

IEEE P802.20a Draft Standard for Local and Metropolitan Area Networks—Standard Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility — Physical and Media Access Control Layer Specification — Amendment 1: MIB Enhancements and Corrigenda Items

NOTE—The editing instructions contained in this amendment define how to merge the material contained herein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in *bold italic*. The instruction “*Replace*” is used to make changes in text, figures or equations by removing the existing figure or equation and replacing it with a new one.

--Replace Chapter 17 of 802.20-2008 with the following text

17 MAC and PHY MIB

17.1 Overview

This chapter defines a Management Information Base (MIB) module for managing the MAC and PHY. For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to Section 7 of IETF RFC 3410.

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified here is compliant to the SMIV2, which is described in IETF STD 58, RFC 2578, RFC 2579, and RFC 2580.

17.2 MIB Structure

The MIB structure is based on the architecture reference model in Figure 1 and the layering architecture for the air interface Figure 2. The MIB object is composed of two groups:

- dot20An: This group contains managed objects defined for the access network.
- dot20Cmn: This group contains managed objects defined for the access network and the access terminal.

17.3 Security Considerations

This MIB relates to a system which provides mobile broadband wireless access. As such, improper manipulation of the objects represented by this MIB can result in denial of service to a large number of end-users.

The MIB objects in the Dot20AnChannelBandsEntry SEQUENCE contain 8 objects used to set the frequency band of the transmitting base station. An administrator should take great care to include

1 only authorized, licensed channel bands in the table. Failure to take these measures might cause a
2 base station to violate local regulatory laws (e.g. FCC licensing in the USA) by transmitting power
3 into unauthorized channels in the country where the base station is deployed.

4 The Dot20AnTransmitPower OBJECT sets the power for the base station in dBm. Unauthorized
5 access to this object could allow an attacker to boost power and violate local regulatory laws (e.g.
6 FCC licensing in the USA) by transmitting excessive power into a licensed band. This could also
7 lead to excessive sideband emissions in adjacent bands.

8 The Dot20AnNeighborListEntry SEQUENCE defines information about adjacent sectors that is
9 broadcast by the overhead channels of a base station. Terminals functioning in any sector can read the
10 overhead channels from other sectors, including those whose MIB may have become compromised or
11 corrupted due to unauthorized access. Such terminals could therefore incorporate incorrect handoff
12 information into their databases of potential sectors for handoff. Thus, unauthorized access to the
13 MIB in one sector, can affect the performance and handoff characteristics of terminals operating
14 correctly in adjacent sectors.

15 There are no MIB objects that could allow a user to increase their access rights to system service
16 levels. None of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other
17 than not-accessible) can be considered capable of revealing sensitive or vulnerable personal
18 information. This MIB is not capable of revealing user information that could violate privacy laws.

19 There are no MIB objects that could be used to turn off or change the security parameter
20 configuration of an 802.20 access node. The presence or absence of security (encryption,
21 authentication) is controlled by the session state record for each individual user, and cannot be
22 modified by an attacker accessing the MIB.

23 SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is
24 secure (for example by using IPsec), there is no control as to who on the secure network is allowed to
25 access and GET/SET (read/change/create/delete) the objects in this MIB module.

26 It is recommended that implementers consider the security features as provided by the SNMPv3
27 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic
28 mechanisms (for authentication and privacy).

29 Further, deployment of SNMP versions prior to SNMPv3 is not recommended. Instead, it is
30 recommended to deploy SNMPv3 and to enable cryptographic security. It is a customer/operator
31 responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is
32 properly configured to give access to the objects only to those that have legitimate rights to indeed
33 GET or SET (change/create/delete) them.

34

35 17.4 IANA Considerations

36 No IANA actions are required by this document.

37

17.5 Definition

```

1
2
3 IEEE802dot20-MIB DEFINITIONS ::= BEGIN
4
5 IMPORTS
6     ifIndex
7         FROM IF-MIB
8     MODULE-COMPLIANCE, OBJECT-GROUP
9         FROM SNMPv2-CONF
10    Counter32, Counter64, Integer32, MODULE-IDENTITY, OBJECT-IDENTITY,
11    OBJECT-TYPE, transmission
12        FROM SNMPv2-SMI
13    RowPointer, RowStatus, TEXTUAL-CONVENTION, TruthValue
14        FROM SNMPv2-TC
15    ;
16
17 ieee802dot20 MODULE-IDENTITY
18     LAST-UPDATED "200811031622PDT" -- November 03, 2008
19     ORGANIZATION
20         "IEEE 802.20"
21     CONTACT-INFO
22         "Contact: IEEE 802.20 Working Group
23         Postal:
24
25         Tel:
26         Fax:
27         E-mail: "
28     DESCRIPTION
29         "The MIB module for IEEE 802.20 entities.
30         (The transmission oid used for this MIB needs to be updated
31         when a valid one is obtained from IANA along with the new
32         802.20 ifType)"
33     ::= { transmission 9999 }
34
35 Dot20AnChannelBandsEntry ::= SEQUENCE
36 {
37     dot20AnChannelBandIndex      Integer32,
38     dot20AnSystemType           Integer32,
39     dot20AnBandClass            Integer32,
40     dot20AnChannelNumber        Integer32,
41     dot20AnHalfDuplexSupported  TruthValue,
42     dot20AnReverseChannelBandClass Integer32,
43     dot20AnReverseChannelNumber Integer32,
44     dot20AnCyclicPrefixLength  Integer32,
45     dot20AnFFTSize             Integer32,
46     dot20AnCBNumGuardSubcarriers Integer32,
47     dot20AnChannelBandShortId   Integer32,
48     dot20AnChannelBandAccessHashMask Integer32,
49     dot20AnChannelBandStatus    RowStatus
50 }
51
52 Dot20AnIdleStateStatsEntry ::= SEQUENCE
53 {
54     dot20AnAccessAttemptCounts  Counter32,
55     dot20AnAccessAttemptFailCounts Counter32,
56     dot20AnPageAttemptCounts    Counter32,
57     dot20AnPageFailureCounts    Counter32
58 }
59
60 Dot20AnNeighborListEntry ::= SEQUENCE
61 {
62     dot20AnNeighborIndex        Integer32,
63     dot20AnNeighborSectorPointer RowPointer,
64     dot20AnNeighborRowStatus    RowStatus
65 }
66

```

```

1  Dot20AnNeighborSectorsEntry ::= SEQUENCE
2  {
3      dot20AnNeighborSectorIndex      Integer32,
4      dot20AnNeighborPilotID          Integer32,
5      dot20AnNeighborEffTransmitPower Integer32,
6      dot20AnNeighborChannelBandRef   Integer32,
7      dot20AnNeighborChannelShortID   Integer32,
8      dot20AnNeighborSameANAsPrimSect TruthValue,
9      dot20AnNeighborSectorPilotGrpId Integer32,
10     dot20AnNeighborSynchGroupId      Integer32,
11     dot20AnNeighborSectorCellGroupId Integer32,
12     dot20AnNeighborSectorStatus      RowStatus
13 }
14
15 Dot20AnOtherTechNghbrsEntry ::= SEQUENCE
16 {
17     dot20AnOtherTechnologyIndex      Integer32,
18     dot20AnTechnologyType            Integer32,
19     dot20AnTechNghbrListLength       Integer32,
20     dot20AnTechnologyNeighborList    OCTET STRING,
21     dot20AnOtherTechNghbrRowStatus   RowStatus
22 }
23
24 Dot20AnSecondaryRegZoneCodeEntry ::= SEQUENCE
25 {
26     dot20AnSecondaryRegZoneCodeIndex Integer32,
27     dot20AnSecRegZoneCode            Integer32,
28     dot20AnSecondaryRegZoneRowStatus RowStatus
29 }
30
31 Dot20AnSectorCdmaSubSegEntry ::= SEQUENCE
32 {
33     dot20AnInterlaceId                Integer32,
34     dot20AnCdmaSubSegmentNum          Integer32,
35     dot20AnSectorCdmaSubSegRowStatus  RowStatus
36 }
37
38 Dot20AnSectorConfigEntry ::= SEQUENCE
39 {
40     dot20AnTotalNumSubcarriers        Integer32,
41     dot20AnNumGuardSubcarriers        Integer32,
42     dot20AnFlSubzoneSize              Integer32,
43     dot20AnResourceChannelMuxMode     Integer32,
44     dot20AnNumDRCHSubzones            Integer32,
45     dot20AnFLReservedInterlaces       INTEGER,
46     dot20AnNumFLReservedSubzones      Integer32,
47     dot20AnCpichHoppingMode           Integer32,
48     dot20AnNumEffectiveAntennas       Integer32,
49     dot20AnNumCommonSegmentHopPorts   Integer32,
50     dot20AnNumLABSegments             Integer32,
51     dot20AnMinScchResourceIndex       Integer32,
52     dot20AnSinglePAForXCarriers       Integer32,
53     dot20AnFlSdmaNumSubtrees          Integer32,
54     dot20AnFDPICHCodeOffsetSubtree0   Integer32,
55     dot20AnFDPICHCodeOffsetSubtree1   Integer32,
56     dot20AnFDPICHCodeOffsetSubtree2   Integer32,
57     dot20AnFDPICHCodeOffsetSubtree3   Integer32,
58     dot20AnNumCmnPilotTxAnt           Integer32,
59     dot20AnModSymbolsPerQPSKLAB       Integer32,
60     dot20AnUseDrchForFlcs             Integer32,
61     dot20AnEnableExpandedQPCH         TruthValue,
62     dot20AnSectorConfigRowStatus      RowStatus
63 }
64
65 Dot20AnSectorExtChanInfoEntry ::= SEQUENCE
66 {
67     dot20AnPilotID                    Integer32,
68     dot20AnHalfDuplexModeSupported    TruthValue,

```

```

1      dot20AnFACKBandwidthFactor      Integer32,
2      dot20AnSFNCellID                Integer32,
3      dot20AnCellNullID               Integer32,
4      dot20AnMaxNumSharedLABs         Integer32,
5      dot20AnMaxNumLABs               Integer32,
6      dot20AnMax16QamScchBlocks       Integer32,
7      dot20AnPdCabResSharingEnabled   TruthValue,
8      dot20AnNumAckableLABs           Integer32,
9      dot20An16QamScchT2PRatio        INTEGER,
10     dot20AnEffectiveTransmitPower    Integer32,
11     dot20AnAssignmentAckHARQTx      Integer32,
12     dot20AnCQIPilotTransmitPower    Integer32,
13     dot20AnCommonPilotTransmitPower Integer32,
14     dot20AnCDMAInterlacesBitmap     Integer32,
15     dot20AnNumOdccchReports          Integer32,
16     dot20AnNumRLCdmaSubsegments     Integer32,
17     dot20AnRackBandwidthFactor       Integer32,
18     dot20AnRlNumSdmaDimensions       Integer32,
19     dot20AnRlDpichCodeOffsetSubtree0 Integer32,
20     dot20AnRlDpichCodeOffsetSubtree1 Integer32,
21     dot20AnRlDpichCodeOffsetSubtree2 Integer32,
22     dot20AnRlDpichCodeOffsetSubtree3 Integer32,
23     dot20AnRlSubzoneSize             Integer32,
24     dot20AnSilenceIntervalPeriod     Integer32,
25     dot20AnSilenceIntervalDuration  Integer32,
26     dot20AnNumSilenceIntervalSubzone Integer32,
27     dot20AnAckInterferenceOffset     Integer32,
28     dot20AnMacIdRange                INTEGER,
29     dot20AnFlPcReportInterval        Integer32,
30     dot20AnFlPqiReportInterval       Integer32,
31     dot20AnFlIotReportInterval       Integer32,
32     dot20AnFastIoTEnabled            TruthValue,
33     dot20AnFastOSIEnabled            TruthValue,
34     dot20AnRabEnabled                TruthValue,
35     dot20AnOsiResponseMode           INTEGER,
36     dot20AnSlowInterferenceOffset    Integer32,
37     dot20AnCtrlAccessOffset          Integer32,
38     dot20AnRlAuxPilotPower           Integer32,
39     dot20AnReqQoSPowerBoost          Integer32,
40     dot20AnErasureTargetCtoI0        Integer32,
41     dot20AnErasureTargetCtoI1        Integer32,
42     dot20AnErasureTargetCtoI2        Integer32,
43     dot20AnErasureTargetCtoI3        Integer32,
44     dot20AnAccessCycleDuration       Integer32,
45     dot20AnMaxProbesPerSequence      Integer32,
46     dot20AnProbeRampUpStepSize       Integer32,
47     dot20AnPilotThreshold1           Integer32,
48     dot20AnPilotThreshold2           Integer32,
49     dot20AnOpenLoopAdjust            Integer32,
50     dot20AnAccessRetryPersistence0   Integer32,
51     dot20AnAccessRetryPersistence1   Integer32,
52     dot20AnAccessRetryPersistence2   Integer32,
53     dot20AnAccessRetryPersistence3   Integer32,
54     dot20AnAccessRetryPersistence4   Integer32,
55     dot20AnAccessRetryPersistence5   Integer32,
56     dot20AnAccessRetryPersistence6   Integer32,
57     dot20AnAccessRetryPersistence7   Integer32,
58     dot20AnSectorExtChanRowStatus    RowStatus
59 }
60
61 Dot20AnSectorGrpResSetsEntry ::= SEQUENCE
62 {
63     dot20AnResourceSetId              Integer32,
64     dot20AnResourceSetBitmap          Integer32,
65     dot20AnBRCHSubzoneCyclingEnabled  TruthValue,
66     dot20AnResourceSetSubZoneSpacing  Integer32,
67     dot20AnNumResourceSubzones        Integer32,
68     dot20AnResourceSubzoneOffset      Integer32,

```

```

1      dot20AnResourceSetRowStatus      RowStatus
2  }
3
4  Dot20AnSectorIpsiEntry ::= SEQUENCE
5  {
6      dot20AnIpsiIndex      Integer32,
7      dot20AnSupportedIpsi Integer32,
8      dot20AnIpsiRowStatus RowStatus
9  }
10
11 Dot20AnSectorParamEntry ::= SEQUENCE
12 {
13     dot20AnMobileCountryCode Integer32,
14     dot20AnMobileNetworkCode Integer32,
15     dot20AnSectorID          OCTET STRING,
16     dot20AnChannelBandRef    Integer32,
17     dot20AnLatitude          Integer32,
18     dot20AnLongitude         Integer32,
19     dot20AnLeapSeconds       Integer32,
20     dot20AnLocalTimeOffset   Integer32,
21     dot20AnPrimaryRegZoneCode Integer32,
22     dot20AnAnGroupId         Integer32,
23     dot20AnPilotGroupId       Integer32,
24     dot20AnSynchronousGroupId Integer32,
25     dot20AnCellGroupId        Integer32,
26     dot20AnSectorParamRowStatus RowStatus
27 }
28
29 Dot20AnSectorToIfIndexEntry ::= SEQUENCE
30 {
31     dot20AnIfChannelBandRef Integer32
32 }
33
34 Dot20CmnAuthStatsEntry ::= SEQUENCE
35 {
36     dot20CmnAuthFailureCounts Counter64,
37     dot20CmnAuthSuccessCounts Counter64
38 }
39
40 Dot20CmnLMACPacketStatsEntry ::= SEQUENCE
41 {
42     dot20CmnPacketFormatIndex Integer32,
43     dot20CmnARQAttemptsIndex  Integer32,
44     dot20CmnFwdTxPacketCounts Counter64,
45     dot20CmnRevRxPacketCounts Counter64
46 }
47
48 Dot20CmnLMACStatsEntry ::= SEQUENCE
49 {
50     dot20CmnFLABCounts      Counter64,
51     dot20CmnRLABCounts      Counter64,
52     dot20CmnAccessGrantCounts Counter64
53 }
54
55 Dot20CmnQmpStatsEntry ::= SEQUENCE
56 {
57     dot20CmnActiveReservationsCounts Counter64,
58     dot20CmnIdleReservationsCounts   Counter64,
59     dot20CmnReservationOpenCounts    Counter64,
60     dot20CmnReservationCloseCounts   Counter64,
61     dot20CmnReservationFailCounts    Counter64
62 }
63
64 Dot20CmnRlpStatsEntry ::= SEQUENCE
65 {
66     dot20CmnStreamId          Integer32,
67     dot20CmnRlpTxBytes        Counter64,
68     dot20CmnRlpReTxBytes      Counter64,

```

```

1      dot20CmnRlpTxDropBytes      Counter64,
2      dot20CmnRlpTxStatus        Counter64,
3      dot20CmnRlpRxBytes        Counter64,
4      dot20CmnRlpRxStatus        Counter64,
5      dot20CmnRlpTxPackets       Counter64,
6      dot20CmnRlpReTxPackets     Counter64,
7      dot20CmnRlpTxrDropPackets Counter64,
8      dot20CmnRlpRxPackets       Counter64,
9      dot20CmnRlpTxNAKTimeouts   Counter64,
10     dot20CmnRlpTxACKTimeouts   Counter64
11 }
12
13 dot20An OBJECT-IDENTITY
14     STATUS          current
15     DESCRIPTION
16         "AN specific configuration and statistics."
17     ::= { ieee802dot20 1 }
18
19 dot20AnMac OBJECT-IDENTITY
20     STATUS          current
21     DESCRIPTION
22         "MAC layer objects"
23     ::= { dot20An 1 }
24
25 dot20AnConnectionControl OBJECT IDENTIFIER ::= { dot20AnMac 3 }
26
27 dot20AnIdleState OBJECT IDENTIFIER ::= { dot20AnConnectionControl 1 }
28
29 dot20AnIdleStateStatsTable OBJECT-TYPE
30     SYNTAX          SEQUENCE OF Dot20AnIdleStateStatsEntry
31     MAX-ACCESS      not-accessible
32     STATUS          current
33     DESCRIPTION
34         "This table provides one row of Idle State protocol statistics
35         per 802.20 interface (i.e. sector for a specific ChannelBand)
36         and carrier."
37     ::= { dot20AnIdleState 1 }
38
39 dot20AnIdleStateStatsEntry OBJECT-TYPE
40     SYNTAX          Dot20AnIdleStateStatsEntry
41     MAX-ACCESS      not-accessible
42     STATUS          current
43     DESCRIPTION
44         "An Entry (conceptual row) in the IdleStateStats table. This
45         table is indexed by ifIndex and CarrierID(see 11.5.5.8). ifIndex:
46     Each IEEE
47         802.20 interface (uniquely identified by SectorID) is
48         represented by an ifEntry. In the case of a multicarrier
49         Sector, the carrierID indentifies one specific carrier."
50     REFERENCE
51         "IEEE Std. 802.20-2008, Subclause 8.4 (Access Channel MAC
52         Protocol)"
53     INDEX
54         { ifIndex }
55     ::= { dot20AnIdleStateStatsTable 1 }
56
57 dot20AnAccessAttemptCounts OBJECT-TYPE
58     SYNTAX          Counter32
59     MAX-ACCESS      read-only
60     STATUS          current
61     DESCRIPTION
62         "Number of Access Attempts among all Terminals"
63     REFERENCE
64         "IEEE Std. 802.20-2008, Subclause 8.4.5.5.2,
65         (Access Channel MAC Protocol / AN Requirements)"
66     ::= { dot20AnIdleStateStatsEntry 1 }
67
68 dot20AnAccessAttemptFailCounts OBJECT-TYPE

```

```

1      SYNTAX      Counter32
2      MAX-ACCESS  read-only
3      STATUS      current
4      DESCRIPTION
5          "Number of Failed Access Attempts among all Terminals.
6          Incremented when access RLAB is not used by a terminal."
7      REFERENCE
8          "IEEE Std. 802.20-2008, Subclause 11.5.4.3.2 (BindATI), and
9          Subclause 11.2.4.6.2.1 (issuing ConnectedState.Deactivate)"
10     ::= { dot20AnIdleStateStatsEntry 2 }
11
12     dot20AnPageAttemptCounts OBJECT-TYPE
13     SYNTAX      Counter32
14     MAX-ACCESS  read-only
15     STATUS      current
16     DESCRIPTION
17         "Number of Page Attempts"
18     REFERENCE
19         "IEEE Std. 802.20-2008, Subclause 8.3.5.8, and Table 208
20     (RouteOpenRequestReason encoding)"
21     ::= { dot20AnIdleStateStatsEntry 3 }
22
23     dot20AnPageFailureCounts OBJECT-TYPE
24     SYNTAX      Counter32
25     MAX-ACCESS  read-only
26     STATUS      current
27     DESCRIPTION
28         "Number of Failed Page Attempts"
29     REFERENCE
30         "IEEE Std. 802.20-2008, Subclause 8.3.5.8, and Table 358
31     (RouteOpenRequestReason encoding)"
32     ::= { dot20AnIdleStateStatsEntry 4 }
33
34     dot20AnOverheadMessages OBJECT IDENTIFIER ::= { dot20AnConnectionControl 4 }
35
36     dot20AnSectorConfigTable OBJECT-TYPE
37     SYNTAX      SEQUENCE OF Dot20AnSectorConfigEntry
38     MAX-ACCESS  not-accessible
39     STATUS      current
40     DESCRIPTION
41         "This table provides one row per 802.20 interface, i.e. sector
42         for a specific ChannelBand. This table's attributes specify the
43         configuration of the corresponding sector, and can be used to
44         populate fields in SystemInfo block and QuickChannelInfo
45         message, which are transmitted by the Overhead Messages Protocol."
46     ::= { dot20AnOverheadMessages 1 }
47
48     dot20AnSectorConfigEntry OBJECT-TYPE
49     SYNTAX      Dot20AnSectorConfigEntry
50     MAX-ACCESS  not-accessible
51     STATUS      current
52     DESCRIPTION
53         "An Entry (conceptual row) in the SectorConfig table. This
54         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
55         interface (uniquely identified by SectorID) is represented by
56         an ifEntry."
57     REFERENCE
58         "IEEE Std. 802.20-2008, Subclause 11.6 (Overhead Messages Protocol)"
59     INDEX
60         { ifIndex }
61     ::= { dot20AnSectorConfigTable 1 }
62
63     dot20AnTotalNumSubcarriers OBJECT-TYPE
64     SYNTAX      Integer32 (0..7)
65     MAX-ACCESS  read-write
66     STATUS      current
67     DESCRIPTION
68         "This parameter takes the value 2^(7+n), where n is the

```



```

1         value of the 3 bit field. This field is not to be set to a
2         value of 5 or above."
3     REFERENCE
4         "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
5     ::= { dot20AnSectorConfigEntry 29 }
6
7     dot20AnNumGuardSubcarriers OBJECT-TYPE
8     SYNTAX      Integer32 (0..7)
9     MAX-ACCESS  read-write
10    STATUS      current
11    DESCRIPTION
12        "This attribute determines the number of guard subcarriers
13         as defined in 802.20 Physical layer clause."
14    REFERENCE
15        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
16    ::= { dot20AnSectorConfigEntry 30 }
17
18    dot20AnFlSubzoneSize OBJECT-TYPE
19    SYNTAX      Integer32 (0..1)
20    MAX-ACCESS  read-write
21    STATUS      current
22    DESCRIPTION
23        "This field determines the number of subzones on the
24         forward link. If n=0, this parameter is set to 64 and if
25         n=1, this parameter is set to 128."
26    REFERENCE
27        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
28    ::= { dot20AnSectorConfigEntry 31 }
29
30    dot20AnResourceChannelMuxMode OBJECT-TYPE
31    SYNTAX      Integer32 (0..1)
32    MAX-ACCESS  read-write
33    STATUS      current
34    DESCRIPTION
35        "This field determines the number of subzones on the
36         forward link. If ResourceChannelMuxMode=0, this parameter is set to
37         64 and if
38         ResourceChannelMuxMode=1, this parameter is set to 128."
39    REFERENCE
40        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo Block)"
41    ::= { dot20AnSectorConfigEntry 32 }
42
43    dot20AnNumDRCHSubzones OBJECT-TYPE
44    SYNTAX      Integer32
45    MAX-ACCESS  read-write
46    STATUS      current
47    DESCRIPTION
48        "This field takes values between 0 and N_FFT/64 - 1"
49    REFERENCE
50        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo Block)"
51    ::= { dot20AnSectorConfigEntry 33 }
52
53    dot20AnFLReservedInterlaces OBJECT-TYPE
54    SYNTAX      INTEGER {
55        zero(1),
56        zeroToOne(2),
57        zeroToTwo(3),
58        zeroToThree(4),
59        zeroToFour(5),
60        zeroToFive(6),
61        zeroToSix(7),
62        zeroToSeven(8),
63        zeroAndThree(9),
64        zeroAndSix(10),
65        zeroTwoAndFour(11),
66        zeroTwoFourAndSix(12),
67        reserved(13),
68        reserved2(14),

