfCmnClassifierRuleEnetProtocolType OBJECT-TYPE
        SYNTAX      INTEGER {none(0),
                             ethertype(1),
                             dsap(2)}
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object indicates the format of the layer 3 protocol
            id in the Ethernet packet. A value of none(0) means that
            the rule does not use the layer 3 protocol type as a
            matching criteria. A value of ethertype(1) means that the
            rule applies only to frames which contains an EtherType
            value. Ethertype values are contained in packets using
            the Dec-Intel-Xerox (DIX) encapsulation or the RFC1042
            Sub-Network Access Protocol (SNAP) encapsulation formats.
            A value of dsap(2) means that the rule applies only to
            frames using the IEEE802.3 encapsulation format with a
            Destination Service Access Point (DSAP) other than 0xAA
            (which is reserved for SNAP). If the Ethernet frame
            contains an 802.1P/Q Tag header (i.e. EtherType 0x8100),
            this object applies to the embedded EtherType field within
            the 802.1P/Q header. If the referenced parameter is not
            present in a classifier, this object reports the value of
            0."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 21 }

wmanIfCmnClassifierRuleEnetProtocol OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "If wmanIfCmnClassifierRuleEnetProtocolType is none(0),
```

```
                this object is ignored when considering whether a packet
                matches the current rule.
                If wmanIfCmnClassifierRuleEnetProtocolType is ethertype(1),
                this object gives the 16-bit value of the EtherType that
                the packet must match in order to match the rule.
                If wmanIfCmnClassifierRuleEnetProtocolType is dsap(2), the
                lower 8 bits of this object's value must match the DSAP
                byte of the packet in order to match the rule.
                If wmanIfCmnClassifierRuleEnetProtocolType is mac(3), the
                lower 8 bits of this object value represent a lower bound
                (inclusive) of MAC management message type codes matched,
                and the upper 8 bits of this object value represent the
                upper bound (inclusive) of matched MAC message type codes.
                Certain message type codes are excluded from matching, as
                specified in the reference.
                If the Ethernet frame contains an 802.1P/Q Tag header
                (i.e. EtherType 0x8100), this object applies to the
                embedded EtherType field within the 802.1P/Q header.
                If the referenced parameter is not present in the
                classifier, the value of this object is reported as 0."
            REFERENCE   "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 22 }

    wmanIfCmnClassifierRuleUserPriLow OBJECT-TYPE
            SYNTAX      Integer32 (0..7)
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object applies only to Ethernet frames using the
                802.1P/Q tag header (indicated with EtherType 0x8100).
                Such frames include a 16-bit Tag that contains a 3 bit
                Priority field and a 12 bit VLAN number.
                Tagged Ethernet packets must have a 3-bit Priority field
                within the range of wmanIfCmnClassifierRulePriLow and
                wmanIfCmnClassifierRulePriHigh in order to match this
                rule.
                If the referenced parameter is not present in the
                classifier, the value of this object is reported as 0."
            REFERENCE   "802.16 Chapter 11.4.9"
            ::= { wmanIfCmnClassifierRuleEntry 23 }

    wmanIfCmnClassifierRuleUserPriHigh OBJECT-TYPE
            SYNTAX      Integer32 (0..7)
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "This object applies only to Ethernet frames using the
                802.1P/Q tag header (indicated with EtherType 0x8100).
                Such frames include a 16-bit Tag that contains a 3 bit
                Priority field and a 12 bit VLAN number.
                Tagged Ethernet packets must have a 3-bit Priority
                field within the range of wmanIfCmnClassifierRulePriLow
                and wmanIfCmnClassifierRulePriHigh in order to match
                this rule.
```