```

```

1         reserved3(15),
2         none(16)
3     }
4     MAX-ACCESS    read-write
5     STATUS       current
6     DESCRIPTION
7         "This attribute determines which interlaces contain
8         reserved bandwidth on the forward link."
9     REFERENCE
10        "IEEE Std. 802.20-2008, Table 285 (Interpretation of FL
11        Reserved Interlaces), Subclause 11.6.5.2"
12        ::= { dot20AnSectorConfigEntry 34 }
13
14 dot20AnNumFLReservedSubzones OBJECT-TYPE
15     SYNTAX       Integer32 (0..15)
16     MAX-ACCESS   read-write
17     STATUS       current
18     DESCRIPTION
19         "This field determines the number of subzones that are reserved
20         on each interlace that contains reserved bandwidth"
21     REFERENCE
22         "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
23         ::= { dot20AnSectorConfigEntry 35 }
24
25 dot20AnCpichHoppingMode OBJECT-TYPE
26     SYNTAX       Integer32 (0..1)
27     MAX-ACCESS   read-write
28     STATUS       current
29     DESCRIPTION
30         "This field is set to 0 for deterministic, and 1 for
31         random hopping"
32     REFERENCE
33         "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
34         ::= { dot20AnSectorConfigEntry 36 }
35
36 dot20AnNumEffectiveAntennas OBJECT-TYPE
37     SYNTAX       Integer32 (1..8)
38     MAX-ACCESS   read-write
39     STATUS       current
40     DESCRIPTION
41         "This attribute determines the number of
42         effective antennas."
43     REFERENCE
44         "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
45         ::= { dot20AnSectorConfigEntry 37 }
46
47 dot20AnNumCommonSegmentHopPorts OBJECT-TYPE
48     SYNTAX       Integer32 (0..7)
49     MAX-ACCESS   read-write
50     STATUS       current
51     DESCRIPTION
52         "This attribute determines the number of common segment
53         hop ports."
54     REFERENCE
55         "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
56         ::= { dot20AnSectorConfigEntry 38 }
57
58 dot20AnNumLABSegments OBJECT-TYPE
59     SYNTAX       Integer32 (0..7)
60     MAX-ACCESS   read-write
61     STATUS       current
62     DESCRIPTION
63         "This field indicates the number of LABSegments."
64     REFERENCE
65         "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
66         ::= { dot20AnSectorConfigEntry 39 }
67
68 dot20AnMinScchResourceIndex OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (0..31)
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "This parameter is in units of N_FFT/32 resources, and spans
6          from 0 to N_FFT -1"
7      REFERENCE
8          "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
9      ::= { dot20AnSectorConfigEntry 40 }
10
11     dot20AnSinglePAForXCarriers OBJECT-TYPE
12         SYNTAX      Integer32 (0..1)
13         MAX-ACCESS  read-write
14         STATUS      current
15         DESCRIPTION
16             "This field determines the structure of F-BPICH (SinglePAForMultipleChannelBands)"
17         REFERENCE
18             "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
19         ::= { dot20AnSectorConfigEntry 41 }
20
21     dot20AnFlSdmaNumSubtrees OBJECT-TYPE
22         SYNTAX      Integer32 (1..4)
23         MAX-ACCESS  read-write
24         STATUS      current
25         DESCRIPTION
26             "This field determines the number of sub-trees on the
27             forward link. (FLNumSDMADimensions)"
28         REFERENCE
29             "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
30         ::= { dot20AnSectorConfigEntry 42 }
31
32     dot20AnFLDPICHCodeOffsetSubtreeIndex0 OBJECT-TYPE
33         SYNTAX      Integer32 (0..3)
34         MAX-ACCESS  read-write
35         STATUS      current
36         DESCRIPTION
37             "This field is set to the code offset for subtree
38             0. This subtree is always present, and is therefore not
39             described in the overhead channels."
40         REFERENCE
41             "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
42             Pilot Channel) and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
43         ::= { dot20AnSectorConfigEntry 43 }
44
45     dot20AnFLDPICHCodeOffsetSubtreeIndex1 OBJECT-TYPE
46         SYNTAX      Integer32 (0..3)
47         MAX-ACCESS  read-write
48         STATUS      current
49         DESCRIPTION
50             "This field is set to the code offset for subtree
51             1"
52         REFERENCE
53             "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
54             Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
55         ::= { dot20AnSectorConfigEntry 44 }
56
57     dot20AnFLDPICHCodeOffsetSubtree2 OBJECT-TYPE
58         SYNTAX      Integer32 (0..3)
59         MAX-ACCESS  read-write
60         STATUS      current
61         DESCRIPTION
62             "This field is set to the code offset for subtree
63             2"
64         REFERENCE
65             "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
66             Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
67         ::= { dot20AnSectorConfigEntry 45 }

```

```

1
2 dot20AnFLDPICHCodeOffsetSubtreeIndex3 OBJECT-TYPE
3     SYNTAX      Integer32 (0..3)
4     MAX-ACCESS  read-write
5     STATUS      current
6     DESCRIPTION
7         "This field is set to the code offset for subtree
8         3"
9     REFERENCE
10        "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
11        Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
12        ::= { dot20AnSectorConfigEntry 46 }
13
14 dot20AnNumCmnPilotTxAnt OBJECT-TYPE
15     SYNTAX      Integer32 (1..4)
16     MAX-ACCESS  read-write
17     STATUS      current
18     DESCRIPTION
19        "This attribute specifies the number of common pilot
20        transmit antennas."
21     REFERENCE
22        "IEEE Std. 802.20-2008, Table 90 (NumEffectiveAntennas), Subclause
23        9.4.1.2.3.1.1 (Forward
24        Common Pilot Channel Subcarriers), and Subclause 11.6.5.3
25        (QuickChannelInfo Block)"
26        ::= { dot20AnSectorConfigEntry 47 }
27
28 dot20AnModSymbolsPerQPSKLAB OBJECT-TYPE
29     SYNTAX      Integer32 (0..4)
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33        "This field determines the number of modulation symbols
34        for each block carried by the F-SCCH"
35     REFERENCE
36        "IEEE Std. 802.20-2008, Table 287 (Interpretation of
37        ModulationSymbolsPerQPSKLAB)"
38        ::= { dot20AnSectorConfigEntry 48 }
39
40 dot20AnUseDrchForFlcs OBJECT-TYPE
41     SYNTAX      Integer32 (0..1)
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45        "This field determines the hopping pattern on the FLCS. It is set to
46        1 if the hopping pattern is DRCH on the FLCS, and is set to 0 otherwise"
47     REFERENCE
48        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo
49        Block)"
50        ::= { dot20AnSectorConfigEntry 49 }
51
52 dot20AnEnableExpandedQPCH OBJECT-TYPE
53     SYNTAX      TruthValue
54     MAX-ACCESS  read-write
55     STATUS      current
56     DESCRIPTION
57        "This field determines the number of packets delivered to
58        the Physical Layer by the MAC Layer"
59     REFERENCE
60        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo
61        Block)"
62        ::= { dot20AnSectorConfigEntry 50 }
63
64 dot20AnSectorConfigRowStatus OBJECT-TYPE
65     SYNTAX      RowStatus
66     MAX-ACCESS  read-create
67     STATUS      current
68     DESCRIPTION

```

```

1         "The status column used for creating, modifying, and deleting
2         instances of the columnar objects in the SectorConfig Table. If
3         the implementer of this MIB has chosen not to implement
4         'dynamic assignment' of sectors, this attribute is not applicable
5         and should return noSuchName upon SNMP request."
6     DEFVAL      { active }
7     ::= { dot20AnSectorConfigEntry 78 }
8
9     dot20AnSectorExtChanInfoTable OBJECT-TYPE
10        SYNTAX      SEQUENCE OF Dot20AnSectorExtChanInfoEntry
11        MAX-ACCESS   not-accessible
12        STATUS      current
13        DESCRIPTION
14            "This table provides one row per 802.20 interface, i.e. sector
15            for a specific ChannelBand. This table's attributes specify the
16            configuration of the corresponding sector, and can be used to
17            populate fields in extendedChannelInfo message."
18        ::= { dot20AnOverheadMessages 2 }
19
20     dot20AnSectorExtChanInfoEntry OBJECT-TYPE
21        SYNTAX      Dot20AnSectorExtChanInfoEntry
22        MAX-ACCESS   not-accessible
23        STATUS      current
24        DESCRIPTION
25            "An Entry (conceptual row) in the SectorExtChanInfo table. This
26            table is indexed by IfIndex. ifIndex: Each IEEE 802.20
27            interface (uniquely identified by SectorID) is represented by
28            an ifEntry. The Extended Channel Info is transmitted by the
29            Overhead Messages Protocol."
30        REFERENCE
31            "IEEE Std. 802.20-2008, Subclause 11.6.5.4 (ExtendedChannelInfo)"
32        INDEX
33            { ifIndex }
34        ::= { dot20AnSectorExtChanInfoTable 1 }
35
36     dot20AnPilotID OBJECT-TYPE
37        SYNTAX      Integer32 (0..1023)
38        MAX-ACCESS   read-write
39        STATUS      current
40        DESCRIPTION
41            "This attribute is set to the PilotID of the sector."
42        REFERENCE
43            "IEEE Std. 802.20-2008, Subclause 9.2.2.2.3 (PilotID and SectorSeed)"
44        ::= { dot20AnSectorExtChanInfoEntry 1 }
45
46     dot20AnHalfDuplexModeSupported OBJECT-TYPE
47        SYNTAX      TruthValue
48        MAX-ACCESS   read-write
49        STATUS      current
50        DESCRIPTION
51            "This attribute is set to True if the access network
52            supports half duplex terminals, and is set to False
53            otherwise. If half-duplex terminals are supported, the access
54            network should assign MAC IDs and channel assignments in a
55            manner that enables half-duplex terminal operation. A
56            half-duplex access terminal is not required to monitor forward
57            link transmissions on a PHY Frame where it is scheduled to make
58            a reverse link transmission."
59        REFERENCE
60            "IEEE Std. 802.20-2008, Subclause 11.7.5.4 (MACResourceAssignment)"
61        ::= { dot20AnSectorExtChanInfoEntry 2 }
62
63     dot20AnFACKBandwidthFactor OBJECT-TYPE
64        SYNTAX      Integer32 (1..4)
65        MAX-ACCESS   read-write
66        STATUS      current
67        DESCRIPTION
68            "Forward Acknowledgement channel (FACK) bandwidth factor"

```

```

1     REFERENCE
2     "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
3     ::= { dot20AnSectorExtChanInfoEntry 3 }
4
5     dot20AnSFNCellID OBJECT-TYPE
6     SYNTAX      Integer32 (0..511)
7     MAX-ACCESS  read-write
8     STATUS      current
9     DESCRIPTION
10    "This field determines the ID of the single frequency network
11    cell (for Broadcast Multicast Service)"
12    REFERENCE
13    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup),
14    and Subclause 9.2.2.2.2 (SFNCellID and SFNPhase)"
15    ::= { dot20AnSectorExtChanInfoEntry 5 }
16
17    dot20AnCellNullID OBJECT-TYPE
18    SYNTAX      Integer32 (0..511)
19    MAX-ACCESS  read-write
20    STATUS      current
21    DESCRIPTION
22    "Cell Null Id"
23    REFERENCE
24    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
25    ::= { dot20AnSectorExtChanInfoEntry 6 }
26
27    dot20AnMaxNumSharedLABs OBJECT-TYPE
28    SYNTAX      Integer32 (1..4)
29    MAX-ACCESS  read-write
30    STATUS      current
31    DESCRIPTION
32    "This field determines the maximum number of shared LABs
33    that are transmitted by this sector"
34    REFERENCE
35    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
36    ::= { dot20AnSectorExtChanInfoEntry 7 }
37
38    dot20AnMaxNumLABs OBJECT-TYPE
39    SYNTAX      Integer32 (0..63)
40    MAX-ACCESS  read-write
41    STATUS      current
42    DESCRIPTION
43    "This field is set to the Maximum number of LABs that can
44    be transmitted by this sector"
45    REFERENCE
46    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
47    ::= { dot20AnSectorExtChanInfoEntry 9 }
48
49    dot20AnMax16QamScchBlocks OBJECT-TYPE
50    SYNTAX      Integer32 (0..15)
51    MAX-ACCESS  read-write
52    STATUS      current
53    DESCRIPTION
54    "This field is set to the maximum number of 16-QAM blocks
55    that may be transmitted by the access network"
56    REFERENCE
57    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
58    ::= { dot20AnSectorExtChanInfoEntry 10 }
59
60    dot20AnPdCabResSharingEnabled OBJECT-TYPE
61    SYNTAX      TruthValue
62    MAX-ACCESS  read-write
63    STATUS      current
64    DESCRIPTION
65    "This field determines if resource sharing using PDCABs is
66    enabled"
67    REFERENCE
68    "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"

```

```

1      ::= { dot20AnSectorExtChanInfoEntry 11 }
2
3      dot20AnNumAckableLABs OBJECT-TYPE
4          SYNTAX          Integer32 (0..7)
5          MAX-ACCESS      read-write
6          STATUS          current
7          DESCRIPTION
8              "This field is set to the number of LABs on SCCH that the
9              access terminal is to acknowledge"
10         REFERENCE
11             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
12         ::= { dot20AnSectorExtChanInfoEntry 12 }
13
14     dot20An16QamScchT2PRatio OBJECT-TYPE
15         SYNTAX          INTEGER {
16             minusSevenDb(1),
17             minusFourDb(2),
18             zeroDb(3),
19             minusTenDb(4)
20         }
21         MAX-ACCESS      read-write
22         STATUS          current
23         DESCRIPTION
24             "16 Qam Scch T2P Ratio"
25         REFERENCE
26             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
27         ::= { dot20AnSectorExtChanInfoEntry 13 }
28
29     dot20AnEffectiveTransmitPower OBJECT-TYPE
30         SYNTAX          Integer32 (0..63)
31         MAX-ACCESS      read-write
32         STATUS          current
33         DESCRIPTION
34             "This attribute is set to the effective transmit power of the
35             sector in units of dBm"
36         REFERENCE
37             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
38         ::= { dot20AnSectorExtChanInfoEntry 14 }
39
40     dot20AnAssignmentAckHARQTx OBJECT-TYPE
41         SYNTAX          Integer32 (0..7)
42         MAX-ACCESS      read-write
43         STATUS          current
44         DESCRIPTION
45             "The value 0 indicates that no ACK is sent in response to an
46             assignment. The rules for interpreting other values of this
47             field are provided in the Lower MAC Sublayer. The value 7 is
48             reserved"
49         REFERENCE
50             "IEEE Std. 802.20-2008, Table 196, and Subclause 11.6.5.4.2
51             (ReverseChannelGroup)"
52         ::= { dot20AnSectorExtChanInfoEntry 15 }
53
54     dot20AnCQIPilotTransmitPower OBJECT-TYPE
55         SYNTAX          Integer32 (0..15)
56         MAX-ACCESS      read-write
57         STATUS          current
58         DESCRIPTION
59             "The field determines the power spectral density of the F-CQIPICH
60             relative to the reference transmit power density defined by the
61             Physical
62             Layer. This parameter may take the value (-4 + n*0.5) dB."
63         REFERENCE
64             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
65         ::= { dot20AnSectorExtChanInfoEntry 16 }
66
67     dot20AnCommonPilotTransmitPower OBJECT-TYPE
68         SYNTAX          Integer32 (0..15)

```

```

1     MAX-ACCESS    read-write
2     STATUS        current
3     DESCRIPTION
4         "The attribute's value (denoted n) determines the power
5         spectral density of the F-CPICH during the FL PHY frame
6         relative to the F-ACQCH. The pilot power density is equal
7         to (-4 + n*0.5) dB."
8     REFERENCE
9         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
10    ::= { dot20AnSectorExtChanInfoEntry 17 }
11
12    dot20AnCDMAInterlacesBitmap OBJECT-TYPE
13        SYNTAX      Integer32 (0..255)
14        MAX-ACCESS  read-write
15        STATUS      current
16        DESCRIPTION
17            "The j'th bit of this field is set to 1 if interlace i
18            contains a Reverse Link CDMA Segment. Here j is assumed to range
19            from 0 through 7, and an interlace i is the set of PHY Frames
20            that satisfy PHY Frame Index mod 8 = i"
21        REFERENCE
22            "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
23            (ReverseChannelGroup)"
24        ::= { dot20AnSectorExtChanInfoEntry 18 }
25
26    dot20AnNumOdcchReports OBJECT-TYPE
27        SYNTAX      Integer32 (0..31)
28        MAX-ACCESS  read-write
29        STATUS      current
30        DESCRIPTION
31            "Num ODCCH reports, specified in units of 16"
32        REFERENCE
33            "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
34            (ReverseChannelGroup)"
35        ::= { dot20AnSectorExtChanInfoEntry 27 }
36
37    dot20AnNumRLCdmaSubsegments OBJECT-TYPE
38        SYNTAX      Integer32 (1..16)
39        MAX-ACCESS  read-write
40        STATUS      current
41        DESCRIPTION
42            "This field determines the number of RLCdmaSubsegments on
43            this sector."
44        REFERENCE
45            "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
46            (ReverseChannelGroup)"
47        ::= { dot20AnSectorExtChanInfoEntry 28 }
48
49    dot20AnRackBandwidthFactor OBJECT-TYPE
50        SYNTAX      Integer32 (0..3)
51        MAX-ACCESS  read-write
52        STATUS      current
53        DESCRIPTION
54            "This parameter is set to 2^n, where n is the value of
55            the two bit field."
56        REFERENCE
57            "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
58            (ReverseChannelGroup)"
59        ::= { dot20AnSectorExtChanInfoEntry 30 }
60
61    dot20AnRlNumSdmaDimensions OBJECT-TYPE
62        SYNTAX      Integer32 (1..4)
63        MAX-ACCESS  read-write
64        STATUS      current
65        DESCRIPTION
66            "This field determines the number of spatial dimensions on
67            the reverse link."
68        REFERENCE

```



```

1         "IEEE Std. 802.20-2008, Table 137 and
2         Subclause 11.6.5.4.2 (ReverseChannelGroup)"
3         ::= { dot20AnSectorExtChanInfoEntry 31 }
4
5 dot20AnRLDpichCodeOffsetSubtreeIndex0 OBJECT-TYPE
6     SYNTAX      Integer32 (0..3)
7     MAX-ACCESS  read-write
8     STATUS      current
9     DESCRIPTION
10        "This field is set to the code offset for subtree 0"
11    REFERENCE
12        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
13 and Parameters), and
14        Subclause 11.6.5.4.2 (Reverse Channel Group)"
15        ::= { dot20AnSectorExtChanInfoEntry 32 }
16
17 dot20AnRLDpichCodeOffsetSubtreeIndex1 OBJECT-TYPE
18     SYNTAX      Integer32 (0..3)
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22        "This field is set to the code offset for subtreetree 1"
23    REFERENCE
24        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
25 and Parameters), and
26        Subclause 11.6.5.4.1 (Reverse Channel Group)"
27        ::= { dot20AnSectorExtChanInfoEntry 33 }
28
29 dot20AnRLDpichCodeOffsetSubtreeIndex2 OBJECT-TYPE
30     SYNTAX      Integer32 (0..3)
31     MAX-ACCESS  read-write
32     STATUS      current
33     DESCRIPTION
34        "This field is set to the code offset for tree 2"
35    REFERENCE
36        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
37 and Parameters), and
38        Subclause 11.6.5.4.1 (Reverse Channel Group)"
39        ::= { dot20AnSectorExtChanInfoEntry 34 }
40
41 dot20AnRLDpichCodeOffsetSubtreeIndex3 OBJECT-TYPE
42     SYNTAX      Integer32 (0..3)
43     MAX-ACCESS  read-write
44     STATUS      current
45     DESCRIPTION
46        "This field is set to the code offset for subtree 3"
47    REFERENCE
48        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
49 and Parameters), and
50        Subclause 11.6.5.4.1 (Reverse Channel Group)"
51        ::= { dot20AnSectorExtChanInfoEntry 35 }
52
53 dot20AnRLSubzoneSize OBJECT-TYPE
54     SYNTAX      Integer32 (0..1)
55     MAX-ACCESS  read-write
56     STATUS      current
57     DESCRIPTION
58        "This field determines the size of subzones on the reverse
59        link. If n=0, this parameter takes the value 64 and if
60        n=1, this parameter takes the value 128"
61    REFERENCE
62        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
63 and Parameters), and
64        Subclause 11.6.5.4.2 (Reverse Channel Group)"
65        ::= { dot20AnSectorExtChanInfoEntry 36 }
66
67 dot20AnSilenceIntervalPeriod OBJECT-TYPE
68     SYNTAX      Integer32 (0..15)

```

```

1     MAX-ACCESS    read-write
2     STATUS        current
3     DESCRIPTION
4         "This field determines the period in units of super frames
5         when the silence interval repeats. The SilenceInterval takes
6         a value of 2^n super frames, where n is the value of this parameter"
7     REFERENCE
8         "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
9     and Parameters), and
10        Subclause 11.6.5.4.2 (ReverseChannelGroup)"
11    ::= { dot20AnSectorExtChanInfoEntry 38 }
12
13    dot20AnSilenceIntervalDuration OBJECT-TYPE
14        SYNTAX      Integer32 (1..8)
15        MAX-ACCESS  read-write
16        STATUS      current
17        DESCRIPTION
18            "This field determines the duration silence interval in
19            units of 8 OFDM symbols"
20        REFERENCE
21            "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
22    and Parameters), and
23            Subclause 11.6.5.4.2 (ReverseChannelGroup)"
24        ::= { dot20AnSectorExtChanInfoEntry 39 }
25
26    dot20AnNumSilenceIntervalSubzone OBJECT-TYPE
27        SYNTAX      Integer32 (0..15)
28        MAX-ACCESS  read-write
29        STATUS      current
30        DESCRIPTION
31            "This field specifies the set of subzones that are blanked
32            during the silence interval."
33        REFERENCE
34            "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
35    and Parameters), and
36            Subclause 11.6.5.4.2 (ReverseChannelGroup)"
37        ::= { dot20AnSectorExtChanInfoEntry 40 }
38
39    dot20AnAckInterferenceOffset OBJECT-TYPE
40        SYNTAX      Integer32 (0..15)
41        MAX-ACCESS  read-write
42        STATUS      current
43        DESCRIPTION
44            "This field may take values in units of dB"
45        REFERENCE
46            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
47        ::= { dot20AnSectorExtChanInfoEntry 42 }
48
49    dot20AnMacIdRange OBJECT-TYPE
50        SYNTAX      INTEGER {
51            upTo63(1),
52            upTo127(2),
53            upTo255(3),
54            upTo511(4),
55            upTo1023(5),
56            upTo2047(6),
57            reserved(7),
58            upTo31(8)
59        }
60        MAX-ACCESS  read-write
61        STATUS      current
62        DESCRIPTION
63            "This field is set to indicate the range of assigned
64            MACID values in the sector. For example, a MACIDRange of 63
65            indicates that the sector has not assigned MACID values 64 and
66            above"
67        REFERENCE
68            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"