```
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 7."
        REFERENCE   "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 24 }

    wmanIfCmnClassifierRuleVlanId OBJECT-TYPE
        SYNTAX      Integer32 (0..4095)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object applies only to Ethernet frames using the
            802.1P/Q tag header.
            If this object's value is nonzero, tagged packets must
            have a VLAN Identifier that matches the value in order
            to match the rule.
            Only the least significant 12 bits of this object's
            value are valid.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported as 0."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 25 }

    wmanIfCmnClassifierRuleState OBJECT-TYPE
        SYNTAX      INTEGER {active(1),
                             inactive(2)}
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object indicates whether or not the classifier is
            enabled to classify packets to a Service Flow.
            If the referenced parameter is not present in the
            classifier, the value of this object is reported
            as active(1)."
        REFERENCE      "802.16 Chapter 11.4.9"
        ::= { wmanIfCmnClassifierRuleEntry 26 }

    wmanIfCmnClassifierRulePkts OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "This object counts the number of packets that have
            been classified using this entry."
        ::= { wmanIfCmnClassifierRuleEntry 27 }

    wmanIfCmnClassifierRuleRowStatus OBJECT-TYPE
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This object is used to create a new row or modify or
            delete an existing row in this table.

            If the implementator of this MIB has choosen not
```

```
                to implement 'dynamic assignment' of profiles, this
                object is not useful and should return noSuchName
                upon SNMP request."
            ::= { wmanIfCmnClassifierRuleEntry 28 }

    -- Common PKM  group
    -- wmanIfCmnPkmObjects contain the Privacy Sublayer objects that are
    -- common to both Base Station and Subscriber Station
    wmanIfCmnPkmObjects OBJECT IDENTIFIER ::= { wmanIfCommonObjects  3 }

      --
      -- Table wmanIfCmnCryptoSuiteTable
      --

      wmanIfCmnCryptoSuiteTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF   WmanIfCmnCryptoSuiteEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "This table describes the PKM cryptographic suite
                 capabilites for each SS or BS wireless interface."
      ::= { wmanIfCmnPkmObjects  4 }


      wmanIfCmnCryptoSuiteEntry OBJECT-TYPE
            SYNTAX      WmanIfCmnCryptoSuiteEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "Each entry contains the cryptographic suite pair that SS
                 or BS supports."
            INDEX       { ifIndex, wmanIfSsCryptoSuiteIndex }
      ::= { wmanIfCmnCryptoSuiteTable  1 }


       WmanIfCmnCryptoSuiteEntry ::= SEQUENCE {
            wmanIfSsCryptoSuiteIndex                        Integer32,
            wmanIfCmnCryptoSuiteDataEncryptAlg              INTEGER,
            wmanIfCmnCryptoSuiteDataAuthentAlg              INTEGER,
            wmanIfCmnCryptoSuiteTEKEncryptAlg               INTEGER
            }


      wmanIfSsCryptoSuiteIndex OBJECT-TYPE
            SYNTAX      Integer32
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                 "The index for a cryptographic suite row."
            ::= { wmanIfCmnCryptoSuiteEntry 1 }


      wmanIfCmnCryptoSuiteDataEncryptAlg OBJECT-TYPE
            SYNTAX      INTEGER { none(0),
```

```
                                des56CbcMode(1) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the data encryption algorithm
             for this cryptographic suite capability."
        REFERENCE
             "IEEE 802.16 standard; Table 301"
        ::= { wmanIfCmnCryptoSuiteEntry 2 }


    wmanIfCmnCryptoSuiteDataAuthentAlg OBJECT-TYPE
        SYNTAX      INTEGER { none(0) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the data authentication
             algorithm for this cryptographic suite capability."
        REFERENCE
             "IEEE 802.16 standard; Table 302"
        ::= { wmanIfCmnCryptoSuiteEntry 3 }


    wmanIfCmnCryptoSuiteTEKEncryptAlg OBJECT-TYPE
        SYNTAX      INTEGER { tripleDES(0),
                              rsa1024(1) }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
             "The value of this object is the TEK key encryption
             algorithm for this cryptographic suite capability."
        REFERENCE
             "IEEE 802.16 standard; Table 303"
        ::= { wmanIfCmnCryptoSuiteEntry 4 }

--
-- wmanIfCmnOfdmPhy contain the OFDM PHY objects that are common to both
-- Base Station and Subscriber Station. When the objects are implemented
-- in the BS, they should have the read-write access. When the objects
-- are implemented the SS, they should have the read-only access.
--
wmanIfCmnOfdmPhy OBJECT IDENTIFIER ::= { wmanIfCommonObjects 4 }

wmanIfOfdmUplinkChannelTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF WmanIfOfdmUplinkChannelEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table contains UCD channel attributes, defining the
            transmission characteristics of uplink channels"
        REFERENCE
            "Section 11.3.1, table 276 and 279, in IEEE
            802.16REVd/D3-2004"
        ::= { wmanIfCmnOfdmPhy 1 }
```

```
1
2   wmanIfOfdmUplinkChannelEntry OBJECT-TYPE
3           SYNTAX      WmanIfOfdmUplinkChannelEntry
4           MAX-ACCESS  not-accessible
5           STATUS      current
6           DESCRIPTION
7                 "This table provides one row for each uplink 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 wmanUpstream.
11                The objects in each entry will be implemented as
12                read-create in BS and read-only in SS."
13          INDEX { ifIndex }
14          ::= { wmanIfOfdmUplinkChannelTable 1 }
15
16  WmanIfOfdmUplinkChannelEntry ::= SEQUENCE {
17          wmanIfOfdmCtBasedResvTimeout     INTEGER,
18          wmanIfOfdmBwReqOppSize           INTEGER,
19          wmanIfOfdmRangReqOppSize         INTEGER,
20          wmanIfOfdmUplinkCenterFreq       INTEGER,
21          wmanIfOfdmUlAntennaNmr           INTEGER,
22          wmanIfOfdmSubChReqRegionFull     INTEGER,
23          wmanIfOfdmSubChFocusCtCode       INTEGER,
24          wmanIfOfdmChannelWidth           INTEGER,
25          wmanIfOfdmUplinkChannelRowStatus RowStatus
26          }
27
28  wmanIfOfdmCtBasedResvTimeout OBJECT-TYPE
29          SYNTAX      INTEGER (1..255)
30          MAX-ACCESS  read-write
31          STATUS      current
32          DESCRIPTION
33              "The number of UL-MAPs to receive before contention-based
34              reservation is attempted again for the same connection."
35          REFERENCE
36              "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
37          ::= { wmanIfOfdmUplinkChannelEntry 1 }
38
39  wmanIfOfdmBwReqOppSize OBJECT-TYPE
40          SYNTAX      INTEGER (1..65535)
41          MAX-ACCESS  read-write
42          STATUS      current
43          DESCRIPTION
44              " Size (in units of PS) of PHY payload that SS may use to
45              format and transmit a bandwidth request message in a
46              contention request opportunity. The value includes all
47              PHY overhead as well as allowance for the MAC data the
48              message may hold."
49          REFERENCE
50              "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
51          ::= { wmanIfOfdmUplinkChannelEntry 2 }
52
53  wmanIfOfdmRangReqOppSize OBJECT-TYPE
54          SYNTAX      INTEGER (1..65535)
```

```
        UNITS       "PS"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Size (in units of PS) of PHY payload that SS may use to
            format and transmit a RNG-REQ message in a contention
            request opportunity. The value includes all PHY overhead
            as well as allowance for the MAC data the message may
            hold and the maxiumum SS/BS roundtrip propagation delay."
        REFERENCE
            "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmUplinkChannelEntry 3 }

wmanIfOfdmUplinkCenterFreq OBJECT-TYPE
        SYNTAX      INTEGER
        UNITS       "KHz"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Uplink center frequency (KHz)"
        REFERENCE
            "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmUplinkChannelEntry 4 }

wmanIfOfdmUlAntennaNmr OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Antenna number or pointing reference"
        REFERENCE
            "Section 11.3.1, table 276, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmUplinkChannelEntry 5 }

wmanIfOfdmSubChReqRegionFull OBJECT-TYPE
        SYNTAX      INTEGER {oneSubchannel(0),
                             twoSubchannels(1),
                             fourSubchannels(2),
                             eightSubchannels(3),
                             sixteenSubchannels(4)}
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "Bits 0…2 Number of subchannels used by each transmit
            opportunity when REQ Region-Full is allocated in
            subchannelization region, per the following enumeration:
               0: 1 Subchannel.
               1: 2 Subchannels.
               2: 4 Subchannels.
               3: 8 Subchannels.
               4: 16 Subchannels.
               5-7: Shall not be used.
             Bits 3…7: Number of OFDM symbols used by each transmit
             opportunity when REQ Region-Full is allocated in
```