```

```

1      ::= { dot20AnSectorExtChanInfoEntry 43 }
2
3      dot20AnFlPcReportInterval OBJECT-TYPE
4          SYNTAX          Integer32 (0..7)
5          MAX-ACCESS      read-write
6          STATUS          current
7          DESCRIPTION
8              "This field determines the periodicity at which power
9              control commands are sent to the access terminal. This
10             parameter can take the value 2^n, where n is the value of the
11             three bit field."
12         REFERENCE
13             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
14         ::= { dot20AnSectorExtChanInfoEntry 44 }
15
16     dot20AnFlPqiReportInterval OBJECT-TYPE
17         SYNTAX          Integer32 (0..3)
18         MAX-ACCESS      read-write
19         STATUS          current
20         DESCRIPTION
21             "This field determines the periodicity at which PQI
22             reports commands are sent by this sector. This parameter
23             takes the value 16*2^n, where n is the value of the three bit
24             field"
25         REFERENCE
26             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
27         ::= { dot20AnSectorExtChanInfoEntry 45 }
28
29     dot20AnFlIotReportInterval OBJECT-TYPE
30         SYNTAX          Integer32 (0..3)
31         MAX-ACCESS      read-write
32         STATUS          current
33         DESCRIPTION
34             "This field determines the periodicity at which IoT values
35             are sent to the access terminal. This parameter takes the value 1, 8,
36             16, or 32, depending on the two bit field taking values 0, 1, 2, or 3.
37             "
38         REFERENCE
39             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
40         ::= { dot20AnSectorExtChanInfoEntry 46 }
41
42     dot20AnIoTEnabled OBJECT-TYPE
43         SYNTAX          TruthValue
44         MAX-ACCESS      read-write
45         STATUS          current
46         DESCRIPTION
47             "This field determines if the sector transmits IOT"
48         REFERENCE
49             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
50         ::= { dot20AnSectorExtChanInfoEntry 47 }
51
52     dot20AnFastOSIEnabled OBJECT-TYPE
53         SYNTAX          TruthValue
54         MAX-ACCESS      read-write
55         STATUS          current
56         DESCRIPTION
57             "This field determines if the access terminal is required
58             to read OSI from this sector"
59         REFERENCE
60             "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
61         ::= { dot20AnSectorExtChanInfoEntry 48 }
62
63     dot20AnRabEnabled OBJECT-TYPE
64         SYNTAX          TruthValue
65         MAX-ACCESS      read-write
66         STATUS          current
67         DESCRIPTION
68             "This field is set to TRUE if this sector transmits RAB, and

```

```

1         is set to FALSE otherwise"
2     REFERENCE
3         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
4     ::= { dot20AnSectorExtChanInfoEntry 49 }
5
6     dot20AnOsiResponseMode OBJECT-TYPE
7         SYNTAX      INTEGER {
8             stochastic(1),
9             deterministic(2)
10        }
11        MAX-ACCESS   read-write
12        STATUS       current
13        DESCRIPTION
14            "This field determines the type of response to OSI modes. It is set to
15            0 for stochastic response and is set to 1 for deterministic response."
16        REFERENCE
17            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
18        ::= { dot20AnSectorExtChanInfoEntry 50 }
19
20        dot20AnSlowInterferenceOffset OBJECT-TYPE
21            SYNTAX      Integer32 (0..15)
22            MAX-ACCESS   read-write
23            STATUS       current
24            DESCRIPTION
25                "This field is set in units of dB"
26            REFERENCE
27                "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
28            ::= { dot20AnSectorExtChanInfoEntry 51 }
29
30        dot20AnCtrlAccessOffset OBJECT-TYPE
31            SYNTAX      Integer32 (0..3)
32            MAX-ACCESS   read-write
33            STATUS       current
34            DESCRIPTION
35                "This field determines the initial gain of the R-CQICH over the
36                R-ACH. The value of this parameter is -11+n dB, where n
37                is the value of this parameter"
38            REFERENCE
39                "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
40            ::= { dot20AnSectorExtChanInfoEntry 52 }
41
42        dot20AnRlAuxPilotPower OBJECT-TYPE
43            SYNTAX      Integer32 (0..7)
44            MAX-ACCESS   read-write
45            STATUS       current
46            DESCRIPTION
47                "This field is determine the offset of R-AuxPICH with
48                respect to R-PICH. This parameter may take the value 4+n."
49            REFERENCE
50                "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
51                Attribute)"
52            ::= { dot20AnSectorExtChanInfoEntry 53 }
53
54        dot20AnReqQoSPowerBoost OBJECT-TYPE
55            SYNTAX      Integer32 (0..3)
56            MAX-ACCESS   read-write
57            STATUS       current
58            DESCRIPTION
59                " This parameter specifies a power boost value for r-reqch
60                transmissions on R-CDCCCH for flows that allow request boost, and takes the
61                values 0, 1, 3, or 5 dB when the two-bit field is set to '00', '01', '10', or
62                '11', respectively.
63                "
64            REFERENCE
65                "IEEE Std. 802.20-2008, Subclause 8.7.7.2.1 (PowerControl
66                Attribute)"
67            ::= { dot20AnSectorExtChanInfoEntry 54 }
68

```

```

1 dot20AnErasureTargetCtoI0 OBJECT-TYPE
2     SYNTAX      Integer32 (0..15)
3     MAX-ACCESS  read-write
4     STATUS      current
5     DESCRIPTION
6         "This attribute's value (denoted n) determines the target C/I value of
7         erasure sequences for different assignment sizes. The
8         transmit power is equal to n-12 dB."
9     REFERENCE
10        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
11        Attribute)"
12    ::= { dot20AnSectorExtChanInfoEntry 55 }
13
14 dot20AnErasureTargetCtoI1 OBJECT-TYPE
15     SYNTAX      Integer32 (0..15)
16     MAX-ACCESS  read-write
17     STATUS      current
18     DESCRIPTION
19        "This attribute's value (denoted n) determines the target C/I value of
20        erasure sequences for different assignment sizes. The
21        transmit power is equal to n-12 dB."
22    REFERENCE
23        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
24        Attribute)"
25    ::= { dot20AnSectorExtChanInfoEntry 56 }
26
27 dot20AnErasureTargetCtoI2 OBJECT-TYPE
28     SYNTAX      Integer32 (0..15)
29     MAX-ACCESS  read-write
30     STATUS      current
31     DESCRIPTION
32        "This attribute's value (denoted n) determines the target C/I value of
33        erasure sequences for different assignment sizes. The
34        transmit power is equal to n-12 dB."
35    REFERENCE
36        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
37        Attribute)"
38    ::= { dot20AnSectorExtChanInfoEntry 57 }
39
40 dot20AnErasureTargetCtoI3 OBJECT-TYPE
41     SYNTAX      Integer32 (0..15)
42     MAX-ACCESS  read-write
43     STATUS      current
44     DESCRIPTION
45        "This attribute's value (denoted n) determines the target C/I value of
46        erasure sequences for different assignment sizes. The
47        transmit power is equal to n-12 dB."
48    REFERENCE
49        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
50        Attribute)"
51    ::= { dot20AnSectorExtChanInfoEntry 58 }
52
53 dot20AnAccessCycleDuration OBJECT-TYPE
54     SYNTAX      Integer32 (0..1)
55     MAX-ACCESS  read-write
56     STATUS      current
57     DESCRIPTION
58        "This attribute determines the duration of the access
59        cycle in units of Access Opportunities (as defined by the
60        Physical Layer.)"
61    REFERENCE
62        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
63        Group)"
64    ::= { dot20AnSectorExtChanInfoEntry 59 }
65
66 dot20AnMaxProbesPerSequence OBJECT-TYPE
67     SYNTAX      Integer32 (0..7)
68     MAX-ACCESS  read-write

```

```

1      STATUS      current
2      DESCRIPTION
3          "This attribute (denoted n) determines the maximum number of probe
4          sequences that can be part of one access sequence. The
5          number of probes is n+2"
6      REFERENCE
7          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
8          Group)"
9      ::= { dot20AnSectorExtChanInfoEntry 60 }
10
11     dot20AnProbeRampUpStepSize OBJECT-TYPE
12     SYNTAX      Integer32 (0..3)
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "This attribute's value (denoted n) determines the power ramp
17         up used for probes within a probe sequence and indicates
18         a ramp up value of 2*(1+n) dB."
19     REFERENCE
20         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
21         Group)"
22     ::= { dot20AnSectorExtChanInfoEntry 61 }
23
24     dot20AnPilotThreshold1 OBJECT-TYPE
25     SYNTAX      Integer32 (0..7)
26     MAX-ACCESS  read-write
27     STATUS      current
28     DESCRIPTION
29         "This attribute's value (denoted n) determines
30         PilotThreshold1 used by the Access Channel MAC Protocol. The
31         value is -10 + 2n dB."
32     REFERENCE
33         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
34         Group)"
35     ::= { dot20AnSectorExtChanInfoEntry 62 }
36
37     dot20AnPilotThreshold2 OBJECT-TYPE
38     SYNTAX      Integer32 (0..7)
39     MAX-ACCESS  read-write
40     STATUS      current
41     DESCRIPTION
42         "This attribute's value (denoted n) determines
43         PilotThreshold2 used by the Access Channel MAC Protocol. The
44         value is -2n dB."
45     REFERENCE
46         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
47         Group)"
48     ::= { dot20AnSectorExtChanInfoEntry 63 }
49
50     dot20AnOpenLoopAdjust OBJECT-TYPE
51     SYNTAX      Integer32 (0..255)
52     MAX-ACCESS  read-write
53     STATUS      current
54     DESCRIPTION
55         "This attribute's value (denoted n) determines the nominal
56         power to be used by access terminal in the open loop power
57         estimate. The value of nominal power is 70+n dB."
58     REFERENCE
59         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
60         Group)"
61     ::= { dot20AnSectorExtChanInfoEntry 64 }
62
63     dot20AnAccessRetryPersistence0 OBJECT-TYPE
64     SYNTAX      Integer32 (0..7)
65     MAX-ACCESS  read-write
66     STATUS      current
67     DESCRIPTION
68         "This attribute determines the persistence probability for

```

```

1         determining access sequence backoff. If this attribute's value
2         is set to n (n<7), the access terminal will use 2^(-n/2) as the
3         retry persistence. For n=7, the access terminal will set
4         AccessRetryPersistence0 to 0."
5     REFERENCE
6         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
7         Group)"
8     ::= { dot20AnSectorExtChanInfoEntry 65 }
9
10    dot20AnAccessRetryPersistence1 OBJECT-TYPE
11        SYNTAX      Integer32 (0..7)
12        MAX-ACCESS  read-write
13        STATUS      current
14        DESCRIPTION
15            "This attribute determines the persistence probability for
16            determining access sequence backoff. If this attribute's value
17            is set to n (n<7), the access terminal will use 2^(-n/2) as the
18            retry persistence. For n=7, the access terminal will set
19            AccessRetryPersistence1 to 0."
20        REFERENCE
21            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
22            Group)"
23        ::= { dot20AnSectorExtChanInfoEntry 66 }
24
25    dot20AnAccessRetryPersistence2 OBJECT-TYPE
26        SYNTAX      Integer32 (0..7)
27        MAX-ACCESS  read-write
28        STATUS      current
29        DESCRIPTION
30            "This attribute determines the persistence probability for
31            determining access sequence backoff. If this attribute's value
32            is set to n (n<7), the access terminal will use 2^(-n/2) as the
33            retry persistence. For n=7, the access terminal will set
34            AccessRetryPersistence2 to 0."
35        REFERENCE
36            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
37            Group)"
38        ::= { dot20AnSectorExtChanInfoEntry 67 }
39
40    dot20AnAccessRetryPersistence3 OBJECT-TYPE
41        SYNTAX      Integer32 (0..7)
42        MAX-ACCESS  read-write
43        STATUS      current
44        DESCRIPTION
45            "This attribute determines the persistence probability for
46            determining access sequence backoff. If this attribute's value
47            is set to n (n<7), the access terminal will use 2^(-n/2) as the
48            retry persistence. For n=7, the access terminal sets
49            AccessRetryPersistence3 to 0."
50        REFERENCE
51            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
52            Group)"
53        ::= { dot20AnSectorExtChanInfoEntry 68 }
54
55    dot20AnAccessRetryPersistence4 OBJECT-TYPE
56        SYNTAX      Integer32 (0..7)
57        MAX-ACCESS  read-write
58        STATUS      current
59        DESCRIPTION
60            "This attribute determines the persistence probability for
61            determining access sequence backoff. If this attribute's value
62            is set to n (n<7), the access terminal will use 2^(-n/2) as the
63            retry persistence. For n=7, the access terminal sets
64            AccessRetryPersistence4 to 0."
65        REFERENCE
66            "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
67            Group)"
68        ::= { dot20AnSectorExtChanInfoEntry 69 }

```

```

1
2 dot20AnAccessRetryPersistence5 OBJECT-TYPE
3     SYNTAX      Integer32 (0..7)
4     MAX-ACCESS  read-write
5     STATUS      current
6     DESCRIPTION
7         "This attribute determines the persistence probability for
8         determining access sequence backoff. If this attribute's value
9         is set to n (n<7), the access terminal will use 2^(-n/2) as the
10        retry persistence. For n=7, the access terminal sets
11        AccessRetryPersistence5 to 0."
12    REFERENCE
13        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
14        Group)"
15    ::= { dot20AnSectorExtChanInfoEntry 70 }
16
17 dot20AnAccessRetryPersistence6 OBJECT-TYPE
18     SYNTAX      Integer32 (0..7)
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22        "This attribute determines the persistence probability for
23        determining access sequence backoff. If this attribute's value
24        is set to n (n<7), the access terminal will use 2^(-n/2) as the
25        retry persistence. For n=7, the access terminal sets
26        AccessRetryPersistence6 to 0."
27    REFERENCE
28        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
29        Group)"
30    ::= { dot20AnSectorExtChanInfoEntry 71 }
31
32 dot20AnAccessRetryPersistence7 OBJECT-TYPE
33     SYNTAX      Integer32 (0..7)
34     MAX-ACCESS  read-write
35     STATUS      current
36     DESCRIPTION
37        "This attribute determines the persistence probability for
38        determining access sequence backoff. If this attribute's value
39        is set to n (n<0), the access terminal will use 2^(-n/2) as the
40        retry persistence. For n=7, the access terminal sets
41        AccessRetryPersistence7 to 0."
42    REFERENCE
43        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
44        Group)"
45    ::= { dot20AnSectorExtChanInfoEntry 72 }
46
47 dot20AnSectorExtChanRowStatus OBJECT-TYPE
48     SYNTAX      RowStatus
49     MAX-ACCESS  read-create
50     STATUS      current
51     DESCRIPTION
52        "The status column used for creating, modifying, and deleting
53        instances of the columnar objects in the SectorExtChanInfo
54        Table. If the implementer of this MIB has chosen not to
55        implement 'dynamic assignment' of sectors, this attribute is
56        not useful and should return noSuchName upon SNMP request."
57     DEFVAL     { active }
58     ::= { dot20AnSectorExtChanInfoEntry 73 }
59
60 dot20AnSectorParamTable OBJECT-TYPE
61     SYNTAX      SEQUENCE OF Dot20AnSectorParamEntry
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65        "This table provides one row per 802.20 carrier of a sector for
66        a specific ChannelBand. This table's attributes specify the
67        configuration of the corresponding sector and can be used to
68        populate fields in the SectorParameters message."

```



```

1      ::= { dot20AnOverheadMessages 3 }
2
3      dot20AnSectorParamEntry OBJECT-TYPE
4          SYNTAX          Dot20AnSectorParamEntry
5          MAX-ACCESS      not-accessible
6          STATUS          current
7          DESCRIPTION
8              "An Entry (conceptual row) in the SectorParam table. This table
9              is indexed by ifIndex. ifIndex: Each IEEE 802.20 interface
10             (uniquely identified by SectorID) is represented by an
11             ifEntry."
12         REFERENCE
13             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
14         INDEX
15             { ifIndex }
16         ::= { dot20AnSectorParamTable 1 }
17
18     dot20AnMobileCountryCode OBJECT-TYPE
19         SYNTAX          Integer32 (0..4096)
20         MAX-ACCESS      read-write
21         STATUS          current
22         DESCRIPTION
23             "This attribute is set to the three digit Mobile Country
24             Code associated with this sector (as specified in ITU-T
25             Recommendation E.212, Identification Plan for Land Mobile
26             Stations)."
27         REFERENCE
28             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
29         ::= { dot20AnSectorParamEntry 1 }
30
31     dot20AnMobileNetworkCode OBJECT-TYPE
32         SYNTAX          Integer32 (0..4096)
33         MAX-ACCESS      read-write
34         STATUS          current
35         DESCRIPTION
36             "This field is set three-digit BCD (binary coded
37             decimal) encoded representation of the Mobile Network Code
38             that has been assigned to the operator."
39         REFERENCE
40             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
41         ::= { dot20AnSectorParamEntry 2 }
42
43     dot20AnSectorID OBJECT-TYPE
44         SYNTAX          OCTET STRING (SIZE(16))
45         MAX-ACCESS      read-write
46         STATUS          current
47         DESCRIPTION
48             "Sector Address Identifier. The access network sets the
49             value of the SectorID according to the rules specified in 16.2.2.."
50         REFERENCE
51             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters) and
52             Subclause 16.2.2 (SectorID Construction)"
53         ::= { dot20AnSectorParamEntry 3 }
54
55     dot20AnChannelBandRef OBJECT-TYPE
56         SYNTAX          Integer32
57         MAX-ACCESS      read-write
58         STATUS          current
59         DESCRIPTION
60             "The reference to the ChannelBand defined in ChannelBands table
61             using this value as index (dot20AnChannelBandIndex)"
62         REFERENCE
63             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters,
64             first instance), and Subclause 15.2.1 (ChannelBand Record)"
65         ::= { dot20AnSectorParamEntry 4 }
66
67     dot20AnLatitude OBJECT-TYPE
68         SYNTAX          Integer32 (-1296000..1296000)

```

```

1     MAX-ACCESS    read-write
2     STATUS        current
3     DESCRIPTION
4         "The latitude of the sector. This attribute is set to
5         this sector's latitude in units of 0.25 second, expressed as a
6         two's complement signed number with positive numbers signifying
7         North latitudes. Similarly, negative numbers signify South latitudes.
8     This attribute is set to a value in the
9         range 1296000 to 1296000 inclusive (corresponding to a range of
10        -90 to +90)."
```

```

11    REFERENCE
12        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
13    ::= { dot20AnSectorParamEntry 5 }
14
15    dot20AnLongitude OBJECT-TYPE
16        SYNTAX      Integer32 (-2592000..2592000)
17        MAX-ACCESS  read-write
18        STATUS      current
19        DESCRIPTION
20            "The longitude of the sector. This attribute is set to
21            this sector's longitude in units of 0.25 second, expressed as a
22            two's complement signed number with positive numbers signifying
23            East longitude. Similarly, negative numbers signify West longitudes.
24    This attribute is set to a value in the
25        range 2592000 to 2592000 inclusive (corresponding to a range of
26        -180 degrees to +180 degrees)."
```

```

27    REFERENCE
28        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
29    ::= { dot20AnSectorParamEntry 6 }
30
31    dot20AnLeapSeconds OBJECT-TYPE
32        SYNTAX      Integer32 (0..255)
33        MAX-ACCESS  read-write
34        STATUS      current
35        DESCRIPTION
36            "The number of leap seconds that have occurred since the start
37            of system time."
38    REFERENCE
39        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
40    ::= { dot20AnSectorParamEntry 7 }
41
42    dot20AnLocalTimeOffset OBJECT-TYPE
43        SYNTAX      Integer32 (0..2047)
44        MAX-ACCESS  read-write
45        STATUS      current
46        DESCRIPTION
47            "This attribute is set to the offset of the local time
48            from System Time. This value is in units of minutes,
49            expressed as a two's complement signed number."
50    REFERENCE
51        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
52    ::= { dot20AnSectorParamEntry 8 }
53
54    dot20AnPrimaryRegZoneCode OBJECT-TYPE
55        SYNTAX      Integer32
56        MAX-ACCESS  read-write
57        STATUS      current
58        DESCRIPTION
59            "The PrimaryRegistrationZoneCode for this sector"
60    REFERENCE
61        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
62    ::= { dot20AnSectorParamEntry 9 }
63
64    dot20AnAnGroupId OBJECT-TYPE
65        SYNTAX      Integer32 (0..7)
66        MAX-ACCESS  read-write
67        STATUS      current
68        DESCRIPTION
```

```

1         "Sector's AN Group Id"
2     REFERENCE
3         "IEEE Std. 802.20-2008, Subclause 11.6.4.5.2.9 (AN Groups)"
4     ::= { dot20AnSectorParamEntry 10 }
5
6     dot20AnPilotGroupId OBJECT-TYPE
7         SYNTAX      Integer32 (0..7)
8         MAX-ACCESS  read-write
9         STATUS      current
10        DESCRIPTION
11            "Sector's Pilot Group Id"
12        REFERENCE
13            "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
14        ::= { dot20AnSectorParamEntry 11 }
15
16        dot20AnSynchronousGroupId OBJECT-TYPE
17            SYNTAX      Integer32 (0..7)
18            MAX-ACCESS  read-write
19            STATUS      current
20            DESCRIPTION
21                "Sector's Synchronous Group Id"
22            REFERENCE
23                "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
24            ::= { dot20AnSectorParamEntry 12 }
25
26        dot20AnCellGroupId OBJECT-TYPE
27            SYNTAX      Integer32 (0..7)
28            MAX-ACCESS  read-write
29            STATUS      current
30            DESCRIPTION
31                "Sector's Cell Group Id"
32            REFERENCE
33                "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
34            ::= { dot20AnSectorParamEntry 13 }
35
36        dot20AnSectorParamRowStatus OBJECT-TYPE
37            SYNTAX      RowStatus
38            MAX-ACCESS  read-create
39            STATUS      current
40            DESCRIPTION
41                "The status column used for creating, modifying, and deleting
42                instances of the columnar objects in the SectorParam Table. If
43                the implementer of this MIB has chosen not to implement
44                'dynamic assignment' of sectors, this attribute is not useful
45                and should return noSuchName upon SNMP request."
46            DEFVAL      { active }
47            ::= { dot20AnSectorParamEntry 14 }
48
49        dot20AnSectorGrpResSetsTable OBJECT-TYPE
50            SYNTAX      SEQUENCE OF Dot20AnSectorGrpResSetsEntry
51            MAX-ACCESS  not-accessible
52            STATUS      current
53            DESCRIPTION
54                "This table provides one row per 802.20 sector and Forward
55                Channel group resource set (see ExtendedChannelInfo)."
56            ::= { dot20AnOverheadMessages 4 }
57
58        dot20AnSectorGrpResSetsEntry OBJECT-TYPE
59            SYNTAX      Dot20AnSectorGrpResSetsEntry
60            MAX-ACCESS  not-accessible
61            STATUS      current
62            DESCRIPTION
63                "An Entry (conceptual row) in the
64                AnSectorFwdChanGrpResourceSets table. This table is indexed
65                by ifIndex and resourceSetId ifIndex: Each IEEE 802.20
66                interface (uniquely identified by SectorID) is represented by
67                an ifEntry."
68            REFERENCE

```

```

1         "IEEE Std. 802.20-2008, Subclause 11.7.5.3
2         (SupplementalConfigAssignment)"
3     INDEX
4     { ifIndex, dot20AnResourceSetId }
5     ::= { dot20AnSectorGrpResSetsTable 1 }
6
7     dot20AnResourceSetId OBJECT-TYPE
8     SYNTAX      Integer32 (0..7)
9     MAX-ACCESS  not-accessible
10    STATUS      current
11    DESCRIPTION
12        "Index of the forward channel group resource set for a
13        particular sector."
14    ::= { dot20AnSectorGrpResSetsEntry 1 }
15
16    dot20AnResourceSetBitmap OBJECT-TYPE
17    SYNTAX      Integer32 (0..255)
18    MAX-ACCESS  read-write
19    STATUS      current
20    DESCRIPTION
21        "The j'th bit of this field is set to 1 if a frame with
22        frame index mod InterlaceDepth = j contains a subzone that
23        corresponds to this resource set. If the InterlaceDepth = 6,
24        the last two bits of this field is set to 0"
25    REFERENCE
26        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
27        (SupplementalConfigAssignment)"
28    ::= { dot20AnSectorGrpResSetsEntry 2 }
29
30    dot20AnBRCHSubzoneCyclingEnabled OBJECT-TYPE
31    SYNTAX      TruthValue
32    MAX-ACCESS  read-write
33    STATUS      current
34    DESCRIPTION
35        "This parameter is set to TRUE if BRCHSubzoneCycling is enabled
36        on this sector. For BRCH resource set with BRCHSubzoneCycling
37        disabled or DRCH resource set, the first subzone offset on all
38        interlaces where this resource set is present is set to
39        the ResourceSubzoneOffset. For BRCH resource sets with
40        BRCHSubzoneCycling enabled, the offset of the first subzone
41        over each interlace is shifted cyclically. Since the offset of
42        first subzone over the lowest indexed interlace is defined by
43        ResourceSubzoneOffset, the offset of the first subzone in the
44        next interlace, where the resource set is present, is increased
45        by 1 mod NumBRCHSubzones"
46    REFERENCE
47        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
48        (SupplementalConfigAssignment)"
49    ::= { dot20AnSectorGrpResSetsEntry 3 }
50
51    dot20AnResourceSetSubZoneSpacing OBJECT-TYPE
52    SYNTAX      Integer32 (0..3)
53    MAX-ACCESS  read-write
54    STATUS      current
55    DESCRIPTION
56        "This field indicates the spacing between subzones in a
57        resource set. Subzones belonging to a resource group on an
58        interlace is equally spaced, where the first subzone is
59        defined by ResourceSubzoneOffset and
60        BRCHSubzoneCyclingEnabled"
61    REFERENCE
62        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
63        (SupplementalConfigAssignment)"
64    ::= { dot20AnSectorGrpResSetsEntry 4 }
65
66    dot20AnNumResourceSubzones OBJECT-TYPE
67    SYNTAX      Integer32 (0..31)
68    MAX-ACCESS  read-write

```

```

1      STATUS          current
2      DESCRIPTION
3          "This field determines the number of subzones in each
4          interlace where the resource set is present. An interlace is
5          defined as the set of frames that have the same Frame Index mod
6          InterlaceDepth, where InterlaceDepth is defined by
7          ResourceSetInterlace. This parameter takes the value n+1."
8      REFERENCE
9          "IEEE Std. 802.20-2008, Subclause 11.7.5.3
10         (SupplementalConfigAssignment)"
11     ::= { dot20AnSectorGrpResSetsEntry 5 }
12
13     dot20AnResourceSubzoneOffset OBJECT-TYPE
14         SYNTAX          Integer32 (0..31)
15         MAX-ACCESS      read-write
16         STATUS          current
17         DESCRIPTION
18             "This field is set to the first subzone on the lowest
19             indexed interlace that is part of a resource set. "
20         REFERENCE
21             "IEEE Std. 802.20-2008, Subclause 11.7.5.3
22             (SupplementalConfigAssignment)"
23         ::= { dot20AnSectorGrpResSetsEntry 6 }
24
25     dot20AnResourceSetRowStatus OBJECT-TYPE
26         SYNTAX          RowStatus
27         MAX-ACCESS      read-create
28         STATUS          current
29         DESCRIPTION
30             "The status column used for creating, modifying, and deleting
31             instances of the columnar objects in the
32             SectorFwdChanGrpResourceSet Table. If the implementor of this
33             MIB has chosen not to implement 'dynamic assignment' of
34             sectors, this attribute is not useful and should return
35             noSuchName upon SNMP request."
36         DEFVAL          { active }
37         ::= { dot20AnSectorGrpResSetsEntry 7 }
38
39     dot20AnSecondaryRegZoneCodeTable OBJECT-TYPE
40         SYNTAX          SEQUENCE OF Dot20AnSecondaryRegZoneCodeEntry
41         MAX-ACCESS      not-accessible
42         STATUS          current
43         DESCRIPTION
44             "This table provides one row per 802.20 interface and per
45             secondary registration zone code."
46         ::= { dot20AnOverheadMessages 5 }
47
48     dot20AnSecondaryRegZoneCodeEntry OBJECT-TYPE
49         SYNTAX          Dot20AnSecondaryRegZoneCodeEntry
50         MAX-ACCESS      not-accessible
51         STATUS          current
52         DESCRIPTION
53             "An Entry (conceptual row) in the SecondaryRegZoneCode table,
54             which is used to trigger registration for paging. This table
55             is indexed by IfIndex and dot20AnSecondaryRegZoneCodeIndex.
56             ifIndex: Each IEEE 802.20 interface (uniquely identified by
57             SectorID) is represented by an ifEntry."
58
59         INDEX
60             { ifIndex, dot20AnSecondaryRegZoneCodeIndex }
61         ::= { dot20AnSecondaryRegZoneCodeTable 1 }
62
63     dot20AnSecondaryRegZoneCodeIndex OBJECT-TYPE
64         SYNTAX          Integer32 (0..7)
65         MAX-ACCESS      not-accessible
66         STATUS          current
67         DESCRIPTION
68             "Index of the secondary registration zone code for a particular

```

```

1       sector."
2       ::= { dot20AnSecondaryRegZoneCodeEntry 1 }
3
4 dot20AnSecRegZoneCode OBJECT-TYPE
5     SYNTAX      Integer32 (0..255)
6     MAX-ACCESS  read-write
7     STATUS      current
8     DESCRIPTION
9       "One of the SecondaryRegistrationZoneCode for this sector"
10    REFERENCE
11     "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
12    ::= { dot20AnSecondaryRegZoneCodeEntry 2 }
13
14 dot20AnSecondaryRegZoneRowStatus OBJECT-TYPE
15     SYNTAX      RowStatus
16     MAX-ACCESS  read-create
17     STATUS      current
18     DESCRIPTION
19       "The status column used for creating, modifying, and deleting
20        instances of the columnar objects in the SecondaryRegZoneCode
21        Table. If the implementor of this MIB has chosen not to
22        implement 'dynamic assignment' of sectors, this attribute is
23        not useful and should return noSuchName upon SNMP request."
24     DEFVAL      { active }
25     ::= { dot20AnSecondaryRegZoneCodeEntry 3 }
26
27 dot20AnSectorIpsiTable OBJECT-TYPE
28     SYNTAX      SEQUENCE OF Dot20AnSectorIpsiEntry
29     MAX-ACCESS  not-accessible
30     STATUS      current
31     DESCRIPTION
32       "This table provides one row per 802.20 interface and per
33        IPSI."
34     ::= { dot20AnOverheadMessages 6 }
35
36 dot20AnSectorIpsiEntry OBJECT-TYPE
37     SYNTAX      Dot20AnSectorIpsiEntry
38     MAX-ACCESS  not-accessible
39     STATUS      current
40     DESCRIPTION
41       "An Entry (conceptual row) in the SectorIpsi table, which is a
42        list of personalities supported by the given sector. This table
43        is indexed by IfIndex and dot20AnIpsiIndex. ifIndex: Each IEEE
44        802.20 interface (uniquely identified by SectorID) is
45        represented by an ifEntry."
46     REFERENCE
47       "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
48     INDEX
49       { ifIndex }
50     ::= { dot20AnSectorIpsiTable 1 }
51
52 dot20AnIpsiIndex OBJECT-TYPE
53     SYNTAX      Integer32 (0..7)
54     MAX-ACCESS  not-accessible
55     STATUS      current
56     DESCRIPTION
57       "Index of an Ipsi supported by a particular sector."
58     ::= { dot20AnSectorIpsiEntry 1 }
59
60 dot20AnSupportedIpsi OBJECT-TYPE
61     SYNTAX      Integer32 (0..15)
62     MAX-ACCESS  read-write
63     STATUS      current
64     DESCRIPTION
65       "IPSI supported by a particular sector"
66     REFERENCE
67       "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
68     ::= { dot20AnSectorIpsiEntry 2 }

```

```

1
2 dot20AnIpsiRowStatus OBJECT-TYPE
3     SYNTAX      RowStatus
4     MAX-ACCESS  read-create
5     STATUS      current
6     DESCRIPTION
7         "The status column used for creating, modifying, and deleting
8         instances of the columnar objects in the SectorIpsi Table. If
9         the implementor of this MIB has chosen not to implement
10        'dynamic assignment' of sectors, this attribute is not useful
11        and should return noSuchName upon SNMP request."
12     DEFVAL      { active }
13     ::= { dot20AnSectorIpsiEntry 3 }
14
15 dot20AnSectorCdmaSubSegTable OBJECT-TYPE
16     SYNTAX      SEQUENCE OF Dot20AnSectorCdmaSubSegEntry
17     MAX-ACCESS  not-accessible
18     STATUS      current
19     DESCRIPTION
20         "This table provides one row per 802.20 sector, interlace and
21         Reverse Channel group CDMA Sub segment (see ExtendedChannelInfo
22         message in AIS)."

```

```

1
2 dot20AnChannelBandsTable OBJECT-TYPE
3     SYNTAX          SEQUENCE OF Dot20AnChannelBandsEntry
4     MAX-ACCESS      not-accessible
5     STATUS          current
6     DESCRIPTION
7         "This table provides one row per 802.20 ChannelBand. This
8         table's attributes specify the ChannelBand record of a
9         particular ChannelBand which may be used for a sector defined
10        in the SectorConfig table, or by a member of the neighbor list
11        defined in NeighborSectorsTable."
12    ::= { dot20AnOverheadMessages 9 }
13
14 dot20AnChannelBandsEntry OBJECT-TYPE
15     SYNTAX          Dot20AnChannelBandsEntry
16     MAX-ACCESS      not-accessible
17     STATUS          current
18     DESCRIPTION
19         "An Entry (conceptual row) in the ChannelBands table. The
20         Channel Bands table is referenced by the NeighborSectorsTable
21         or Sector Table. This table is indexed by ChannelBandIndex."
22
23     INDEX
24         { dot20AnChannelBandIndex }
25     ::= { dot20AnChannelBandsTable 1 }
26
27 dot20AnChannelBandIndex OBJECT-TYPE
28     SYNTAX          Integer32 (1..2147483647)
29     MAX-ACCESS      not-accessible
30     STATUS          current
31     DESCRIPTION
32         "Index of the ChannelBand within the ChannelBands table."
33     ::= { dot20AnChannelBandsEntry 1 }
34
35 dot20AnSystemType OBJECT-TYPE
36     SYNTAX          Integer32 (0..2)
37     MAX-ACCESS      read-write
38     STATUS          current
39     DESCRIPTION
40         "This attribute discriminates between the different ChannelBand
41         Records."
42     REFERENCE
43         "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
44     ::= { dot20AnChannelBandsEntry 2 }
45
46 dot20AnBandClass OBJECT-TYPE
47     SYNTAX          Integer32 (0..255)
48     MAX-ACCESS      read-write
49     STATUS          current
50     DESCRIPTION
51         "This attribute is set to the band class number
52         corresponding to the frequency assignment of the ChannelBand
53         specified by this record."
54     REFERENCE
55         "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
56     ::= { dot20AnChannelBandsEntry 3 }
57
58 dot20AnChannelNumber OBJECT-TYPE
59     SYNTAX          Integer32 (0..65535)
60     MAX-ACCESS      read-write
61     STATUS          current
62     DESCRIPTION
63         "This attribute is set to the Channel number
64         corresponding to the frequency assignment of the ChannelBand
65         specified by this record."
66     REFERENCE
67         "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
68     ::= { dot20AnChannelBandsEntry 4 }

```



```

1
2 dot20AnHalfDuplexSupported OBJECT-TYPE
3     SYNTAX      TruthValue
4     MAX-ACCESS  read-write
5     STATUS      current
6     DESCRIPTION
7         "This attribute is set to TRUE if half duplex operation
8         is supported in this system."
9     REFERENCE
10        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
11    ::= { dot20AnChannelBandsEntry 5 }
12 <Edited to here - 2008-11-07 1539>
13 dot20AnReverseChannelBandClass OBJECT-TYPE
14     SYNTAX      Integer32 (0..255)
15     MAX-ACCESS  read-write
16     STATUS      current
17     DESCRIPTION
18         "This attribute is set to the band class number
19         corresponding to the frequency assignment of the reverse
20         ChannelBand specified by this record."
21     REFERENCE
22         "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters),
23         and Subclause 15.2.1 (ChannelBand Record)"
24    ::= { dot20AnChannelBandsEntry 6 }
25
26 dot20AnReverseChannelNumber OBJECT-TYPE
27     SYNTAX      Integer32 (0..65535)
28     MAX-ACCESS  read-write
29     STATUS      current
30     DESCRIPTION
31         "This attribute is set to the Channel number
32         corresponding to the frequency assignment of the Reverse
33         ChannelBand specified by this record."
34     REFERENCE
35         "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters),
36         and Subclause 15.2.1 (ChannelBand Record)"
37    ::= { dot20AnChannelBandsEntry 7 }
38
39 dot20AnCyclicPrefixLength OBJECT-TYPE
40     SYNTAX      Integer32 (0..3)
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44         "This attribute is set to the cyclic prefix length,
45         i.e. it is set to the quantity (N_CP-1) from the Physical
46         Layer."
47     REFERENCE
48         "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record),
49         and Table 165 (Specification for the u Parameter)"
50    ::= { dot20AnChannelBandsEntry 8 }
51
52 dot20AnFFTSIZE OBJECT-TYPE
53     SYNTAX      Integer32 (0..7)
54     MAX-ACCESS  read-write
55     STATUS      current
56     DESCRIPTION
57         "This attribute is set to log2(N_FFT/128)."

```

```

1      REFERENCE
2      "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
3      ::= { dot20AnChannelBandsEntry 10 }
4
5      dot20AnChannelBandShortId OBJECT-TYPE
6      SYNTAX      Integer32 (0..3)
7      MAX-ACCESS  read-write
8      STATUS      current
9      DESCRIPTION
10     "This attribute identifies the two bit index that identifies
11     this channel in beacon transmissions"
12     REFERENCE
13     "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
14     ::= { dot20AnChannelBandsEntry 11 }
15
16     dot20AnChannelBandAccessHashMask OBJECT-TYPE
17     SYNTAX      Integer32 (0..65536)
18     MAX-ACCESS  read-write
19     STATUS      current
20     DESCRIPTION
21     "this attribute is set to the AccessHashingChannelMask for this
22     channel"
23     REFERENCE
24     "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
25     ::= { dot20AnChannelBandsEntry 12 }
26
27     dot20AnChannelBandStatus OBJECT-TYPE
28     SYNTAX      RowStatus
29     MAX-ACCESS  read-create
30     STATUS      current
31     DESCRIPTION
32     "The status column used for creating, modifying, and deleting
33     instances of the columnar objects in the ChannelBands Table.
34     If the implementor of this MIB has chosen not to implement
35     'dynamic assignment' of ChannelBands, this attribute is not
36     useful and should return noSuchName upon SNMP request."
37     REFERENCE
38     "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
39     DEFVAL      { active }
40     ::= { dot20AnChannelBandsEntry 13 }
41
42     dot20AnNeighborSectorsTable OBJECT-TYPE
43     SYNTAX      SEQUENCE OF Dot20AnNeighborSectorsEntry
44     MAX-ACCESS  not-accessible
45     STATUS      current
46     DESCRIPTION
47     "This table provides one row per 802.20 neighbor sector. This
48     table's attributes specify the sector parameters of a
49     particular neighbor sector which may be used as a neighbor to
50     one sector defined in the SectorConfig table."
51     ::= { dot20AnOverheadMessages 10 }
52
53     dot20AnNeighborSectorsEntry OBJECT-TYPE
54     SYNTAX      Dot20AnNeighborSectorsEntry
55     MAX-ACCESS  not-accessible
56     STATUS      current
57     DESCRIPTION
58     "An Entry (conceptual row) in the AnNeighborSectors table. This
59     table is indexed by ChannelBandIndex, NeighborSectorIndex."
60     INDEX
61     { dot20AnChannelBandIndex, dot20AnNeighborSectorIndex }
62     ::= { dot20AnNeighborSectorsTable 1 }
63
64     dot20AnNeighborSectorIndex OBJECT-TYPE
65     SYNTAX      Integer32 (1..2147483647)
66     MAX-ACCESS  not-accessible
67     STATUS      current
68     DESCRIPTION

```

```

1         "Index of the Neighbor Sector for this Neighbor Carrier within
2         the ChannelBand."
3         ::= { dot20AnNeighborSectorsEntry 1 }
4
5 dot20AnNeighborPilotID OBJECT-TYPE
6     SYNTAX      Integer32 (0..1023)
7     MAX-ACCESS  read-write
8     STATUS      current
9     DESCRIPTION
10        "This attribute is set to the PilotID of a neighboring
11        sector that the access terminal should add to its Neighbor
12        Set."
13     REFERENCE
14        "IEEE Std. 802.20-2008, Subclause 5.3.2.1 (PilotPN and PilotPhase)"
15     ::= { dot20AnNeighborSectorsEntry 2 }
16
17 dot20AnNeighborEffTransmitPower OBJECT-TYPE
18     SYNTAX      Integer32 (0..63)
19     MAX-ACCESS  read-write
20     STATUS      current
21     DESCRIPTION
22        "This attribute is set to the transmit power of the
23        sector in units of dBm."
24     REFERENCE
25        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
26     ::= { dot20AnNeighborSectorsEntry 3 }
27
28 dot20AnNeighborChannelBandRef OBJECT-TYPE
29     SYNTAX      Integer32
30     MAX-ACCESS  read-write
31     STATUS      current
32     DESCRIPTION
33        "The reference to the ChannelBand defined in ChannelBands table
34        (dot20AnChannelBandIndex)"
35     REFERENCE
36        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
37     ::= { dot20AnNeighborSectorsEntry 4 }
38
39 dot20AnNeighborChannelShortID OBJECT-TYPE
40     SYNTAX      Integer32 (0..3)
41     MAX-ACCESS  read-write
42     STATUS      current
43     DESCRIPTION
44        "Neighbor Sector's short Channel ID"
45     REFERENCE
46        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
47     ::= { dot20AnNeighborSectorsEntry 5 }
48
49 dot20AnNeighborSameANAsPrimSect OBJECT-TYPE
50     SYNTAX      TruthValue
51     MAX-ACCESS  read-write
52     STATUS      current
53     DESCRIPTION
54        "Set true if same access network as primary sector."
55     REFERENCE
56        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
57     ::= { dot20AnNeighborSectorsEntry 6 }
58
59 dot20AnNeighborSectorPilotGrpId OBJECT-TYPE
60     SYNTAX      Integer32 (0..7)
61     MAX-ACCESS  read-write
62     STATUS      current
63     DESCRIPTION
64        "Neighbor Sector's Pilot Group Id"
65     REFERENCE
66        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
67     ::= { dot20AnNeighborSectorsEntry 7 }
68

```

```

1 dot20AnNeighborSynchGroupId OBJECT-TYPE
2     SYNTAX      Integer32 (0..7)
3     MAX-ACCESS  read-write
4     STATUS      current
5     DESCRIPTION
6         "Neighbor Sector's Synchronous Group Id"
7     REFERENCE
8         "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
9     ::= { dot20AnNeighborSectorsEntry 8 }
10
11 dot20AnNeighborSectorCellGroupId OBJECT-TYPE
12     SYNTAX      Integer32 (0..7)
13     MAX-ACCESS  read-write
14     STATUS      current
15     DESCRIPTION
16         "Neighbor Sector's Cell Group Id"
17     REFERENCE
18         "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
19     ::= { dot20AnNeighborSectorsEntry 9 }
20
21 dot20AnNeighborSectorStatus OBJECT-TYPE
22     SYNTAX      RowStatus
23     MAX-ACCESS  read-create
24     STATUS      current
25     DESCRIPTION
26         "The status column used for creating, modifying, and deleting
27         instances of the columnar objects in the NeighborSectors
28         Table. If the implementor of this MIB has chosen not to
29         implement 'dynamic assignment' of neighbor sectors this
30         attribute is not useful and should return noSuchName upon SNMP
31         request."
32     REFERENCE
33         "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
34     DEFVAL      { active }
35     ::= { dot20AnNeighborSectorsEntry 10 }
36
37 dot20AnOtherTechNghbrsTable OBJECT-TYPE
38     SYNTAX      SEQUENCE OF Dot20AnOtherTechNghbrsEntry
39     MAX-ACCESS  not-accessible
40     STATUS      current
41     DESCRIPTION
42         "This table provides one row per other technology neighbor
43         channel. This table's attributes specify the technology type
44         and neighbor list of a particular neighbor channel which may be
45         used by one sector defined in the SectorConfig table for
46         inter-technology handoff."
47     ::= { dot20AnOverheadMessages 11 }
48
49 dot20AnOtherTechNghbrsEntry OBJECT-TYPE
50     SYNTAX      Dot20AnOtherTechNghbrsEntry
51     MAX-ACCESS  not-accessible
52     STATUS      current
53     DESCRIPTION
54         "An Entry (conceptual row) in the AnOtherTechNghbrs table. This
55         table is indexed by Sector (ifIndex) and OtherTechnologyIndex"
56     INDEX
57         { ifIndex, dot20AnOtherTechnologyIndex }
58     ::= { dot20AnOtherTechNghbrsTable 1 }
59
60 dot20AnOtherTechnologyIndex OBJECT-TYPE
61     SYNTAX      Integer32 (1..2147483647)
62     MAX-ACCESS  not-accessible
63     STATUS      current
64     DESCRIPTION
65         "The neighbor other technology entry index"
66     ::= { dot20AnOtherTechNghbrsEntry 1 }
67
68 dot20AnTechnologyType OBJECT-TYPE

```

```

1      SYNTAX      Integer32 (0..255)
2      MAX-ACCESS  read-write
3      STATUS      current
4      DESCRIPTION
5          "This attribute is set to the type of other technology.
6          Interpretation for its value should as defined in the AIS
7          spec."
8      REFERENCE
9          "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
10     ::= { dot20AnOtherTechNghbrsEntry 2 }
11
12     dot20AnTechNghbrListLength OBJECT-TYPE
13     SYNTAX      Integer32 (0..255)
14     MAX-ACCESS  read-write
15     STATUS      current
16     DESCRIPTION
17         "This attribute is set the length, in bytes, of the
18         neighbor list information for the other technology."
19     REFERENCE
20         "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
21     ::= { dot20AnOtherTechNghbrsEntry 3 }
22
23     dot20AnTechnologyNeighborList OBJECT-TYPE
24     SYNTAX      OCTET STRING (SIZE(256))
25     MAX-ACCESS  read-write
26     STATUS      current
27     DESCRIPTION
28         "This attribute is set to the neighbor list information
29         for the other technology."
30     REFERENCE
31         "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
32     ::= { dot20AnOtherTechNghbrsEntry 4 }
33
34     dot20AnOtherTechNghbrRowStatus OBJECT-TYPE
35     SYNTAX      RowStatus
36     MAX-ACCESS  read-create
37     STATUS      current
38     DESCRIPTION
39         "The status column used for creating, modifying, and deleting
40         instances of the columnar objects in the OtherTechNghbrs Table.
41         If the implementor of this MIB has chosen not to implement
42         'dynamic assignment' of other technology neighbors, this
43         attribute is not useful and should return noSuchName upon SNMP
44         request."
45     DEFVAL      { active }
46     ::= { dot20AnOtherTechNghbrsEntry 5 }
47
48     dot20AnNeighborListTable OBJECT-TYPE
49     SYNTAX      SEQUENCE OF Dot20AnNeighborListEntry
50     MAX-ACCESS  not-accessible
51     STATUS      current
52     DESCRIPTION
53         "This table defines the neighbor lists for the sectors defined
54         in the SectorConfig table. Each row in this table indexed per
55         sector (ifIndex) specifies a pointer to a neighbor sector of
56         this sector."
57     ::= { dot20AnOverheadMessages 12 }
58
59     dot20AnNeighborListEntry OBJECT-TYPE
60     SYNTAX      Dot20AnNeighborListEntry
61     MAX-ACCESS  not-accessible
62     STATUS      current
63     DESCRIPTION
64         "An Entry (conceptual row) in the AnNeighborList table. This
65         table is indexed by Sector (ifIndex) and NeighborIndex indexing
66         each neighbor sector for a particular Sector."
67     INDEX
68     { ifIndex, dot20AnNeighborIndex }