```
                subchannelization region.
            REFERENCE
                Section 11.3.1, table 279, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmUplinkChannelEntry 6 }

    wmanIfOfdmSubChFocusCtCode OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Number of contention codes (CSE) that shall only be used to
                request a subchannelized allocation. Default value 0.
    Allowed
                values 0-48."
            REFERENCE
                "Section 11.3.1, table 279, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmUplinkChannelEntry 7 }

    wmanIfOfdmChannelWidth OBJECT-TYPE
            SYNTAX      INTEGER
            UNITS       "10KHz"
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Channel width, increments of 10 KHz Shall not be used
                in license-exempt bands."
            ::= { wmanIfOfdmUplinkChannelEntry 8 }

    wmanIfOfdmUplinkChannelRowStatus OBJECT-TYPE
            SYNTAX          RowStatus
            MAX-ACCESS      read-create
            STATUS          current
            DESCRIPTION
                "This object is used to create a new row or modify or
                delete an existing row in this table.

                If the implementator of this MIB has choosen not
                to implement 'dynamic assignment' of profiles, this
                object is not useful and should return noSuchName
                upon SNMP request."
            ::= { wmanIfOfdmUplinkChannelEntry 9 }

    wmanIfOfdmDownlinkChannelTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF WmanIfOfdmDownlinkChannelEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "This table contains DCD channel attributes, defining the
                transmission characteristics of downlink channels"
            REFERENCE
                "Section 11.4.1, Table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfCmnOfdmPhy 2 }

    wmanIfOfdmDownlinkChannelEntry OBJECT-TYPE
```

```
        SYNTAX      WmanIfOfdmDownlinkChannelEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
              "This table provides one row for each downlink channel of
              multi-sector BS, and is indexed by BS ifIndex. An entry
              in this table exists for each ifEntry of BS with an
              ifType of wmanDownstream.
              The objects in each entry will be implemented as
              read-create in BS and read-only in SS."
        INDEX { ifIndex }
        ::= { wmanIfOfdmDownlinkChannelTable 1 }

WmanIfOfdmDownlinkChannelEntry ::= SEQUENCE {
        wmanIfOfdmBsEIRP                    INTEGER,
        wmanIfOfdmChannelNumber             INTEGER,
        wmanIfOfdmTTG                       INTEGER,
        wmanIfOfdmRTG                       INTEGER,
        wmanIfOfdmInitRngMaxRSS             INTEGER,
        wmanIfOfdmChSwitchFrameNmr          INTEGER,
        wmanIfOfdmDlAntennaNmr              INTEGER,
        wmanIfOfdmDownlinkCenterFreq        INTEGER,
        wmanIfOfdmMacVersion                INTEGER,
        wmanIfOfdmDownlinkChannelRowStatus RowStatus
        }

wmanIfOfdmBsEIRP OBJECT-TYPE
        SYNTAX      INTEGER
        UNITS       "dbM"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Signed in units of 1 dBM."
        REFERENCE
            "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmDownlinkChannelEntry 1 }

wmanIfOfdmChannelNumber OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Downlink channel number as defined in 8.5.
              Used for license-exempt operation only."
        REFERENCE
            "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmDownlinkChannelEntry 2 }

wmanIfOfdmTTG OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            " Transmit / Receive Transition Gap."
```

```
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 3 }

    wmanIfOfdmRTG OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " Receive / Transmit Transition Gap."
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 4 }

    wmanIfOfdmInitRngMaxRSS OBJECT-TYPE
            SYNTAX      INTEGER
            UNITS       "dbM"
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " Initial Ranging Max. Received Signal Strength at BS
                Signed in units of 1 dBm."
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 5 }

    wmanIfOfdmChSwitchFrameNmr OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " Channel switch frame number as defined in 6.4.14.7,
                Used for license-exempt operation only."
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 6 }

    wmanIfOfdmDlAntennaNmr OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " Antenna number or pointing reference."
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 7 }

    wmanIfOfdmDownlinkCenterFreq OBJECT-TYPE
            SYNTAX      INTEGER
            UNITS       "KHz"
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " Downlink center frequency (kHz)."
```