```

```

1      ::= { dot20AnNeighborListTable 1 }
2
3      dot20AnNeighborIndex OBJECT-TYPE
4          SYNTAX          Integer32 (1..32)
5          MAX-ACCESS      not-accessible
6          STATUS          current
7          DESCRIPTION
8              "This index identifies one neighbor sector for a Sector."
9          ::= { dot20AnNeighborListEntry 1 }
10
11     dot20AnNeighborSectorPointer OBJECT-TYPE
12         SYNTAX          RowPointer
13         MAX-ACCESS      read-create
14         STATUS          current
15         DESCRIPTION
16             "This attribute points to an instance of sector in SectorConfig
17             table or in NeighborSectors table. This sector is defined as a
18             neighbor of the sector identified by the ifIndex of this
19             attribute's entry."
20         ::= { dot20AnNeighborListEntry 2 }
21
22     dot20AnNeighborRowStatus OBJECT-TYPE
23         SYNTAX          RowStatus
24         MAX-ACCESS      read-create
25         STATUS          current
26         DESCRIPTION
27             "The status column used for creating, modifying, and deleting
28             instances of the columnar objects in the NeighborList Table.
29             If the implementor of this MIB has chosen not to implement
30             'dynamic assignment' of neighbor list entries this attribute is
31             not useful and should return noSuchName upon SNMP request."
32         DEFVAL          { active }
33         ::= { dot20AnNeighborListEntry 3 }
34
35     dot20AnSectorToIfIndexTable OBJECT-TYPE
36         SYNTAX          SEQUENCE OF Dot20AnSectorToIfIndexEntry
37         MAX-ACCESS      not-accessible
38         STATUS          current
39         DESCRIPTION
40             "This table can be used to find the ifIndex of an 802.20
41             interface based on its SectorID and ChannelBand information
42             (reverse mapping of the Sector Config table)."
43         ::= { dot20An 2 }
44
45     dot20AnSectorToIfIndexEntry OBJECT-TYPE
46         SYNTAX          Dot20AnSectorToIfIndexEntry
47         MAX-ACCESS      not-accessible
48         STATUS          current
49         DESCRIPTION
50             "An Entry (conceptual row) in the AnSectorToIfIndex table."
51         INDEX
52             { dot20AnSectorID, ifIndex }
53         ::= { dot20AnSectorToIfIndexTable 1 }
54
55     dot20AnIfChannelBandRef OBJECT-TYPE
56         SYNTAX          Integer32
57         MAX-ACCESS      read-write
58         STATUS          current
59         DESCRIPTION
60             "The reference to the ChannelBand defined in ChannelBands table
61             (dot20AnChannelBandIndex)"
62         REFERENCE
63             "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters,
64             first instance), and Subclause 15.2.1 (ChannelBand Record)"
65         ::= { dot20AnSectorToIfIndexEntry 1 }
66
67     dot20Cmn OBJECT-IDENTITY
68         STATUS          current

```

```

1      DESCRIPTION
2      "Common configuration and statistics."
3      ::= { ieee802dot20 2 }
4
5      dot20CmnMac OBJECT-IDENTITY
6      STATUS      current
7      DESCRIPTION
8      "MAC layer objects"
9      ::= { dot20Cmn 1 }
10
11     dot20CmnSessionControl OBJECT IDENTIFIER ::= { dot20CmnMac 1 }
12
13     dot20CmnSessionMgtProtocol OBJECT IDENTIFIER ::= { dot20CmnSessionControl 1 }
14
15     dot20CmnSessionOpenCounts OBJECT-TYPE
16     SYNTAX      Counter64
17     MAX-ACCESS  read-only
18     STATUS      current
19     DESCRIPTION
20     "Number of sessions opened"
21     REFERENCE
22     "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
23     Protocol State Diagram (Access Network))"
24     ::= { dot20CmnSessionMgtProtocol 1 }
25
26     dot20CmnSessionCloseCounts OBJECT-TYPE
27     SYNTAX      Counter64
28     MAX-ACCESS  read-only
29     STATUS      current
30     DESCRIPTION
31     "Number of sessions closed"
32     REFERENCE
33     "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
34     Protocol State Diagram (Access Network))"
35     ::= { dot20CmnSessionMgtProtocol 2 }
36
37     dot20CmnSessionFailureCounts OBJECT-TYPE
38     SYNTAX      Counter64
39     MAX-ACCESS  read-only
40     STATUS      current
41     DESCRIPTION
42     "Number of session open/close failures"
43     REFERENCE
44     "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
45     Protocol State Diagram (Access Network))"
46     ::= { dot20CmnSessionMgtProtocol 3 }
47
48     dot20CmnConnectionControl OBJECT IDENTIFIER ::= { dot20CmnMac 3 }
49
50     dot20CmnConnectedState OBJECT IDENTIFIER ::= { dot20CmnConnectionControl 1 }
51
52     dot20CmnActiveConnectionCounts OBJECT-TYPE
53     SYNTAX      Counter64
54     MAX-ACCESS  read-only
55     STATUS      current
56     DESCRIPTION
57     "Number of current active connections (in Open state.)"
58     REFERENCE
59     "IEEE Std 802.20-2008, Figures 152 and 153"
60     ::= { dot20CmnConnectedState 1 }
61
62     dot20CmnConnectionAttemptCounts OBJECT-TYPE
63     SYNTAX      Counter64
64     MAX-ACCESS  read-only
65     STATUS      current
66     DESCRIPTION
67     "Number of connection attempts (i.e. that reached BindATI state.)"
68     REFERENCE

```

```

1         "IEEE Std 802.20-2008, Figure 152 (Basic Connected State
2         Protocol State Diagram (AT)) and Figure 153 (Basic Connected
3         State Protocol State Diagram (AN))"
4         ::= { dot20CmnConnectedState 2 }
5
6 dot20CmnConnectionFailureCounts OBJECT-TYPE
7     SYNTAX      Counter64
8     MAX-ACCESS  read-only
9     STATUS      current
10    DESCRIPTION
11        "Number of connection failures during connection attempt (i.e.
12        That reached BindATI state without reaching Open state,
13        through timeout or deactivation"
14    REFERENCE
15        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
16        State Diagram (AT)) and 153 (Connected State Protocol State
17        Diagram (AN))"
18    ::= { dot20CmnConnectedState 3 }
19
20 dot20CmnConnectionDropCounts OBJECT-TYPE
21    SYNTAX      Counter64
22    MAX-ACCESS  read-only
23    STATUS      current
24    DESCRIPTION
25        "Number of dropped connections (via a command of
26        ConnectionState.Close) after a connection has been established."
27    REFERENCE
28        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
29        State Diagram (AT)) and 153 (Connected State Protocol State
30        Diagram (AN))"
31    ::= { dot20CmnConnectedState 4 }
32
33 dot20CmnConnectionReleaseCounts OBJECT-TYPE
34    SYNTAX      Counter64
35    MAX-ACCESS  read-only
36    STATUS      current
37    DESCRIPTION
38        "Number of connection release (Tx ConnectionClose or
39        Rx ConnectionClose) after a connection has been established."
40    REFERENCE
41        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
42        State Diagram (AT)) and 153 (Connected State Protocol State
43        Diagram (AN))"
44    ::= { dot20CmnConnectedState 5 }
45
46 dot20CmnRadioLink OBJECT IDENTIFIER ::= { dot20CmnMac 4 }
47
48 dot20CmnRlp OBJECT IDENTIFIER ::= { dot20CmnRadioLink 2 }
49
50 dot20CmnRlpStatsTable OBJECT-TYPE
51    SYNTAX      SEQUENCE OF Dot20CmnRlpStatsEntry
52    MAX-ACCESS  not-accessible
53    STATUS      current
54    DESCRIPTION
55        "This table provides one row of Radio Link Protocol statistics
56        per 802.20 interface"
57    ::= { dot20CmnRlp 1 }
58
59 dot20CmnRlpStatsEntry OBJECT-TYPE
60    SYNTAX      Dot20CmnRlpStatsEntry
61    MAX-ACCESS  not-accessible
62    STATUS      current
63    DESCRIPTION
64        "An Entry (conceptual row) in the RlpStats table. This table is
65        indexed by IfIndex and dot20StreamId."
66    INDEX
67        { ifIndex, dot20CmnStreamId }
68    ::= { dot20CmnRlpStatsTable 1 }

```



```

1
2 dot20CmnStreamId OBJECT-TYPE
3     SYNTAX      Integer32 (0 .. 31)
4     MAX-ACCESS  not-accessible
5     STATUS      current
6     DESCRIPTION
7         "Stream Id"
8     ::= { dot20CmnRlpStatsEntry 1 }
9
10 dot20CmnRlpTxBytes OBJECT-TYPE
11     SYNTAX      Counter64
12     MAX-ACCESS  read-only
13     STATUS      current
14     DESCRIPTION
15         "Number of RLP bytes of payload transmitted"
16     REFERENCE
17         "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
18         Procedures)"
19     ::= { dot20CmnRlpStatsEntry 2 }
20
21 dot20CmnRlpReTxBytes OBJECT-TYPE
22     SYNTAX      Counter64
23     MAX-ACCESS  read-only
24     STATUS      current
25     DESCRIPTION
26         "Number of RLP bytes of payload retransmitted"
27     REFERENCE
28         "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
29         Procedures)"
30     ::= { dot20CmnRlpStatsEntry 3 }
31
32 dot20CmnRlpTxDropBytes OBJECT-TYPE
33     SYNTAX      Counter64
34     MAX-ACCESS  read-only
35     STATUS      current
36     DESCRIPTION
37         "Number of RLP bytes of dropped before transmission"
38     REFERENCE
39         "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
40         Procedures)"
41     ::= { dot20CmnRlpStatsEntry 4 }
42
43 dot20CmnRlpTxStatus OBJECT-TYPE
44     SYNTAX      Counter64
45     MAX-ACCESS  read-only
46     STATUS      current
47     DESCRIPTION
48         "Number of RLP ReceiverStatus messages transmitted"
49     REFERENCE
50         "IEEE 802.20-2008, Subclause 7.3.4.3.3.5 (ATReceiverStatus),
51         and Subclause 7.3.4.3.3.7 (ANReceiverStatus)"
52     ::= { dot20CmnRlpStatsEntry 5 }
53
54 dot20CmnRlpRxBytes OBJECT-TYPE
55     SYNTAX      Counter64
56     MAX-ACCESS  read-only
57     STATUS      current
58     DESCRIPTION
59         "Number of RLP bytes of payload received"
60     ::= { dot20CmnRlpStatsEntry 6 }
61
62 dot20CmnRlpRxStatus OBJECT-TYPE
63     SYNTAX      Counter64
64     MAX-ACCESS  read-only
65     STATUS      current
66     DESCRIPTION
67         "Number of RLP ReceiverStatus messages received"
68     REFERENCE

```

```

1         "IEEE 802.20-2008, Subclause 7.3.4.3.3.5 (ATReceiverStatus),
2         and Subclause 7.3.4.3.3.7 (ANReceiverStatus)"
3     ::= { dot20CmnRlpStatsEntry 7 }
4
5 dot20CmnRlpTxPackets OBJECT-TYPE
6     SYNTAX      Counter64
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10        "Number of RLP Packets transmitted"
11     REFERENCE
12        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
13        Procedures)"
14     ::= { dot20CmnRlpStatsEntry 8 }
15
16 dot20CmnRlpReTxPackets OBJECT-TYPE
17     SYNTAX      Counter64
18     MAX-ACCESS  read-only
19     STATUS      current
20     DESCRIPTION
21        "Number of RLP Packets retransmitted"
22     REFERENCE
23        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
24        Procedures)"
25     ::= { dot20CmnRlpStatsEntry 9 }
26
27 dot20CmnRlpTxrDropPackets OBJECT-TYPE
28     SYNTAX      Counter64
29     MAX-ACCESS  read-only
30     STATUS      current
31     DESCRIPTION
32        "Number of RLP Packets dropped before transmission"
33     REFERENCE
34        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
35        Procedures)"
36     ::= { dot20CmnRlpStatsEntry 10 }
37
38 dot20CmnRlpRxPackets OBJECT-TYPE
39     SYNTAX      Counter64
40     MAX-ACCESS  read-only
41     STATUS      current
42     DESCRIPTION
43        "Number of RLP Packets received"
44     REFERENCE
45        "IEEE Std 802.20-2008, Subclause 7.3.3.4.3 (RLP Receive
46        Procedures)"
47     ::= { dot20CmnRlpStatsEntry 11 }
48
49 dot20CmnRlpTxNAKTimeouts OBJECT-TYPE
50     SYNTAX      Counter64
51     MAX-ACCESS  read-only
52     STATUS      current
53     DESCRIPTION
54        "Number of NAK Timeouts"
55     REFERENCE
56        "IEEE Std 802.20-2008, Subclause 7.3.3.4.3 (RLP Receive
57        Procedures)"
58     ::= { dot20CmnRlpStatsEntry 12 }
59
60 dot20CmnRlpTxACKTimeouts OBJECT-TYPE
61     SYNTAX      Counter64
62     MAX-ACCESS  read-only
63     STATUS      current
64     DESCRIPTION
65        "Number of ACK Timeouts"
66     REFERENCE
67        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
68        Procedures)"

```

```

1      ::= { dot20CmnRlpStatsEntry 13 }
2
3  dot20CmnQmp OBJECT-IDENTITY
4      STATUS          current
5      DESCRIPTION
6          "Qos Management Protocol"
7      ::= { dot20CmnRadioLink 3 }
8
9  dot20CmnQmpStatsTable OBJECT-TYPE
10     SYNTAX          SEQUENCE OF Dot20CmnQmpStatsEntry
11     MAX-ACCESS      not-accessible
12     STATUS          current
13     DESCRIPTION
14         "This table provides one row of QMP statistics per 802.20
15         interface"
16     ::= { dot20CmnQmp 2 }
17
18  dot20CmnQmpStatsEntry OBJECT-TYPE
19     SYNTAX          Dot20CmnQmpStatsEntry
20     MAX-ACCESS      not-accessible
21     STATUS          current
22     DESCRIPTION
23         "An Entry (conceptual row) in the QmpStats table. This table is
24         indexed by IfIndex. ifIndex: Each IEEE 802.20 interface is
25         represented by an ifEntry."
26     INDEX
27         { ifIndex }
28     ::= { dot20CmnQmpStatsTable 1 }
29
30  dot20CmnActiveReservationsCounts OBJECT-TYPE
31     SYNTAX          Counter64
32     MAX-ACCESS      read-only
33     STATUS          current
34     DESCRIPTION
35         "Number of Active (Open State) Reservations"
36     REFERENCE
37         "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
38         State Diagram (AT)), and Figure 22 (Forward Link Reservation State
39         Diagram (AN))"
40     ::= { dot20CmnQmpStatsEntry 1 }
41
42  dot20CmnIdleReservationsCounts OBJECT-TYPE
43     SYNTAX          Counter64
44     MAX-ACCESS      read-only
45     STATUS          current
46     DESCRIPTION
47         "Number of Idle (Close State) Reservations"
48     REFERENCE
49         "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
50         State Diagram (AT)), and Figure 22 (Forward Link Reservation State
51         Diagram (AN))"
52     ::= { dot20CmnQmpStatsEntry 2 }
53
54  dot20CmnReservationOpenCounts OBJECT-TYPE
55     SYNTAX          Counter64
56     MAX-ACCESS      read-only
57     STATUS          current
58     DESCRIPTION
59         "Number of Reservations Open requests"
60     REFERENCE
61         "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
62         State Diagram (AT)), Figure 22 (Forward Link Reservation State
63         Diagram (AN)), Subclause 7.2.3.3.1 (ReservationOnRequest), and
64         Subclause 7.2.3.3.6 (RevReservationOn)"
65     ::= { dot20CmnQmpStatsEntry 3 }
66
67  dot20CmnReservationCloseCounts OBJECT-TYPE
68     SYNTAX          Counter64

```

```

1     MAX-ACCESS      read-only
2     STATUS          current
3     DESCRIPTION
4         "Number of Reservations Close requests"
5     REFERENCE
6         "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
7         State Diagram (AT)), Figure 22 (Forward Link Reservation State
8         Diagram (AN), Subclause 7.2.3.3.2 (ReservationOffRequest),
9         and Subclause 7.2.3.3.7 (RevReservationOn)"
10    ::= { dot20CmnQmpStatsEntry 4 }
11
12    dot20CmnReservationFailCounts OBJECT-TYPE
13        SYNTAX          Counter64
14        MAX-ACCESS      read-only
15        STATUS          current
16        DESCRIPTION
17            "Number of Failed Reservations requests"
18        REFERENCE
19            "IEEE Std 802.20-2008, Subclause 7.2.3.3.5 (ReservationReject)"
20    ::= { dot20CmnQmpStatsEntry 5 }
21
22    dot20CmnSecurity OBJECT IDENTIFIER ::= { dot20CmnMac 5 }
23
24    dot20CmnKeyExchangeProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 1 }
25
26    dot20CmnKeyExchangeAttemptCounts OBJECT-TYPE
27        SYNTAX          Counter64
28        MAX-ACCESS      read-only
29        STATUS          current
30        DESCRIPTION
31            "Number of key exchanges attempts"
32        REFERENCE
33            "IEEE Std 802.20-2008, Subclause 10.4.5.2.1 (KeyRequest)"
34    ::= { dot20CmnKeyExchangeProtocol 1 }
35
36    dot20CmnKeyExchangeFailureCounts OBJECT-TYPE
37        SYNTAX          Counter64
38        MAX-ACCESS      read-only
39        STATUS          current
40        DESCRIPTION
41            "Number of key exchanges failures"
42        REFERENCE
43            "IEEE Std 802.20-2008, Subclause 10.4.5.2.4 (KeyReject)"
44    ::= { dot20CmnKeyExchangeProtocol 2 }
45
46    dot20CmnMessageIntegrityProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 2 }
47
48    dot20CmnAuthStatsTable OBJECT-TYPE
49        SYNTAX          SEQUENCE OF Dot20CmnAuthStatsEntry
50        MAX-ACCESS      not-accessible
51        STATUS          current
52        DESCRIPTION
53            "This table provides one row of Authentication statistics per
54            802.20 interface (i.e. sector for a specific ChannelBand.)"
55    ::= { dot20CmnMessageIntegrityProtocol 1 }
56
57    dot20CmnAuthStatsEntry OBJECT-TYPE
58        SYNTAX          Dot20CmnAuthStatsEntry
59        MAX-ACCESS      not-accessible
60        STATUS          current
61        DESCRIPTION
62            "Authentication statistics per 802.20 interfaces"
63        INDEX
64            { ifIndex }
65    ::= { dot20CmnAuthStatsTable 1 }
66
67    dot20CmnAuthFailureCounts OBJECT-TYPE
68        SYNTAX          Counter64

```

```

1     MAX-ACCESS      read-only
2     STATUS          current
3     DESCRIPTION
4         "Number of Authentication failures (i.e. failure code 0x03 for
5         RouteOpenReject.)"
6     REFERENCE
7         "IEEE Std 802.20-2008, Subclause 13.2.6.2.1
8         (RouteOpenRequest), and Subclause 13.2.6.12
9         (RouteOpenReject)"
10    ::= { dot20CmnAuthStatsEntry 1 }
11
12    dot20CmnAuthSuccessCounts OBJECT-TYPE
13        SYNTAX          Counter64
14        MAX-ACCESS      read-only
15        STATUS          current
16        DESCRIPTION
17            "Number of successful Authentications"
18        REFERENCE
19            "IEEE Std 802.20-2008, Subclause 13.2.6.2.1
20            (RouteOpenRequest), and Subclause 13.2.6.3
21            (RouteOpenAccept)"
22        ::= { dot20CmnAuthStatsEntry 2 }
23
24    dot20CmnLowerMAC OBJECT IDENTIFIER ::= { dot20CmnMac 6 }
25
26    dot20CmnLMACPacketStatsTable OBJECT-TYPE
27        SYNTAX          SEQUENCE OF Dot20CmnLMACPacketStatsEntry
28        MAX-ACCESS      not-accessible
29        STATUS          current
30        DESCRIPTION
31            "This table provides one row of Lower MAC protocol statistics
32            per 802.20 interface, packet format and nb of ARQ attempts
33            needed in order to successfully transmit/receive a packet."
34        ::= { dot20CmnLowerMAC 1 }
35
36    dot20CmnLMACPacketStatsEntry OBJECT-TYPE
37        SYNTAX          Dot20CmnLMACPacketStatsEntry
38        MAX-ACCESS      not-accessible
39        STATUS          current
40        DESCRIPTION
41            "An Entry (conceptual row) in the LMACPacketStats table. This
42            table is indexed by IfIndex, PacketFormatIndex and
43            ARQAttemptsIndex."
44        INDEX
45            { ifIndex, dot20CmnPacketFormatIndex, dot20CmnARQAttemptsIndex
46            }
47        ::= { dot20CmnLMACPacketStatsTable 1 }
48
49    dot20CmnPacketFormatIndex OBJECT-TYPE
50        SYNTAX          Integer32 (0..15)
51        MAX-ACCESS      not-accessible
52        STATUS          current
53        DESCRIPTION
54            "The packet format index as defined in 802.20 AIS spec."
55        ::= { dot20CmnLMACPacketStatsEntry 1 }
56
57    dot20CmnARQAttemptsIndex OBJECT-TYPE
58        SYNTAX          Integer32 (0..15)
59        MAX-ACCESS      not-accessible
60        STATUS          current
61        DESCRIPTION
62            "Number of ARQ attempts that were needed in order to transmit
63            or receive a packet. Index 0 means that the packets failed to
64            be transmitted/received."
65        ::= { dot20CmnLMACPacketStatsEntry 2 }
66
67
68    dot20CmnFwdTxPacketCounts OBJECT-TYPE

```

```

1      SYNTAX          Counter64
2      MAX-ACCESS      read-only
3      STATUS          current
4      DESCRIPTION
5          "Number of transmitted packets"
6      REFERENCE
7          "IEEE Std 802.20-2008, Subclause 8.6.5.5.2.2 (F-DCH TX Associated
8          with Persistent Assignments), Subclause 8.6.5.5.2.3 (F-DCH TX
9          Associated with Non-Persistent Assignments and Residual Resource
10         Assignments), and Subclause 8.6.5.5.2.4 (F-DCH TX Associated with
11         Group Resource Assignments)"
12     ::= { dot20CmnLMACPacketStatsEntry 3 }
13
14 dot20CmnRevRxPacketCounts OBJECT-TYPE
15     SYNTAX          Counter64
16     MAX-ACCESS      read-only
17     STATUS          current
18     DESCRIPTION
19         "Number of received packets"
20     REFERENCE
21         "IEEE Std 802.20-2008,
22         Subclause 8.6.5.5.1.2.2 (AT Processing for Non-Persistent
23         Assignments),
24         Subclause 8.6.5.5.1.2.3 (AT Processing for Residual Resource
25         Assignments),
26         Subclause 8.6.5.5.1.2.4 (AT Processing for Group Resource
27         Assignments)"
28     ::= { dot20CmnLMACPacketStatsEntry 4 }
29
30 dot20CmnLMACStatsTable OBJECT-TYPE
31     SYNTAX          SEQUENCE OF Dot20CmnLMACStatsEntry
32     MAX-ACCESS      not-accessible
33     STATUS          current
34     DESCRIPTION
35         "This table provides one row of Lower MAC protocol statistics
36         per 802.20 interface and packet formats."
37     ::= { dot20CmnLowerMAC 2 }
38
39 dot20CmnLMACStatsEntry OBJECT-TYPE
40     SYNTAX          Dot20CmnLMACStatsEntry
41     MAX-ACCESS      not-accessible
42     STATUS          current
43     DESCRIPTION
44         "An Entry (conceptual row) in the LMACStats table. This table
45         is indexed by IfIndex, PacketFormatIndex."
46     INDEX
47         { ifIndex, dot20CmnPacketFormatIndex }
48     ::= { dot20CmnLMACStatsTable 1 }
49
50 dot20CmnFLABCounts OBJECT-TYPE
51     SYNTAX          Counter64
52     MAX-ACCESS      read-only
53     STATUS          current
54     DESCRIPTION
55         "Number of Forward Link Assignment Blocks"
56     REFERENCE
57         "IEEE Std 802.20-2008, Table 44 (F-SCCH Blocks), and Subclause
58         8.5.5.4.1.2 (Framing of F-SCCH Blocks)"
59     ::= { dot20CmnLMACStatsEntry 1 }
60
61 dot20CmnRLABCounts OBJECT-TYPE
62     SYNTAX          Counter64
63     MAX-ACCESS      read-only
64     STATUS          current
65     DESCRIPTION
66         "Number of Reverse Link Assignment Block"
67     REFERENCE
68         "IEEE Std 802.20-2008, Table 44 (F-SCCH Blocks), and Subclause

```

```

1           8.5.5.4.1.2 (Framing of F-SCCH Blocks), and Subclause
2           8.5.5.3.1.1.3.3 (RLAB) "
3 ::= { dot20CmnLMACStatsEntry 2 }
4
5 dot20CmnAccessGrantCounts OBJECT-TYPE
6     SYNTAX      Counter64
7     MAX-ACCESS  read-only
8     STATUS      current
9     DESCRIPTION
10      "Number of Access Grants (the number of times the indication
11      ForwardLinkControlSegmentMAC.AccessGrantSent is raised)"
12     REFERENCE
13      "IEEE Std 802.20-2008, Subclause 8.5.5.4.1.1.3.1.1 (Procedures
14      for Sending an Access Grant)"
15     ::= { dot20CmnLMACStatsEntry 3 }
16
17 dot20Conformance OBJECT IDENTIFIER ::= { ieee802dot20 4 }
18
19 dot20Groups OBJECT IDENTIFIER ::= { dot20Conformance 1 }
20
21 dot20CmnSessionMgtPGroup OBJECT-GROUP
22     OBJECTS
23     { dot20CmnSessionCloseCounts, dot20CmnSessionFailureCounts,
24       dot20CmnSessionOpenCounts }
25     STATUS      current
26     DESCRIPTION
27     "The session management protocol statistics"
28     ::= { dot20Groups 1 }
29
30 dot20CmnKeyExchangePGroup OBJECT-GROUP
31     OBJECTS
32     { dot20CmnKeyExchangeAttemptCounts,
33       dot20CmnKeyExchangeFailureCounts }
34     STATUS      current
35     DESCRIPTION
36     "The key exchange protocol statistics"
37     ::= { dot20Groups 4 }
38
39 dot20CmnConnectedStatePGroup OBJECT-GROUP
40     OBJECTS
41     { dot20CmnActiveConnectionCounts,
42       dot20CmnConnectionAttemptCounts, dot20CmnConnectionDropCounts,
43       dot20CmnConnectionFailureCounts, dot20CmnConnectionReleaseCounts
44     }
45     STATUS      current
46     DESCRIPTION
47     "The connected state protocol statistics"
48     ::= { dot20Groups 5 }
49
50 dot20CmnRadioLinkGroup OBJECT-GROUP
51     OBJECTS
52     { dot20CmnActiveReservationsCounts,
53       dot20CmnIdleReservationsCounts, dot20CmnReservationCloseCounts,
54       dot20CmnReservationFailCounts, dot20CmnReservationOpenCounts,
55       dot20CmnRevRxPacketCounts, dot20CmnRlpReTxBytes,
56       dot20CmnRlpReTxPackets, dot20CmnRlpRxBytes,
57       dot20CmnRlpRxPackets, dot20CmnRlpRxStatus,
58       dot20CmnRlpTxACKTimeouts, dot20CmnRlpTxBytes,
59       dot20CmnRlpTxDropBytes, dot20CmnRlpTxNAKTimeouts,
60       dot20CmnRlpTxPackets, dot20CmnRlpTxStatus,
61       dot20CmnRlpTxrDropPackets }
62     STATUS      current
63     DESCRIPTION
64     "The radio link layer statistics"
65     ::= { dot20Groups 7 }
66
67 dot20CmnAuthGroup OBJECT-GROUP
68     OBJECTS

```

```

1      { dot20CmnAuthFailureCounts, dot20CmnAuthSuccessCounts }
2      STATUS          current
3      DESCRIPTION
4      "The authentication protocol statistics"
5      ::= { dot20Groups 8 }
6
7      dot20CmnLowerMACGroup OBJECT-GROUP
8      OBJECTS
9      { dot20CmnAccessGrantCounts, dot20CmnFLABCounts,
10     dot20CmnFwdTxPacketCounts, dot20CmnRLABCounts,
11     dot20CmnRevRxPacketCounts }
12     STATUS          current
13     DESCRIPTION
14     "The lower mac sublayer statistics"
15     ::= { dot20Groups 9 }
16
17     dot20AnIdleStatePGroup OBJECT-GROUP
18     OBJECTS
19     { dot20AnAccessAttemptCounts, dot20AnAccessAttemptFailCounts,
20     dot20AnPageAttemptCounts, dot20AnPageFailureCounts }
21     STATUS          current
22     DESCRIPTION
23     "The An idle state protocol statistics"
24     ::= { dot20Groups 10 }
25
26     dot20AnOverheadGroup OBJECT-GROUP
27     OBJECTS
28     { dot20An16QamScchT2PRatio, dot20AnAccessCycleDuration,
29     dot20AnAccessRetryPersistence0, dot20AnAccessRetryPersistence1,
30     dot20AnAccessRetryPersistence2, dot20AnAccessRetryPersistence3,
31     dot20AnAccessRetryPersistence4, dot20AnAccessRetryPersistence5,
32     dot20AnAccessRetryPersistence6, dot20AnAccessRetryPersistence7,
33     dot20AnAckInterferenceOffset, dot20AnAnGroupId,
34     dot20AnAssignmentAckHARQTx, dot20AnBRCHSubzoneCyclingEnabled,
35     dot20AnBandClass, dot20AnCBNumGuardSubcarriers,
36     dot20AnCDMAInterlacesBitmap, dot20AnCQIPilotTransmitPower,
37     dot20AnCdmaSubSegmentNum, dot20AnCellGroupId, dot20AnCellNullID,
38     dot20AnChannelBandAccessHashMask, dot20AnChannelBandRef,
39     dot20AnChannelBandShortId, dot20AnChannelNumber,
40     dot20AnCommonPilotTransmitPower, dot20AnCpichHoppingMode,
41     dot20AnCtrlAccessOffset, dot20AnCyclicPrefixLength,
42     dot20AnEffectiveTransmitPower, dot20AnEnableExpandedQPCH,
43     dot20AnErasureTargetCtoI0, dot20AnErasureTargetCtoI1,
44     dot20AnErasureTargetCtoI2, dot20AnErasureTargetCtoI3,
45     dot20AnFACKBandwidthFactor, dot20AnFFTSsize,
46     dot20AnFDPICHCodeOffsetSubtree0, dot20AnFDPICHCodeOffsetSubtree1,
47     dot20AnFDPICHCodeOffsetSubtree2, dot20AnFDPICHCodeOffsetSubtree3,
48     dot20AnFLReservedInterlaces, dot20AnFastIoTEnabled,
49     dot20AnFastOSIEnabled, dot20AnFlIotReportInterval,
50     dot20AnFlPcReportInterval, dot20AnFlPqiReportInterval,
51     dot20AnFlSdmaNumSubtrees, dot20AnFlSubzoneSize,
52     dot20AnHalfDuplexModeSupported, dot20AnHalfDuplexSupported,
53     dot20AnIfChannelBandRef, dot20AnLatitude, dot20AnLeapSeconds,
54     dot20AnLocalTimeOffset, dot20AnLongitude, dot20AnMacIdRange,
55     dot20AnMax16QamScchBlocks, dot20AnMaxNumLABs, dot20AnMaxNumSharedLABs,
56     dot20AnMaxProbesPerSequence, dot20AnMinScchResourceIndex,
57     dot20AnMobileCountryCode, dot20AnMobileNetworkCode,
58     dot20AnNeighborPilotID, dot20AnNeighborChannelShortID,
59     dot20AnNeighborSameANAsPrimSect, dot20AnNeighborSectorCellGroupId,
60     dot20AnNeighborSectorPilotGrpId, dot20AnNeighborChannelBandRef,
61     dot20AnNeighborSectorPointer, dot20AnNeighborSynchGroupId,
62     dot20AnNeighborEffTransmitPower, dot20AnNumAckableLABs,
63     dot20AnNumCmnPilotTxAnt, dot20AnNumCommonSegmentHopPorts,
64     dot20AnNumDRCHSubzones, dot20AnNumEffectiveAntennas,
65     dot20AnNumFLReservedSubzones,
66     dot20AnNumGuardSubcarriers, dot20AnNumLABSegments,
67     dot20AnNumOdcchReports, dot20AnNumRLCdmaSubsegments,
68     dot20AnNumResourceSubzones, dot20AnNumSilenceIntervalSubzone,

```



```

1      dot20AnOpenLoopAdjust, dot20AnOsiResponseMode,
2      dot20AnPdCabResSharingEnabled, dot20AnPilotGroupId, dot20AnPilotID,
3      dot20AnPilotThreshold1, dot20AnPilotThreshold2,
4      dot20AnPrimaryRegZoneCode, dot20AnProbeRampUpStepSize,
5      dot20AnRabEnabled, dot20AnRackBandwidthFactor,
6      dot20AnReqQoSPowerBoost, dot20AnResourceChannelMuxMode,
7      dot20AnResourceSetBitmap, dot20AnResourceSetSubZoneSpacing,
8      dot20AnResourceSubzoneOffset, dot20AnReverseChannelBandClass,
9      dot20AnReverseChannelNumber, dot20AnRlAuxPilotPower,
10     dot20AnRlDpichCodeOffsetSubtree0, dot20AnRlDpichCodeOffsetSubtree1,
11     dot20AnRlDpichCodeOffsetSubtree2, dot20AnRlDpichCodeOffsetSubtree3,
12     dot20AnRlNumSdmaDimensions, dot20AnModSymbolsPerQPSKLAB,
13     dot20AnSFNCellID, dot20AnSecRegZoneCode, dot20AnSectorID,
14     dot20AnSilenceIntervalDuration, dot20AnSilenceIntervalPeriod,
15     dot20AnSinglePAForXCarriers, dot20AnSlowInterferenceOffset,
16     dot20AnSupportedIpsi, dot20AnSynchronousGroupId, dot20AnSystemType,
17     dot20AnTechNghbrListLength, dot20AnTechnologyNeighborList,
18     dot20AnTechnologyType, dot20AnTotalNumSubcarriers,
19     dot20AnUseDrchForFlcs, dot20AnRlSubzoneSize }
20     STATUS          current
21     DESCRIPTION "The overhead messages protocol configuration"
22     ::= { dot20Groups 11 }
23
24     dot20AnOverheadGroup2 OBJECT-GROUP
25     OBJECTS
26         { dot20AnChannelBandStatus, dot20AnIpsiRowStatus,
27           dot20AnNeighborRowStatus, dot20AnNeighborSectorStatus,
28           dot20AnOtherTechNghbrRowStatus, dot20AnResourceSetRowStatus,
29           dot20AnSecondaryRegZoneRowStatus,
30           dot20AnSectorCdmaSubSegRowStatus, dot20AnSectorConfigRowStatus,
31           dot20AnSectorExtChanRowStatus, dot20AnSectorParamRowStatus }
32     STATUS          current
33     DESCRIPTION
34         "If the MIB is created with pre-configured sector list tables and
35         neighbor list tables, this Overhead Group is unnecessary. Otherwise,
36         these items are used to add rows to these tables in the MIB, so
37         that additional sectors and/or neighbors can be added after MIB
38         creation, through SNMPv2."
39     ::= { dot20Groups 12 }
40
41     dot20Compliances OBJECT IDENTIFIER ::= { dot20Conformance 2 }
42
43     dot20AnCompliance MODULE-COMPLIANCE
44     STATUS          current
45     DESCRIPTION
46         "The compliance statement for SNMPv2 entities that implement
47         the IEEE 802.20 MIB for the An."
48     MODULE          IEEE802dot20-MIB
49     MANDATORY-GROUPS
50         { dot20AnIdleStatePGroup, dot20AnOverheadGroup,
51           dot20CmnAuthGroup, dot20CmnConnectedStatePGroup,
52           dot20CmnKeyExchangePGroup, dot20CmnLowerMACGroup,
53           dot20CmnRadioLinkGroup, dot20CmnSessionMgtPGroup }
54     GROUP          dot20AnOverheadGroup2
55     DESCRIPTION
56         "This group is required only if 'dynamic assignment' of
57         rows in the OverheadGroup tables is supported."
58     ::= { dot20Compliances 1 }
59
60     END
61

```


1 --Replace Chapter 31 of 802.20-2008 with the following text

2 **31 625k-MC OA & M Radio Network Quality Monitor and Control** 3 **Enhancement**

4 *This Chapter is an added to the baseline specification HC-SDMA.*

5 625k-MC network systems provide radio network quality monitoring and control functionality. The
6 MIB of 625k-MC mode comprises of the managed objects, attributes, actions, and notifications
7 required to manage a BS. The definition of these managed objects, attributes, actions, and
8 notifications, as well as their structure, is presented below.

9 **31.1 625k-MC Mode MIB**

10 **31.1.1 Overview**

11 This chapter defines a Management Information Base (MIB) module for managing the 625k-MC
12 mode. Managed objects are accessed via a virtual information store, termed the Management
13 Information Base or MIB. MIB objects are generally accessed through the Simple Network
14 Management Protocol (SNMP). The objects in this MIB are defined using the mechanisms specified
15 in the Structure of Management Information (SMI). The MIB module specified is compliant to
16 SMIV2 which is described in RFC 2578, RFC 2579, and RFC 2580.

17 **31.1.2 625k-MC MIB Structure**

18 The 625k-MC MIB is implemented as a single flat structure.

19 **31.1.3 Security Considerations**

20 This MIB relates to a system which provides mobile broadband wireless access. As such, improper
21 manipulation of the objects represented by this MIB can result in denial of service to a large number
22 of end-users.

23 The MIB objects in the Dot20AnChannelBandsEntry SEQUENCE contain 8 objects used to set the
24 frequency band of the transmitting base station. An administrator should take great care to include
25 only authorized, licensed channel bands in the table. Failure to take these measures might cause a
26 base station to violate local regulatory laws (e.g. FCC licensing in the USA) by transmitting power
27 into unauthorized channels in the country where the base station is deployed.

28 There are no MIB objects that could allow a user to increase their access rights to system service
29 levels. None of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other
30 than not-accessible) can be considered capable of revealing sensitive or vulnerable personal
31 information. This MIB is not capable of revealing user information that could violate privacy laws.

32 There are no MIB objects that could be used to turn off or change the security parameter
33 configuration of an 802.20 access node. The presence or absence of security (encryption,
34 authentication) is controlled by the session state record for each individual user, and cannot be
35 modified by an attacker accessing the MIB.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is recommended that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is not recommended. Instead, it is recommended to deploy SNMPv3 and to enable cryptographic security. It is a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those that have legitimate rights to indeed GET or SET (change/create/delete) them.

31.1.4 IANA Considerations

No IANA actions are required by this document.

31.1.5 Definition

```

IEEE802dot20-625k-MC-MIB DEFINITIONS ::= BEGIN
IMPORTS
    ifIndex
        FROM IF-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    Counter32, Counter64, Integer32, Unsigned32, MODULE-IDENTITY,
    OBJECT-IDENTITY, OBJECT-TYPE, transmission
        FROM SNMPv2-SMI
    TruthValue, Gauge, Counter, Cauge32          FROM SNMPv2-TC
;

IEEE802dot20-625k-MC-MIB MODULE-IDENTITY
    LAST-UPDATED      "2008110072000Z" - Norvember 08, 2008
    ORGANIZATION      "IEEE 802.20"
    CONTACT-INFO      "Contact: R. Canchi
Postal: 472 Kato Terrace, FREMONT, CA 94539, USA
Tel: 510-257-0132
Fax: 510 257 0131
E-mail: cradhak@ktrc-na.com"

    DESCRIPTION
        "The MIB module for IEEE802.20 625k-MC mode entities"
    ::= { transmission 9999 }

    _625k-MCSystem          OBJECT IDENTIFIER      -- DESCRIPTION
    "System Elements"
    ::= { IEEE802dot20-625k-MC-MIB 1 }

    _625k-MCSysAlarms      OBJECT IDENTIFIER      -- DESCRIPTION
    "Alarms"
    ::= { _625k-MCSystem 1 }

```

```

1
2
3  _625k-MCAAlarmScalars                                OBJECT IDENTIFIER
4  -- DESCRIPTION      "Alarm Scalars"
5  ::= { _625k-MCSysAlarms 1 }
6
7
8
9  _625k-MCCommonAlarmStatus                            OBJECT-TYPE
10 SYNTAX              Unsigned32 -- Unsigned32Type
11 MAX-ACCESS          read-only
12 STATUS              Current
13 DESCRIPTION
14   "Common alarm atatus.
15
16
17   (From mibCtl ElementType 16 CommonAlarmStatus)
18   Description for mibCtl Type 14 Unsigned32Type :
19     32 bit unsigned integer.
20   Type derived from mibCtl Type 11 Word32Type :
21     32 bits of raw opaque data.
22   Derived from basic 32 bit word type.
23   "
24   ::= { _625k-MCAAlarmScalars 1 }
25
26
27
28
29  _625k-MCFailReasonForAlarm                          OBJECT-TYPE
30 SYNTAX              Unsigned32 -- Unsigned32Type
31 MAX-ACCESS          read-only
32 STATUS              Current
33 DESCRIPTION
34   "Fail reason for alarm.
35
36
37   (From mibCtl ElementType 15 FailReasonForAlarm)
38   Description for mibCtl Type 14 Unsigned32Type :
39     32 bit unsigned integer.
40   Type derived from mibCtl Type 11 Word32Type :
41     32 bits of raw opaque data.
42   Derived from basic 32 bit word type.
43   "
44   ::= { _625k-MCAAlarmScalars 2 }
45
46
47
48
49
50  _625k-MCAAlarmSummaryTable                          OBJECT-TYPE
51 SYNTAX SEQUENCE OF _625k-MCAAlarmSummaryTableEntry
52 MAX-ACCESS          not-accessible
53 STATUS              Current
54 DESCRIPTION        "Alarm Summary Table"
55 ::= { _625k-MCSysAlarms 2 }
56
57
58
59  _625k-MCAAlarmSummaryTableEntry                    OBJECT-TYPE
60 SYNTAX              _625k-MCAAlarmSummaryTableEntry
61 MAX-ACCESS          not-accessible
62 STATUS              Current
63 DESCRIPTION        ""
64 INDEX { _625k-MCAAlarmSummaryTableIndex }
65 ::= { _625k-MCAAlarmSummaryTable 1 }
66
67  _625k-MCAAlarmSummaryTableEntry ::= SEQUENCE {
68  _625k-MCAAlarmSummaryTableIndex                    INTEGER, -- AlarmEventType

```

```

1  _625k-MCAAlarmSummary          INTEGER -- AlarmStateType
2  }
3
4
5
6  _625k-MCAAlarmSummaryTableIndex OBJECT-TYPE
7  SYNTAX          INTEGER -- AlarmEventType
8  MAX-ACCESS      read-only
9  STATUS          Current
10 DESCRIPTION    "
11     Description for mibCtl Type 85 AlarmEventType :
12     Enumeration of alarm event types.
13
14     Defines semantics of events that are also alarms.
15     All alarm events are enumerated first in the list of event types.
16     The highest alarm event index will never be more than 255.
17     [Limits: 0 255 ]
18     Type derived from mibCtl Type 3 EventType :
19     Enumeration of event types.
20
21     Defines semantics of events.
22     An event is re. an event log message.
23     [Limits: 0 255 ]
24     Type derived from mibCtl Type 14 Unsigned32Type :
25     32 bit unsigned integer.
26     Type derived from mibCtl Type 11 Word32Type :
27     32 bits of raw opaque data.
28     Derived from basic 32 bit word type.
29     "
30 ::= { _625k-MCAAlarmSummaryTableEntry 1 }
31
32
33
34 _625k-MCAAlarmSummary          OBJECT-TYPE
35 SYNTAX          INTEGER -- AlarmStateType
36 MAX-ACCESS      read-only
37 STATUS          Current
38 DESCRIPTION    "Summary of all alarms generated by the base station.
39
40     Each element contains the summary of a type of alarm (e.g. module
41     over temperature).  When the management station sees that alarm
42     summary is SET, it can, for example, query AlarmModuleOverTemp
43     to see which module(s) is over temperature.
44
45     (From mibCtl ElementType 5210 AlarmSummary)
46     Description for mibCtl Type 80 AlarmStateType :
47     Current state of an alarm.
48
49     This value is CLEARED when
50     the conditions which caused the alarm to occur are taken care of
51     and no longer exist.
52     The value is SET when due to some conditions, the Base Station
53     software decides that an alarm is necessary.
54     Typically (though this may not be true for all alarms
55     or if the alarm changes state too frequently)
56     an event is logged when an alarm is SET and then again when it
57     is CLEARED.
58     [Limits: 0 1 ]
59     Description for mibCtl AlarmStateType 0 CLEARED :
60     No alarm.
61     Description for mibCtl AlarmStateType 1 SET :
62     Alarm is set.
63     "
64 ::= { _625k-MCAAlarmSummaryTableEntry 2 }
65
66
67
68

```

```

1  _625k-MCSysFiles                OBJECT IDENTIFIER
2      -- DESCRIPTION                "Files"
3      ::= { _625k-MCSystem 2 }
4
5
6
7  _625k-MCStatsFiles                OBJECT IDENTIFIER
8      -- DESCRIPTION                "Statistics file"
9      ::= { _625k-MCSysFiles 1 }
10
11
12
13 _625k-MCStatsUploadURL            OBJECT-TYPE
14     SYNTAX                        OCTET STRING (SIZE(0..64)) -- URLType
15     MAX-ACCESS                    read-write
16     STATUS                        Current
17     DESCRIPTION
18         "EMS location to upload BS statistics file.
19
20
21
22         (From mibCtl ElementType 2831 StatsUploadURL)
23         Description for mibCtl Type 401 URLType :
24         Universal Resource Locator (URL).
25
26         A Universal Resource Locator (URL) is a text string
27         that specifies a network location for a file.
28         The general format for a URL consists of 2 parts:
29
30         1. Protocol name: lower case letters, followed by a colon.
31         See below for supported protocols.
32         This field may be omitted, to default to the file: protocol.
33
34         2. Additional information, depending on the protocol.
35         For many protocols, a host name is required,
36         which consists of a dotted numerical Internet Protocol (IP)
37         address,
38         or a dotted symbolic name with alphanumerical components,
39         where supported.
40
41         Supported protocols are:
42
43         tftp: is the Trivial File Transfer Protocol.
44         The additional information should begin with two slashes (//)
45         followed by a host name, a slash (/) and a file path.
46         The file path is interpreted by the host system,
47         frequently relative to a special directory set up for this
48         purpose.
49
50         file: is the plain old file protocol.
51         The additional information consists of a file path, which
52         should begin with a slash (/).
53         This is only useful if Base Station has been configured
54         to be an Network File System (NFS) client of the host.
55         The filepath is interpreted on the Base Station, so it
56         must begin with the mount name specified in the NFS configuration.
57         Type derived from mibCtl Type 15 TextType :
58         ASCII or compatible text.
59         Type derived from mibCtl Type 12 OctetType :
60         8 bits of raw opaque data.
61         Derived from basic 8 bit word type.
62         "
63     ::= { _625k-MCStatsFiles 1 }
64
65
66
67 _625k-MCStatsUploadStatus          OBJECT-TYPE
68     SYNTAX                        INTEGER -- FileUploadStatusType

```

```

1     MAX-ACCESS          read-only
2     STATUS              Current
3     DESCRIPTION
4         "Stats file upload status.
5
6
7
8         (From mibCtl ElementType 2832 StatsUploadStatus)
9     Description for mibCtl Type 403 FileUploadStatusType :
10        File upload status.
11
12
13    Description for mibCtl FileUploadStatusType 0 Unknown :
14        File upload status is unknown.
15    Description for mibCtl FileUploadStatusType 1 Missing :
16        File is missing or invalid.
17    Description for mibCtl FileUploadStatusType 2 PartialUpload :
18        File is in the process of being upload to EMS.
19    Description for mibCtl FileUploadStatusType 4 Complete :
20        File is completely uploaded to EMS.
21    Description for mibCtl FileUploadStatusType 5 Failure :
22        Upload process is failure.
23    Description for mibCtl FileUploadStatusType 6 NotManaged :
24        File upload is not being managed.
25    "
26    ::= { _625k-MCStatsFiles 2 }
27
28
29
30    _625k-MCStatsUploadFailReason          OBJECT-TYPE
31    SYNTAX          INTEGER -- FileUploadFailReasonType
32    MAX-ACCESS      read-only
33    STATUS          Current
34    DESCRIPTION
35        "Reason for last stats file upload failure.
36
37
38
39        (From mibCtl ElementType 2833 StatsUploadFailReason)
40    Description for mibCtl Type 406 FileUploadFailReasonType :
41        Reason for failure to upload a file..
42
43
44    Description for mibCtl FileUploadFailReasonType 0 NoFailure :
45        File upload in progress or completed without problem.
46    Description for mibCtl FileUploadFailReasonType 1 BadPathSpecified :
47        File upload failed because network path not found.
48    Description for mibCtl FileUploadFailReasonType 2 FlashDiskReadError :
49        File upload failed because of flash disk read error.
50    Description for mibCtl FileUploadFailReasonType 3 Aborted :
51        File upload aborted due to change of specification.
52    Description for mibCtl FileUploadFailReasonType 4 WriteError :
53        Error in putting a file.
54    "
55    ::= { _625k-MCStatsFiles 3 }
56
57
58
59    _625k-MCStatsUploadBytes          OBJECT-TYPE
60    SYNTAX          Unsigned32 -- Unsigned32Type
61    MAX-ACCESS      read-only
62    STATUS          Current
63    DESCRIPTION
64        "Upload size of BS stats file in bytes.
65
66
67
68        (From mibCtl ElementType 2834 StatsUploadBytes)