```
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 8 }

    wmanIfOfdmMacVersion OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " This parameter specifies the version of 802.16 to which
                the message originator conforms."
            REFERENCE
                "Section 11.4.1, table 286, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDownlinkChannelEntry 9 }

    wmanIfOfdmDownlinkChannelRowStatus OBJECT-TYPE
            SYNTAX      RowStatus
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This object is used to create a new row or modify or
                delete an existing row in this table.

                If the implementator of this MIB has choosen not
                to implement 'dynamic assignment' of profiles, this
                object is not useful and should return noSuchName
                upon SNMP request."
            ::= { wmanIfOfdmDownlinkChannelEntry 10 }

    wmanIfOfdmUcdBurstProfileTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF WmanIfOfdmUcdBurstProfileEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "This table contains UCD burst profiles for each uplink
                channel"
            REFERENCE
                "Section 11.3.1.1, table 281 and 284, in IEEE
                802.16REVd/D3-2004"
            ::= { wmanIfCmnOfdmPhy 3 }

    wmanIfOfdmUcdBurstProfileEntry OBJECT-TYPE
            SYNTAX      WmanIfOfdmUcdBurstProfileEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                  "This table provides one row for each UCD burst profile.
                  This table is double indexed. The primary index is an
                  ifIndex with an ifType of wmanUpstream. The secondary
    index
                  is wmanIfOfdmOfdmUcdBurstProfIndex.
                  The objects in each entry will be implemented as
                  read-create in BS and read-only in SS."
            INDEX { ifIndex, wmanIfOfdmOfdmUcdBurstProfIndex }
```

```
        ::= { wmanIfOfdmUcdBurstProfileTable 1 }

WmanIfOfdmUcdBurstProfileEntry ::= SEQUENCE {
        wmanIfOfdmOfdmUcdBurstProfIndex    INTEGER,
        wmanIfOfdmUplinkFrequency          INTEGER,
        wmanIfOfdmUcdAntennaNmr            INTEGER,
        wmanIfOfdmUcdFecCodeType           INTEGER,
        wmanIfOfdmFocusCtPowerBoost        INTEGER,
        wmanIfOfdmUcdBurstProfileRowStatus RowStatus
        }

wmanIfOfdmOfdmUcdBurstProfIndex OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "ifIndex and wmanIfOfdmOfdmUcdBurstProfIndex uniquely
            identify an entry in the wmanIfOfdmUcdBurstProfileTable."
        ::= { wmanIfOfdmUcdBurstProfileEntry 1 }

wmanIfOfdmUplinkFrequency OBJECT-TYPE
        SYNTAX      INTEGER
        UNITS       "KHz"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "Uplink Frequency (kHz)."
        REFERENCE
            "Section 11.3.1.1, table 281, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmUcdBurstProfileEntry 2 }

wmanIfOfdmUcdAntennaNmr OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "Antenna number or pointing reference."
        REFERENCE
            "Section 11.3.1.1, table 281, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmUcdBurstProfileEntry 3 }

wmanIfOfdmUcdFecCodeType OBJECT-TYPE
        SYNTAX      INTEGER {qpskRsCcCc1-2(0),
                             qpskRsCcCc3-4(1),
                             sixteenQamRsCcCc1-2(2),
                             sixteenQamRsCcCc3-4(3),
                             sixtyFourQamRsCcCc2-3(4),
                             sixtyFourQamRsCcCc3-4(5),
                             qpskBtc1-2(6),
                             qpskBtc3-4(7),
                             sixteenQamBtc3-5(8),
                             sixteenQamBtc4-5(9),
                             sixtyFourQamBtc2-3(10),
                             sixtyFourQamBtc5-6(11),
```