```



```

1      Description for mibCtl Type 14 Unsigned32Type :
2          32 bit unsigned integer.
3      Type derived from mibCtl Type 11 Word32Type :
4          32 bits of raw opaque data.
5      Derived from basic 32 bit word type.
6      "
7      ::= { _625k-MCStatsFiles 4 }
8
9
10
11     _625k-MCStatsUploadDate          OBJECT-TYPE
12     SYNTAX          Gauge32 -- AbsoluteTimeType
13     MAX-ACCESS      read-only
14     STATUS          Current
15     DESCRIPTION
16         "BS Stats File upload complete time.
17
18
19
20     (From mibCtl ElementType 2835 StatsUploadDate)
21     Description for mibCtl Type 801 AbsoluteTimeType :
22         Absolute time in GPS seconds.
23
24         GPS (Global Positioning System) time in seconds since Jan. 6,
25     1980.
26
27         Note that this differs from UTC (in addition to a possible
28         offset due to starting time) due to leap seconds; see
29         the GpsLeapSecond element.
30     Type derived from mibCtl Type 18 Gauge32Type :
31         32 bits of Gauge data.
32     Derived from basic 32 bit word type.
33     "
34     ::= { _625k-MCStatsFiles 5 }
35
36
37     _625k-MCUploadStatsFile          OBJECT-TYPE
38     SYNTAX          Unsigned32 -- Unsigned32Type
39     MAX-ACCESS      read-write -- REALLY: write-only
40     STATUS          Current
41     DESCRIPTION
42         "Upload Stats file.
43
44
45
46     (From mibCtl ElementType 2836 UploadStatsFile)
47     Description for mibCtl Type 14 Unsigned32Type :
48         32 bit unsigned integer.
49     Type derived from mibCtl Type 11 Word32Type :
50         32 bits of raw opaque data.
51     Derived from basic 32 bit word type.
52     "
53     ::= { _625k-MCStatsFiles 6 }
54
55
56     _625k-MCSysInterfaces            OBJECT IDENTIFIER
57     -- DESCRIPTION          "System Interfaces"
58     ::= { _625k-MCSystem 3 }
59
60
61
62     _625k-MCInterfaceNetwork        OBJECT IDENTIFIER
63     -- DESCRIPTION          "Network Interfaces"
64     ::= { _625k-MCSysInterfaces 1 }
65
66
67
68     _625k-MCTypeOfNetworkProtocol    OBJECT-TYPE

```

```

1      SYNTAX          INTEGER -- NetworkProtocolType
2      MAX-ACCESS      read-only
3      STATUS          Current
4      DESCRIPTION
5          "Type of Network Protocol used with the Network.
6
7          Type of Network Protocol is Ethernet or ATM.
8
9          (From mibCtl ElementType 3002 TypeOfNetworkProtocol)
10         Description for mibCtl Type 214 NetworkProtocolType :
11             Network Protocol type.
12         Description for mibCtl NetworkProtocolType 0 Unknown :
13             Network protocol type is unknown.
14         Description for mibCtl NetworkProtocolType 1 Ethernet :
15             Ethernet interface.
16         Description for mibCtl NetworkProtocolType 2 ATM :
17             ATM interface.
18         "
19         ::= { _625k-MCInterfaceNetwork 1 }
20
21
22
23
24     _625k-MCMgmtNetConfigTable          OBJECT-TYPE
25     SYNTAX SEQUENCE OF _625k-MCMgmtNetConfigTableEntry
26     MAX-ACCESS      not-accessible
27     STATUS          Current
28     DESCRIPTION    "Mgmt Network Configuration"
29     ::= { _625k-MCInterfaceNetwork 2 }
30
31
32
33     _625k-MCMgmtNetConfigTableEntry      OBJECT-TYPE
34     SYNTAX          _625k-MCMgmtNetConfigTableEntry
35     MAX-ACCESS      not-accessible
36     STATUS          Current
37     DESCRIPTION    ""
38     INDEX          { _625k-MCMgmtNetConfigTableIndex }
39     ::= { _625k-MCMgmtNetConfigTable 1 }
40
41     _625k-MCMgmtNetConfigTableEntry ::= SEQUENCE {
42         _625k-MCMgmtNetConfigTableIndex      INTEGER, -- MoNerdAddressType
43         _625k-MCEthernetIPAddress            OCTET STRING (SIZE(0..15)), --
44         IPAddressTextType                     OCTET STRING (SIZE(0..15)), --
45         _625k-MCEthernetIPLocalBits          OCTET STRING (SIZE(0..15)), --
46         IPAddressTextType                     OCTET STRING (SIZE(0..20)) --
47         _625k-MCEthernetHostName            OCTET STRING (SIZE(0..20)) --
48         TextType X 20
49     }
50
51
52
53     _625k-MCMgmtNetConfigTableIndex      OBJECT-TYPE
54     SYNTAX          INTEGER -- MoNerdAddressType
55     MAX-ACCESS      read-only
56     STATUS          Current
57     DESCRIPTION    "
58         Description for mibCtl Type 204 MoNerdAddressType :
59             Base station network component address.
60
61             A network address is a subset of Base Station component addresses,
62             restricted to network components only.
63             Network components interface with a telephony switch or similar.
64             [Limits: 0 1 ]
65         Type derived from mibCtl Type 14 Unsigned32Type :
66             32 bit unsigned integer.
67         Type derived from mibCtl Type 11 Word32Type :
68             32 bits of raw opaque data.

```

```

1      Derived from basic 32 bit word type.
2      "
3      ::= { _625k-MCMgmtNetConfigTableEntry 1 }
4
5
6
7      _625k-MCEthernetIPAddress                OBJECT-TYPE
8      SYNTAX                OCTET STRING (SIZE(0..15)) -- IPAddressTextType
9      MAX-ACCESS            read-write
10     STATUS                Current
11     DESCRIPTION
12         "Internet Protocol (IP) address for ethernet port of Module.
13
14         This is the actual IP address in use for the ethernet port
15         of a given Module.
16         If IP is not being used on the ethernet port, or there is
17         no ethernet port, then an empty string is provided for this element.
18
19         (From mibCtl ElementType 2811 EthernetIPAddress)
20         Description for mibCtl Type 420 IPAddressTextType :
21         Internet Protocol Address (Text).
22
23         This text must currently be in the dotted abc.def.ghi.jkl format.
24         In the future, hostnames might be allowed.
25         Type derived from mibCtl Type 15 TextType :
26         ASCII or compatible text.
27         Type derived from mibCtl Type 12 OctetType :
28         8 bits of raw opaque data.
29         Derived from basic 8 bit word type.
30     "
31     ::= { _625k-MCMgmtNetConfigTableEntry 2 }
32
33
34
35     _625k-MCEthernetIPLocalBits                OBJECT-TYPE
36     SYNTAX                OCTET STRING (SIZE(0..15)) -- IPAddressTextType
37     MAX-ACCESS            read-write
38     STATUS                Current
39     DESCRIPTION
40         "Ethernet IP (Internet Protocol) local routing bit count.
41
42         This indicates how many of the low-order bits of
43         the IP address of the ethernet connection are used
44         within the local network.
45         The remaining (high-order) bits are the same for all
46         hosts on the local network.
47         This is used as the first part of the routing algorithm.
48         IP addresses that do not share the upper bits of the ethernet
49         IP address and which are not otherwise resolved will be sent
50         through the gateway, if defined.
51
52         For example, 255.255.255.0
53
54         (From mibCtl ElementType 2812 EthernetIPLocalBits)
55         Description for mibCtl Type 420 IPAddressTextType :
56         Internet Protocol Address (Text).
57
58         This text must currently be in the dotted abc.def.ghi.jkl format.
59         In the future, hostnames might be allowed.
60         Type derived from mibCtl Type 15 TextType :
61         ASCII or compatible text.
62         Type derived from mibCtl Type 12 OctetType :
63         8 bits of raw opaque data.
64         Derived from basic 8 bit word type.
65     "
66     ::= { _625k-MCMgmtNetConfigTableEntry 3 }
67
68

```

```

1
2  _625k-MCEthernetHostName          OBJECT-TYPE
3      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
4      MAX-ACCESS      read-write
5      STATUS          Current
6      DESCRIPTION
7          "Ethernet IP host name for module.
8
9          (From mibCtl ElementType 2813 EthernetHostName)
10         Description for mibCtl Type 15 TextType :
11             ASCII or compatible text.
12         Type derived from mibCtl Type 12 OctetType :
13             8 bits of raw opaque data.
14         Derived from basic 8 bit word type.
15         "
16     ::= { _625k-MCMgmtNetConfigTableEntry 4 }
17
18
19
20
21  _625k-MCUserNetConfigTable         OBJECT-TYPE
22      SYNTAX SEQUENCE OF _625k-MCUserNetConfigTableEntry
23      MAX-ACCESS      not-accessible
24      STATUS          Current
25      DESCRIPTION    "User Network Configuration"
26     ::= { _625k-MCInterfaceNetwork 3 }
27
28
29
30  _625k-MCUserNetConfigTableEntry    OBJECT-TYPE
31      SYNTAX          _625k-MCUserNetConfigTableEntry
32      MAX-ACCESS      not-accessible
33      STATUS          Current
34      DESCRIPTION    ""
35      INDEX { _625k-MCUserNetConfigTableIndex }
36     ::= { _625k-MCUserNetConfigTable 1 }
37
38  _625k-MCUserNetConfigTableEntry ::= SEQUENCE {
39      _625k-MCUserNetConfigTableIndex    INTEGER, -- MoNerdAddressType
40      _625k-MCUserEthernetIPAddress      OCTET STRING (SIZE(0..15)), --
41      IPAddressTextType
42      _625k-MCUserEthernetIPLocalBits    OCTET STRING (SIZE(0..15)), --
43      IPAddressTextType
44      _625k-MCUserEthernetHostName       OCTET STRING (SIZE(0..20)) --
45      TextType X 20
46      }
47
48
49
50  _625k-MCUserNetConfigTableIndex     OBJECT-TYPE
51      SYNTAX          INTEGER -- MoNerdAddressType
52      MAX-ACCESS      read-only
53      STATUS          Current
54      DESCRIPTION    "
55          Description for mibCtl Type 204 MoNerdAddressType :
56              Base station network component address.
57
58              A network address is a subset of Base Station component addresses,
59              restricted to network components only.
60              Network components interface with a telephony switch or similar.
61              [Limits: 0 1 ]
62          Type derived from mibCtl Type 14 Unsigned32Type :
63              32 bit unsigned integer.
64          Type derived from mibCtl Type 11 Word32Type :
65              32 bits of raw opaque data.
66          Derived from basic 32 bit word type.
67          "
68     ::= { _625k-MCUserNetConfigTableEntry 1 }

```

```

1
2
3
4  _625k-MCUserEthernetIPAddress          OBJECT-TYPE
5  SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
6  MAX-ACCESS      read-write
7  STATUS          Current
8  DESCRIPTION
9      "Internet Protocol (IP) address for user ethernet port of Module.
10
11     This is the actual IP address in use for the ethernet port
12     of a given Module.
13     If IP is not being used on the ethernet port, or there is
14     no ethernet port, then an empty string is provided for this element.
15
16     (From mibCtl ElementType 2817 UserEthernetIPAddress)
17     Description for mibCtl Type 420 IPAddressTextType :
18         Internet Protocol Address (Text).
19
20         This text must currently be in the dotted abc.def.ghi.jkl format.
21         In the future, hostnames might be allowed.
22     Type derived from mibCtl Type 15 TextType :
23         ASCII or compatible text.
24     Type derived from mibCtl Type 12 OctetType :
25         8 bits of raw opaque data.
26     Derived from basic 8 bit word type.
27     "
28     ::= { _625k-MCUserNetConfigTableEntry 2 }
29
30
31
32  _625k-MCUserEthernetIPLocalBits        OBJECT-TYPE
33  SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
34  MAX-ACCESS      read-write
35  STATUS          Current
36  DESCRIPTION
37      "Ethernet IP subnet mask for user network.
38
39     Ethernet IP subnet mask for user network
40
41     (From mibCtl ElementType 2818 UserEthernetIPLocalBits)
42     Description for mibCtl Type 420 IPAddressTextType :
43         Internet Protocol Address (Text).
44
45         This text must currently be in the dotted abc.def.ghi.jkl format.
46         In the future, hostnames might be allowed.
47     Type derived from mibCtl Type 15 TextType :
48         ASCII or compatible text.
49     Type derived from mibCtl Type 12 OctetType :
50         8 bits of raw opaque data.
51     Derived from basic 8 bit word type.
52     "
53     ::= { _625k-MCUserNetConfigTableEntry 3 }
54
55
56
57  _625k-MCUserEthernetHostName           OBJECT-TYPE
58  SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
59  MAX-ACCESS      read-write
60  STATUS          Current
61  DESCRIPTION
62      "User ethernet IP host name for module.
63
64     (From mibCtl ElementType 2819 UserEthernetHostName)
65     Description for mibCtl Type 15 TextType :
66         ASCII or compatible text.
67     Type derived from mibCtl Type 12 OctetType :
68         8 bits of raw opaque data.

```

```

1         Derived from basic 8 bit word type.
2         "
3         ::= { _625k-MCUserNetConfigTableEntry 4 }
4
5
6
7
8         _625k-MCUserNetStatusTable                OBJECT-TYPE
9         SYNTAX SEQUENCE OF _625k-MCUserNetStatusTableEntry
10        MAX-ACCESS not-accessible
11        STATUS Current
12        DESCRIPTION "Network Status"
13        ::= { _625K-MCInterfaceNetwork 4 }
14
15
16
17        _625k-MCUserNetStatusTableEntry            OBJECT-TYPE
18        SYNTAX _625k-MCUserNetStatusTableEntry
19        MAX-ACCESS not-accessible
20        STATUS Current
21        DESCRIPTION ""
22        INDEX { _625k-MCUserNetStatusTableIndex }
23        ::= { _625k-MCUserNetStatusTable 1 }
24
25        _625k-MCUserNetStatusTableEntry ::= SEQUENCE {
26        _625k-MCUserNetStatusTableIndex            INTEGER, -- MoNerdAddressType
27        _625k-MCNetworkInOctets                    Counter, -- Counter32Type
28        _625k-MCNetworkOutOctets                   Counter -- Counter32Type
29        }
30
31
32
33        _625k-MCUserNetStatusTableIndex            OBJECT-TYPE
34        SYNTAX INTEGER -- MoNerdAddressType
35        MAX-ACCESS read-only
36        STATUS Current
37        DESCRIPTION "
38        Description for mibCtl Type 204 MoNerdAddressType :
39        Base station network component address.
40
41        A network address is a subset of Base Station component addresses,
42        restricted to network components only.
43        Network components interface with a telephony switch or similar.
44        [Limits: 0 1 ]
45        Type derived from mibCtl Type 14 Unsigned32Type :
46        32 bit unsigned integer.
47        Type derived from mibCtl Type 11 Word32Type :
48        32 bits of raw opaque data.
49        Derived from basic 32 bit word type.
50        "
51        ::= { _625k-MCUserNetStatusTableEntry 1 }
52
53
54
55        _625k-MCNetworkInOctets                    OBJECT-TYPE
56        SYNTAX Counter32 -- Counter32Type
57        MAX-ACCESS read-only
58        STATUS Current
59        DESCRIPTION
60        "In octets user data of network.
61
62
63
64        (From mibCtl ElementType 1000 NetworkInOctets)
65        Description for mibCtl Type 19 Counter32Type :
66        32 bits of Counter data.
67        Derived from basic 32 bit word type.
68        "

```

```

1      ::= { _625k-MCUserNetStatusTableEntry 2 }
2
3
4
5      _625k-MCNetworkOutOctets          OBJECT-TYPE
6      SYNTAX          Counter32 -- Counter32Type
7      MAX-ACCESS      read-only
8      STATUS          Current
9      DESCRIPTION
10     "Out octets user data of network.
11
12
13     (From mibCtl ElementType 1001 NetworkOutOctets)
14     Description for mibCtl Type 19 Counter32Type :
15     32 bits of Counter data.
16     Derived from basic 32 bit word type.
17     "
18     ::= { _625k-MCUserNetStatusTableEntry 3 }
19
20
21
22
23
24     _625k-MCL2TPConfigTable          OBJECT-TYPE
25     SYNTAX SEQUENCE OF _625k-MCL2TPConfigTableEntry
26     MAX-ACCESS      not-accessible
27     STATUS          Current
28     DESCRIPTION    "L2TP Configuration Table"
29     ::= { _625k-MCInterfaceNetwork 5 }
30
31
32
33     _625k-MCL2TPConfigTableEntry     OBJECT-TYPE
34     SYNTAX          _625k-MCL2TPConfigTableEntry
35     MAX-ACCESS      not-accessible
36     STATUS          Current
37     DESCRIPTION    ""
38     INDEX { _625k-MCL2TPConfigTableIndex }
39     ::= { _625k-MCL2TPConfigTable 1 }
40
41     _625k-MCL2TPConfigTableEntry ::= SEQUENCE {
42         _625k-MCL2TPConfigTableIndex    INTEGER, -- MoNerdAddressType
43         _625k-MCL2TPPeerName            OCTET STRING (SIZE(0..20)), --
44         TextType X 20
45         _625k-MCL2TPPeerIPAddress       OCTET STRING (SIZE(0..15)), --
46         IPAddressTextType
47         _625k-MCL2TPAVPHostName         OCTET STRING (SIZE(0..20)), --
48         TextType X 20
49         _625k-MCL2TPAVPChallAndRes     OCTET STRING (SIZE(0..20)) --
50         TextType X 20
51     }
52
53
54
55     _625k-MCL2TPConfigTableIndex     OBJECT-TYPE
56     SYNTAX          INTEGER -- MoNerdAddressType
57     MAX-ACCESS      read-only
58     STATUS          Current
59     DESCRIPTION    "
60     Description for mibCtl Type 204 MoNerdAddressType :
61     Base station network component address.
62
63     A network address is a subset of Base Station component addresses,
64     restricted to network components only.
65     Network components interface with a telephony switch or similar.
66     [Limits: 0 1 ]
67     Type derived from mibCtl Type 14 Unsigned32Type :
68     32 bit unsigned integer.

```

```

1      Type derived from mibCtl Type 11 Word32Type :
2      32 bits of raw opaque data.
3      Derived from basic 32 bit word type.
4      "
5      ::= { _625k-MCL2TPConfigTableEntry 1 }
6
7
8
9      _625k-MCL2TPPeerName                OBJECT-TYPE
10     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
11     MAX-ACCESS      read-write
12     STATUS          Current
13     DESCRIPTION
14         "L2TP peer name.
15
16         Tunnel switch host name
17
18         (From mibCtl ElementType 2000 L2TPPeerName)
19         Description for mibCtl Type 15 TextType :
20             ASCII or compatible text.
21         Type derived from mibCtl Type 12 OctetType :
22             8 bits of raw opaque data.
23         Derived from basic 8 bit word type.
24         "
25     ::= { _625k-MCL2TPConfigTableEntry 2 }
26
27
28
29     _625k-MCL2TPPeerIPAddress            OBJECT-TYPE
30     SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
31     MAX-ACCESS      read-write
32     STATUS          Current
33     DESCRIPTION
34         "L2TP peer IP Address.
35
36         Tunnel switch IP Address
37
38         (From mibCtl ElementType 2001 L2TPPeerIPAddress)
39         Description for mibCtl Type 420 IPAddressTextType :
40             Internet Protocol Address (Text).
41
42             This text must currently be in the dotted abc.def.ghi.jkl format.
43             In the future, hostnames might be allowed.
44         Type derived from mibCtl Type 15 TextType :
45             ASCII or compatible text.
46         Type derived from mibCtl Type 12 OctetType :
47             8 bits of raw opaque data.
48         Derived from basic 8 bit word type.
49         "
50     ::= { _625k-MCL2TPConfigTableEntry 3 }
51
52
53
54     _625k-MCL2TPAVPHostName              OBJECT-TYPE
55     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
56     MAX-ACCESS      read-write
57     STATUS          Current
58     DESCRIPTION
59         "BS host name using L2TP.
60
61
62
63         (From mibCtl ElementType 2008 L2TPAVPHostName)
64         Description for mibCtl Type 15 TextType :
65             ASCII or compatible text.
66         Type derived from mibCtl Type 12 OctetType :
67             8 bits of raw opaque data.
68         Derived from basic 8 bit word type.

```



```

1      "
2      ::= { _625k-MCL2TPConfigTableEntry 4 }
3
4
5
6      _625k-MCL2TPAVPChallAndRes          OBJECT-TYPE
7      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
8      MAX-ACCESS      read-write
9      STATUS          Current
10     DESCRIPTION
11         "AVP challenge and response name.
12
13
14
15         (From mibCtl ElementType 2012 L2TPAVPChallAndRes)
16         Description for mibCtl Type 15 TextType :
17             ASCII or compatible text.
18         Type derived from mibCtl Type 12 OctetType :
19             8 bits of raw opaque data.
20         Derived from basic 8 bit word type.
21     "
22     ::= { _625k-MCL2TPConfigTableEntry 5 }
23
24
25
26
27     _625k-MCL2TPStatusTable              OBJECT-TYPE
28     SYNTAX SEQUENCE OF _625k-MCL2TPStatusTableEntry
29     MAX-ACCESS      not-accessible
30     STATUS          Current
31     DESCRIPTION    "L2TP Status Table"
32     ::= { _625k-MCInterfaceNetwork 6 }
33
34
35
36     _625k-MCL2TPStatusTableEntry        OBJECT-TYPE
37     SYNTAX          _625k-MCL2TPStatusTableEntry
38     MAX-ACCESS      not-accessible
39     STATUS          Current
40     DESCRIPTION    ""
41     INDEX { _625k-MCL2TPStatusTableIndex }
42     ::= { _625k-MCL2TPStatusTable 1 }
43
44     _625k-MCL2TPStatusTableEntry ::= SEQUENCE {
45         _625k-MCL2TPStatusTableIndex    INTEGER, -- MoNerdAddressType
46         _625k-MCL2TPActiveSession       Unsigned32, -- Unsigned32Type
47         _625k-MCL2TPActiveTunnel        Unsigned32 -- Unsigned32Type
48     }
49
50
51
52     _625k-MCL2TPStatusTableIndex        OBJECT-TYPE
53     SYNTAX          INTEGER -- MoNerdAddressType
54     MAX-ACCESS      read-only
55     STATUS          Current
56     DESCRIPTION    "
57         Description for mibCtl Type 204 MoNerdAddressType :
58         Base station network component address.
59
60         A network address is a subset of Base Station component addresses,
61         restricted to network components only.
62         Network components interface with a telephony switch or similar.
63         [Limits: 0 1 ]
64         Type derived from mibCtl Type 14 Unsigned32Type :
65             32 bit unsigned integer.
66         Type derived from mibCtl Type 11 Word32Type :
67             32 bits of raw opaque data.
68         Derived from basic 32 bit word type.