```
                                qpskCtc1-2(12),
                                qpskCtc2-3(13),
                                qpskCtc3-4(14),
                                sixteenQamCtc3-4(16),
                                sixteenQamCtc2-3(17),
                                sixtyFourQamCtc3-4(18)}
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " 0= QPSK (RS+CC/CC) 1/2
                    1= QPSK (RS+CC/Cc) 3/4
                    2= 16-QAM (RS+CC/CC) 1/2
                    3= 16-QAM (RS+CC/CC) 3/4
                    4= 64-QAM (RS+CC/CC) 2/3
                    5= 64-QAM (RS+CC/CC) 3/4
                    6= QPSK (BTC) 1/2
                    7= QPSK (BTC) 3/4
                    8= 16-QAM (BTC) 3/5
                    9= 16-QAM (BTC) 4/5
                    10 = 64-QAM (BTC) 2/3
                    11 = 64-QAM (BTC) 5/6
                    12 = QPSK (CTC) ½
                    13 = QPSK (CTC) 2/3
                    14 = QPSK (CTC) ¾
                    15 = 16-QAM (CTC) ½
                    16 = 16-QAM (CTC) ¾
                    17 = 64-QAM (CTC) 2/3
                    18 = 64-QAM (CTC) ¾
                    19-255 Reserved."
            REFERENCE
                "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmUcdBurstProfileEntry 4 }

    wmanIfOfdmFocusCtPowerBoost OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "The power boost in dB of focused contention carriers, as
               described in 8.3.6.3.3."
            REFERENCE
                "Section 11.3.1.1, table 284, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmUcdBurstProfileEntry 5 }

    wmanIfOfdmUcdBurstProfileRowStatus OBJECT-TYPE
            SYNTAX      RowStatus
            MAX-ACCESS  read-create
            STATUS      current
            DESCRIPTION
                "This object is used to create a new row or modify or
                delete an existing row in this table.

                If the implementator of this MIB has choosen not
                to implement 'dynamic assignment' of profiles, this
```

```
                object is not useful and should return noSuchName
                upon SNMP request."
            ::= { wmanIfOfdmUcdBurstProfileEntry 6 }

    wmanIfOfdmDcdBurstProfileTable OBJECT-TYPE
            SYNTAX      SEQUENCE OF WmanIfOfdmDcdBurstProfileEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                  "This table provides one row for each DCD burst profile.
                  This table is double indexed. The primary index is an
                  ifIndex with an ifType of wmanDownstream. The secondary
    index
                  is wmanIfOfdmOfdmDcdBurstProfIndex"
            ::= { wmanIfCmnOfdmPhy 4 }


    wmanIfOfdmDcdBurstProfileEntry OBJECT-TYPE
            SYNTAX      WmanIfOfdmDcdBurstProfileEntry
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                  "This table provides one row for each DCD burst profile.
                  This table is double indexed. The primary index is an
                  ifIndex with an ifType of wmanDownstream. The secondary
    index
                  is wmanIfOfdmOfdmDcdBurstProfIndex.
                  The objects in each entry will be implemented as
                  read-create in BS and read-only in SS."
            INDEX { ifIndex, wmanIfOfdmOfdmDcdBurstProfIndex }
            ::= { wmanIfOfdmDcdBurstProfileTable 1 }

    WmanIfOfdmDcdBurstProfileEntry ::= SEQUENCE {
            wmanIfOfdmOfdmDcdBurstProfIndex    INTEGER,
            wmanIfOfdmDownlinkFrequency        INTEGER,
            wmanIfOfdmDcdAntennaNmr            INTEGER,
            wmanIfOfdmDcdFecCodeType           INTEGER,
            wmanIfOfdmDiucMandatoryExitThresh  INTEGER,
            wmanIfOfdmDiucMinEntryThresh       INTEGER,
            wmanIfOfdmDcdBurstProfileRowStatus RowStatus
            }

    wmanIfOfdmOfdmDcdBurstProfIndex OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  not-accessible
            STATUS      current
            DESCRIPTION
                "ifIndex and wmanIfOfdmOfdmDcdBurstProfIndex uniquely
                identify an entry in the wmanIfOfdmOfdmDcdBurstProfIndex."
            ::= { wmanIfOfdmDcdBurstProfileEntry 1 }

    wmanIfOfdmDownlinkFrequency OBJECT-TYPE
            SYNTAX      INTEGER
            UNITS       "KHz"
```

```
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Downlink Frequency (kHz)."
            REFERENCE
                "Section 11.4.1, table 287, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDcdBurstProfileEntry 2 }

    wmanIfOfdmDcdAntennaNmr OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                "Antenna number or pointing reference."
            REFERENCE
                "Section 11.4.1, table 287, in IEEE 802.16REVd/D3-2004"
            ::= { wmanIfOfdmDcdBurstProfileEntry 3 }

    wmanIfOfdmDcdFecCodeType OBJECT-TYPE
            SYNTAX      INTEGER {qpskRsCc1-2(0),
                                 qpskRsCc3-4(1),
                                 sixtyFourQamRsCc1-2(2),
                                 sixtyFourQamRsCc3-4(3),
                                 sixtyFourQamRsCc2-3(4),
                                 sixtyFourQamRsCc3-4(5),
                                 qpskBtc1-2(6),
                                 qpskBtc3-4(7),
                                 sixteenQamBtc3-4(8),
                                 sixteenQamBtc4-5(9),
                                 sixtyFourQamBtc2-3or5-8(10),
                                 sixtyFourQamBtc5-6or4-5(11),
                                 qpskCtc1-2(12),
                                 qpskCtc2-3(13),
                                 qpskCtc3-4(14),
                                 sixteenQamCtc1-2(16),
                                 sixteenQamCtc3-4(17),
                                 sixtyFourQamCtc3-4(18)}
            MAX-ACCESS  read-write
            STATUS      current
            DESCRIPTION
                " 0= QPSK (RS+CC) 1/2
                  1= QPSK (RS+CC) 3/4
                  2= 16-QAM (RS+CC) 1/2
                  3= 16-QAM (RS+CC) 3/4
                  4= 64-QAM (RS+CC) 2/3
                  5= 64-QAM (RS+CC) 3/4
                  6= QPSK (BTC) 1/2
                  7= QPSK (BTC) 3/4
                  8= 16-QAM (BTC) 3/5
                  9= 16-QAM (BTC) 4/5
                  10 = 64-QAM (BTC) 2/3 or 5/8
                  11 = 64-QAM (BTC) 5/6 or 4/5
                  12 = QPSK (CTC) ½
                  13 = QPSK (CTC) 2/3
```

```
                14 = QPSK (CTC) ¾
                15 = 16-QAM (CTC) ½
                16 = 16-QAM (CTC) ¾
                17 = 64-QAM (CTC) 2/3
                18 = 64-QAM (CTC) ¾
                19-255 Reserved."
        REFERENCE
            "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmDcdBurstProfileEntry 4 }

wmanIfOfdmDiucMandatoryExitThresh OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "DIUC mandatory exit threshold: 0-63.75 dB CINR at or below
            where this DIUC can no longer be used and where this change
            to a more robust DIUC is required, in 0.25 dB units."
        REFERENCE
            "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmDcdBurstProfileEntry 5 }

wmanIfOfdmDiucMinEntryThresh OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "DIUC minimum entry threshold: 0-63.75 dB The minimum CINR
            required to start using this DIUC when changing from a more
            robust DIUC is required, in 0.25 dB units."
        REFERENCE
            "Section 11.4.1, table 290, in IEEE 802.16REVd/D3-2004"
        ::= { wmanIfOfdmDcdBurstProfileEntry 6 }


wmanIfOfdmDcdBurstProfileRowStatus OBJECT-TYPE
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This object is used to create a new row or modify or
            delete an existing row in this table.

            If the implementator of this MIB has choosen not
            to implement 'dynamic assignment' of profiles, this
            object is not useful and should return noSuchName
            upon SNMP request."
        ::= { wmanIfOfdmDcdBurstProfileEntry 7 }

END
```