```

```

1      "
2      ::= { _625k-MCL2TPStatusTableEntry 1 }
3
4
5
6      _625k-MCL2TPActiveSession          OBJECT-TYPE
7      SYNTAX          Unsigned32 -- Unsigned32Type
8      MAX-ACCESS      read-only
9      STATUS          Current
10     DESCRIPTION
11         "L2TP active session.
12
13
14
15         (From mibCtl ElementType 2013 L2TPActiveSession)
16         Description for mibCtl Type 14 Unsigned32Type :
17             32 bit unsigned integer.
18         Type derived from mibCtl Type 11 Word32Type :
19             32 bits of raw opaque data.
20         Derived from basic 32 bit word type.
21         "
22     ::= { _625k-MCL2TPStatusTableEntry 2 }
23
24
25
26     _625k-MCL2TPActiveTunnel            OBJECT-TYPE
27     SYNTAX          Unsigned32 -- Unsigned32Type
28     MAX-ACCESS      read-only
29     STATUS          Current
30     DESCRIPTION
31         "L2TP active tunnel.
32
33
34
35         (From mibCtl ElementType 2014 L2TPActiveTunnel)
36         Description for mibCtl Type 14 Unsigned32Type :
37             32 bit unsigned integer.
38         Type derived from mibCtl Type 11 Word32Type :
39             32 bits of raw opaque data.
40         Derived from basic 32 bit word type.
41         "
42     ::= { _625k-MCL2TPStatusTableEntry 3 }
43
44
45
46
47     _625k-MCATMConfigTable              OBJECT-TYPE
48     SYNTAX SEQUENCE OF _625k-MCATMConfigTableEntry
49     MAX-ACCESS      not-accessible
50     STATUS          Current
51     DESCRIPTION    "ATM Configuration Table"
52     ::= { _625k-MCInterfaceNetwork 7 }
53
54
55
56     _625k-MCATMConfigTableEntry          OBJECT-TYPE
57     SYNTAX          _625k-MCATMConfigTableEntry
58     MAX-ACCESS      not-accessible
59     STATUS          Current
60     DESCRIPTION    ""
61     INDEX { _625k-MCATMConfigTableIndex }
62     ::= { _625k-MCATMConfigTable 1 }
63
64     _625k-MCATMConfigTableEntry ::= SEQUENCE {
65         _625k-MCATMConfigTableIndex      INTEGER, -- MoNerdAddressType
66         _625k-MCATmAddress                OCTET STRING (SIZE(0..40)), --
67         TextType X 40
68         _625k-MCATmVCTypes                INTEGER, -- AtmVCType

```

```

1      _625k-MCAtmFrameTypes                INTEGER, -- AtmFrameType
2      _625k-MCAtmUNIVersion                INTEGER, -- AtmUNIVersionType
3      _625k-MCAtmLineStatus                INTEGER, -- LineStatusType
4      _625k-MCAtmParameterFailReason      Unsigned32, -- Unsigned32Type
5      _625k-MCAtmOpenChannelFailReason    Unsigned32, -- Unsigned32Type
6      _625k-MCAtmChannelNumber            Unsigned32, -- Unsigned32Type
7      _625k-MCAtmAlarmCauseRegister       Unsigned32, -- Unsigned32Type
8      _625k-MCAtmPHYIntrCauseRegister     Unsigned32 -- Unsigned32Type
9      }
10
11
12
13     _625k-MCATMConfigTableIndex           OBJECT-TYPE
14     SYNTAX                               INTEGER -- MoNerdAddressType
15     MAX-ACCESS                           read-only
16     STATUS                               Current
17     DESCRIPTION                          "
18     Description for mibCtl Type 204 MoNerdAddressType :
19     Base station network component address.
20
21     A network address is a subset of Base Station component addresses,
22     restricted to network components only.
23     Network components interface with a telephony switch or similar.
24     [Limits: 0 1 ]
25     Type derived from mibCtl Type 14 Unsigned32Type :
26     32 bit unsigned integer.
27     Type derived from mibCtl Type 11 Word32Type :
28     32 bits of raw opaque data.
29     Derived from basic 32 bit word type.
30     "
31     ::= { _625k-MCATMConfigTableEntry 1 }
32
33
34
35     _625k-MCAtmAddress                     OBJECT-TYPE
36     SYNTAX                               OCTET STRING (SIZE(0..40)) -- TextType X 40
37     MAX-ACCESS                           read-write
38     STATUS                               Current
39     DESCRIPTION                          "
40     "Atm Address.
41
42
43
44     (From mibCtl ElementType 1950 AtmAddress)
45     Description for mibCtl Type 15 TextType :
46     ASCII or compatible text.
47     Type derived from mibCtl Type 12 OctetType :
48     8 bits of raw opaque data.
49     Derived from basic 8 bit word type.
50     "
51     ::= { _625k-MCATMConfigTableEntry 2 }
52
53
54
55     _625k-MCAtmVCTypes                     OBJECT-TYPE
56     SYNTAX                               INTEGER -- AtmVCType
57     MAX-ACCESS                           read-write
58     STATUS                               Current
59     DESCRIPTION                          "
60     "Atm VC Type.
61
62
63
64     (From mibCtl ElementType 1951 AtmVCTypes)
65     Description for mibCtl Type 300 AtmVCType :
66     ATM VC Type.
67     Description for mibCtl AtmVCType 0 Unknown :
68     ATM VC Type is Unknown.

```

```

1      Description for mibCtl AtmVCType 1 PVC :
2          ATM VC Type is PVC.
3      Description for mibCtl AtmVCType 2 SVC :
4          ATM VC Type is SVC.
5      Description for mibCtl AtmVCType 3 PVC SVC :
6          ATM VC Type is PVC & SVC.
7      "
8      ::= { _625k-MCATMConfigTableEntry 3 }
9
10
11
12     _625k-MCATmFrameTypes                                OBJECT-TYPE
13     SYNTAX          INTEGER -- AtmFrameType
14     MAX-ACCESS      read-write
15     STATUS          Current
16     DESCRIPTION
17         "Atm Frame Type.
18
19
20         (From mibCtl ElementType 1952 AtmFrameTypes)
21     Description for mibCtl Type 301 AtmFrameType :
22         ATM Frame Type.
23     Description for mibCtl AtmFrameType 0 Unknown :
24         ATM Frame Type is Unknown.
25     Description for mibCtl AtmFrameType 1 OC48 :
26         ATM Frame Type is OC48.
27     Description for mibCtl AtmFrameType 2 OC36 :
28         ATM Frame Type is OC36.
29     Description for mibCtl AtmFrameType 3 OC24 :
30         ATM Frame Type is OC24.
31     Description for mibCtl AtmFrameType 4 OC18 :
32         ATM Frame Type is OC18.
33     Description for mibCtl AtmFrameType 5 OC12 :
34         ATM Frame Type is OC12.
35     Description for mibCtl AtmFrameType 6 OC9 :
36         ATM Frame Type is OC9.
37     Description for mibCtl AtmFrameType 7 OC3 :
38         ATM Frame Type is OC3.
39     Description for mibCtl AtmFrameType 8 OC1 :
40         ATM Frame Type is OC1.
41     Description for mibCtl AtmFrameType 9 STM16 :
42         ATM Frame Type is STM16.
43     Description for mibCtl AtmFrameType 10 STM4 :
44         ATM Frame Type is STM4.
45     Description for mibCtl AtmFrameType 11 STM1 :
46         ATM Frame Type is STM1.
47     Description for mibCtl AtmFrameType 12 DS3 :
48         ATM Frame Type is DS3.
49     Description for mibCtl AtmFrameType 13 DS2 :
50         ATM Frame Type is DS2.
51     Description for mibCtl AtmFrameType 14 DS1 :
52         ATM Frame Type is DS1.
53     Description for mibCtl AtmFrameType 15 DS0 :
54         ATM Frame Type is DS0.
55     Description for mibCtl AtmFrameType 16 E3 :
56         ATM Frame Type is E3.
57     Description for mibCtl AtmFrameType 17 E2 :
58         ATM Frame Type is E2.
59     Description for mibCtl AtmFrameType 18 E1 :
60         ATM Frame Type is E1.
61     Description for mibCtl AtmFrameType 19 E0 :
62         ATM Frame Type is E0.
63     "
64     ::= { _625k-MCATMConfigTableEntry 4 }
65
66
67
68

```

```

1  _625k-MCATmUNIVersion                OBJECT-TYPE
2  SYNTAX                               INTEGER -- AtmUNIVersionType
3  MAX-ACCESS                           read-write
4  STATUS                                Current
5  DESCRIPTION
6      "Atm UNI version.
7
8
9
10     (From mibCtl ElementType 1953 AtmUNIVersion)
11     Description for mibCtl Type 302 AtmUNIVersionType :
12         ATM UNI Version Type.
13     Description for mibCtl AtmUNIVersionType 0 Unknown :
14         ATM UNI Version Type is Unknown.
15     Description for mibCtl AtmUNIVersionType 1 V30 :
16         ATM UNI Version Type is 3.0.
17     Description for mibCtl AtmUNIVersionType 2 V31 :
18         ATM UNI Version Type is 3.1.
19     Description for mibCtl AtmUNIVersionType 3 V40 :
20         ATM UNI Version Type is 4.0.
21     "
22 ::= { _625k-MCATMConfigTableEntry 5 }
23
24
25
26 _625k-MCATmLineStatus                 OBJECT-TYPE
27 SYNTAX                               INTEGER -- LineStatusType
28 MAX-ACCESS                           read-only
29 STATUS                                Current
30 DESCRIPTION
31     "Atm line status.
32
33     ATM line status
34
35     (From mibCtl ElementType 1956 AtmLineStatus)
36     Description for mibCtl Type 72 LineStatusType :
37         Line status type.
38     Description for mibCtl LineStatusType 0 LinkUp :
39         Line status is link up.
40     Description for mibCtl LineStatusType 1 LinkDown :
41         Line status is link down.
42     "
43 ::= { _625k-MCATMConfigTableEntry 6 }
44
45
46
47 _625k-MCATmParameterFailReason        OBJECT-TYPE
48 SYNTAX                               Unsigned32 -- Unsigned32Type
49 MAX-ACCESS                           read-only
50 STATUS                                Current
51 DESCRIPTION
52     "Atm parameter fail reason.
53
54     ATM parameter fail reason
55
56     (From mibCtl ElementType 1957 AtmParameterFailReason)
57     Description for mibCtl Type 14 Unsigned32Type :
58         32 bit unsigned integer.
59     Type derived from mibCtl Type 11 Word32Type :
60         32 bits of raw opaque data.
61     Derived from basic 32 bit word type.
62     "
63 ::= { _625k-MCATMConfigTableEntry 7 }
64
65
66
67 _625k-MCATmOpenChannelFailReason       OBJECT-TYPE
68 SYNTAX                               Unsigned32 -- Unsigned32Type

```

```

1     MAX-ACCESS          read-only
2     STATUS              Current
3     DESCRIPTION
4         "Atm open channel fail reason."
5
6         ATM open channel fail reason
7
8         (From mibCtl ElementType 1958 AtmOpenChannelFailReason)
9         Description for mibCtl Type 14 Unsigned32Type :
10            32 bit unsigned integer.
11         Type derived from mibCtl Type 11 Word32Type :
12            32 bits of raw opaque data.
13         Derived from basic 32 bit word type.
14         "
15     ::= { _625k-MCATMConfigTableEntry 8 }
16
17
18
19     _625k-MCAtmChannelNumber          OBJECT-TYPE
20     SYNTAX          Unsigned32 -- Unsigned32Type
21     MAX-ACCESS      read-only
22     STATUS          Current
23     DESCRIPTION
24         "Atm open channel fail reason."
25
26         ATM open channel fail reason
27
28         (From mibCtl ElementType 1959 AtmChannelNumber)
29         Description for mibCtl Type 14 Unsigned32Type :
30            32 bit unsigned integer.
31         Type derived from mibCtl Type 11 Word32Type :
32            32 bits of raw opaque data.
33         Derived from basic 32 bit word type.
34         "
35     ::= { _625k-MCATMConfigTableEntry 9 }
36
37
38
39     _625k-MCAtmAlarmCauseRegister     OBJECT-TYPE
40     SYNTAX          Unsigned32 -- Unsigned32Type
41     MAX-ACCESS      read-only
42     STATUS          Current
43     DESCRIPTION
44         "Atm alarm cause register."
45
46         ATM alarm cause register
47
48         (From mibCtl ElementType 1960 AtmAlarmCauseRegister)
49         Description for mibCtl Type 14 Unsigned32Type :
50            32 bit unsigned integer.
51         Type derived from mibCtl Type 11 Word32Type :
52            32 bits of raw opaque data.
53         Derived from basic 32 bit word type.
54         "
55     ::= { _625k-MCATMConfigTableEntry 10 }
56
57
58
59     _625k-MCAtmPHYIntrCauseRegister   OBJECT-TYPE
60     SYNTAX          Unsigned32 -- Unsigned32Type
61     MAX-ACCESS      read-only
62     STATUS          Current
63     DESCRIPTION
64         "Atm PHY Interrupt cause register."
65
66         ATM PHY interrupt cause register
67
68         (From mibCtl ElementType 1961 AtmPHYIntrCauseRegister)

```

```

1      Description for mibCtl Type 14 Unsigned32Type :
2          32 bit unsigned integer.
3      Type derived from mibCtl Type 11 Word32Type :
4          32 bits of raw opaque data.
5      Derived from basic 32 bit word type.
6      "
7      ::= { _625k-MCATMConfigTableEntry 11 }
8
9
10
11
12     _625k-MCA10ConfigTable                OBJECT-TYPE
13     SYNTAX SEQUENCE OF _625k-MCA10ConfigTableEntry
14     MAX-ACCESS                not-accessible
15     STATUS                     Current
16     DESCRIPTION                "A10 Configuration Table"
17     ::= { _625k-MCInterfaceNetwork 9 }
18
19
20
21     _625k-MCA10ConfigTableEntry           OBJECT-TYPE
22     SYNTAX                     _625k-MCA10ConfigTableEntry
23     MAX-ACCESS                not-accessible
24     STATUS                     Current
25     DESCRIPTION                ""
26     INDEX { _625k-MCA10ConfigTableIndex }
27     ::= { _625k-MCA10ConfigTable 1 }
28
29     _625k-MCA10ConfigTableEntry ::= SEQUENCE {
30         _625k-MCA10ConfigTableIndex      INTEGER, -- MoNerdAddressType
31         _625k-MC3GPP2PriPDSNIPAddress    OCTET STRING (SIZE(0..15)), --
32     IPAddressTextType
33         _625k-MC3GPP2PriPDSNSharedSecret OCTET STRING (SIZE(0..64)), --
34     TextType X 64
35         _625k-MC3GPP2PriPDSNSPI         Unsigned32, -- Unsigned32Type
36         _625k-MC3GPP2SecPDSNIPAddress    OCTET STRING (SIZE(0..15)), --
37     IPAddressTextType
38         _625k-MC3GPP2SecPDSNSharedSecret OCTET STRING (SIZE(0..64)), --
39     TextType X 64
40         _625k-MC3GPP2SecPDSNSPI         Unsigned32 -- Unsigned32Type
41     }
42
43
44
45     _625k-MCA10ConfigTableIndex           OBJECT-TYPE
46     SYNTAX                     INTEGER -- MoNerdAddressType
47     MAX-ACCESS                read-only
48     STATUS                     Current
49     DESCRIPTION                "
50         Description for mibCtl Type 204 MoNerdAddressType :
51         Base station network component address.
52
53         A network address is a subset of Base Station component addresses,
54         restricted to network components only.
55         Network components interface with a telephony switch or similar.
56         [Limits: 0 1 ]
57         Type derived from mibCtl Type 14 Unsigned32Type :
58         32 bit unsigned integer.
59         Type derived from mibCtl Type 11 Word32Type :
60         32 bits of raw opaque data.
61         Derived from basic 32 bit word type.
62     "
63     ::= { _625k-MCA10ConfigTableEntry 1 }
64
65
66
67     _625k-MC3GPP2PriPDSNIPAddress         OBJECT-TYPE
68     SYNTAX                     OCTET STRING (SIZE(0..15)) -- IPAddressTextType

```

```

1     MAX-ACCESS          read-write
2     STATUS              Current
3     DESCRIPTION
4         "3GPP2 primary PDSN IP address.
5
6         3GPP2 Primary PDSN IP Address
7
8         (From mibCtl ElementType 2100 3GPP2PriPDSNIPAddress)
9     Description for mibCtl Type 420 IPAddressTextType :
10        Internet Protocol Address (Text).
11
12        This text must currently be in the dotted abc.def.ghi.jkl format.
13        In the future, hostnames might be allowed.
14    Type derived from mibCtl Type 15 TextType :
15        ASCII or compatible text.
16    Type derived from mibCtl Type 12 OctetType :
17        8 bits of raw opaque data.
18    Derived from basic 8 bit word type.
19    "
20 ::= { _625k-MCA10ConfigTableEntry 2 }
21
22
23
24 _625k-MC3GPP2PriPDSNSharedSecret          OBJECT-TYPE
25     SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
26     MAX-ACCESS      read-write
27     STATUS          Current
28     DESCRIPTION
29         "3GPP2 primary PDSN shared secret.
30
31         3GPP2 primary PDSN shared secret
32
33         (From mibCtl ElementType 2101 3GPP2PriPDSNSharedSecret)
34     Description for mibCtl Type 15 TextType :
35        ASCII or compatible text.
36    Type derived from mibCtl Type 12 OctetType :
37        8 bits of raw opaque data.
38    Derived from basic 8 bit word type.
39    "
40 ::= { _625k-MCA10ConfigTableEntry 3 }
41
42
43
44 _625k-MC3GPP2PriPDSNSPI                   OBJECT-TYPE
45     SYNTAX          Unsigned32 -- Unsigned32Type
46     MAX-ACCESS      read-write
47     STATUS          Current
48     DESCRIPTION
49         "3GPP2 primary PDSN SPI.
50
51         3GPP2 primary PDSN SPI
52
53         (From mibCtl ElementType 2102 3GPP2PriPDSNSPI)
54     Description for mibCtl Type 14 Unsigned32Type :
55        32 bit unsigned integer.
56    Type derived from mibCtl Type 11 Word32Type :
57        32 bits of raw opaque data.
58    Derived from basic 32 bit word type.
59    "
60 ::= { _625k-MCA10ConfigTableEntry 4 }
61
62
63
64 _625k-MC3GPP2SecPDSNIPAddress            OBJECT-TYPE
65     SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
66     MAX-ACCESS      read-write
67     STATUS          Current
68     DESCRIPTION

```



```

1      "3GPP2 secondary PDSN IP address.
2
3      3GPP2 secondary PDSN IP address
4
5      (From mibCtl ElementType 2103 3GPP2SecPDSNIPAddress)
6      Description for mibCtl Type 420 IPAddressTextType :
7          Internet Protocol Address (Text).
8
9          This text must currently be in the dotted abc.def.ghi.jkl format.
10         In the future, hostnames might be allowed.
11         Type derived from mibCtl Type 15 TextType :
12             ASCII or compatible text.
13         Type derived from mibCtl Type 12 OctetType :
14             8 bits of raw opaque data.
15         Derived from basic 8 bit word type.
16         "
17     ::= { _625k-MCA10ConfigTableEntry 5 }
18
19
20
21     _625k-MC3GPP2SecPDSNSharedSecret          OBJECT-TYPE
22     SYNTAX          OCTET STRING (SIZE(0..64)) -- TextType X 64
23     MAX-ACCESS      read-write
24     STATUS          Current
25     DESCRIPTION
26         "3GPP2 secondary PDSN shared secret.
27
28         3GPP2 secondary PDSN shared secret
29
30         (From mibCtl ElementType 2104 3GPP2SecPDSNSharedSecret)
31         Description for mibCtl Type 15 TextType :
32             ASCII or compatible text.
33         Type derived from mibCtl Type 12 OctetType :
34             8 bits of raw opaque data.
35         Derived from basic 8 bit word type.
36         "
37     ::= { _625k-MCA10ConfigTableEntry 6 }
38
39
40
41     _625k-MC3GPP2SecPDSNSPI                    OBJECT-TYPE
42     SYNTAX          Unsigned32 -- Unsigned32Type
43     MAX-ACCESS      read-write
44     STATUS          Current
45     DESCRIPTION
46         "3GPP2 secondary PDSN SPI.
47
48         3GPP2 secondary PDSN SPI
49
50         (From mibCtl ElementType 2105 3GPP2SecPDSNSPI)
51         Description for mibCtl Type 14 Unsigned32Type :
52             32 bit unsigned integer.
53         Type derived from mibCtl Type 11 Word32Type :
54             32 bits of raw opaque data.
55         Derived from basic 32 bit word type.
56         "
57     ::= { _625k-MCA10ConfigTableEntry 7 }
58
59
60
61
62     _625k-MCA10StatusTable                    OBJECT-TYPE
63     SYNTAX SEQUENCE OF _625k-MCA10StatusTableEntry
64     MAX-ACCESS      not-accessible
65     STATUS          Current
66     DESCRIPTION    "A10 Status Table"
67     ::= { _625k-MCInterfaceNetwork 10 }
68

```

```

1
2
3  _625k-MCA10StatusTableEntry          OBJECT-TYPE
4  SYNTAX          _625k-MCA10StatusTableEntry
5  MAX-ACCESS      not-accessible
6  STATUS          Current
7  DESCRIPTION     ""
8  INDEX           { _625k-MCA10StatusTableIndex }
9  ::= { _625k-MCA10StatusTable 1 }
10
11 _625k-MCA10StatusTableEntry ::= SEQUENCE {
12   _625k-MCA10StatusTableIndex      INTEGER, -- MoNerdAddressType
13   _625k-MC3GPP2PDSNIPAddress       OCTET STRING (SIZE(0..15)) --
14   IPAddressTextType
15 }
16
17
18
19 _625k-MCA10StatusTableIndex          OBJECT-TYPE
20 SYNTAX          INTEGER -- MoNerdAddressType
21 MAX-ACCESS      read-only
22 STATUS          Current
23 DESCRIPTION     "
24   Description for mibCtl Type 204 MoNerdAddressType :
25   Base station network component address.
26
27   A network address is a subset of Base Station component addresses,
28   restricted to network components only.
29   Network components interface with a telephony switch or similar.
30   [Limits: 0 1 ]
31   Type derived from mibCtl Type 14 Unsigned32Type :
32   32 bit unsigned integer.
33   Type derived from mibCtl Type 11 Word32Type :
34   32 bits of raw opaque data.
35   Derived from basic 32 bit word type.
36   "
37   ::= { _625k-MCA10StatusTableEntry 1 }
38
39
40
41 _625k-MC3GPP2PDSNIPAddress          OBJECT-TYPE
42 SYNTAX          OCTET STRING (SIZE(0..15)) -- IPAddressTextType
43 MAX-ACCESS      read-only
44 STATUS          Current
45 DESCRIPTION     "
46   "3GPP2 PDSN IP Address.
47
48   Current main using PDSN IP address
49
50   (From mibCtl ElementType 2113 3GPP2PDSNIPAddress)
51   Description for mibCtl Type 420 IPAddressTextType :
52   Internet Protocol Address (Text).
53
54   This text must currently be in the dotted abc.def.ghi.jkl format.
55   In the future, hostnames might be allowed.
56   Type derived from mibCtl Type 15 TextType :
57   ASCII or compatible text.
58   Type derived from mibCtl Type 12 OctetType :
59   8 bits of raw opaque data.
60   Derived from basic 8 bit word type.
61   "
62   ::= { _625k-MCA10StatusTableEntry 2 }
63
64
65
66 _625k-MCInterfaceRF                OBJECT IDENTIFIER
67 -- DESCRIPTION          "Radio Frequency Interfaces"
68 ::= { _625k-MCSysInterfaces 2 }

```

```

1
2
3
4  _625k-MCCarrierTable                OBJECT-TYPE
5      SYNTAX SEQUENCE OF _625k-MCCarrierTableEntry
6      MAX-ACCESS not-accessible
7      STATUS Current
8      DESCRIPTION "Carrier Frequency Table"
9      ::= { _625k-MCInterfaceRF 1 }
10
11
12
13 _625k-MCCarrierTableEntry            OBJECT-TYPE
14     SYNTAX _625k-MCCarrierTableEntry
15     MAX-ACCESS not-accessible
16     STATUS Current
17     DESCRIPTION ""
18     INDEX { _625k-MCCarrierTableIndex }
19     ::= { _625k-MCCarrierTable 1 }
20
21 _625k-MCCarrierTableEntry ::= SEQUENCE {
22     _625k-MCCarrierTableIndex        INTEGER, -- BaseStationCarrierType
23     _625k-MCCarrierUsage              INTEGER -- CarrierUsageType
24 }
25
26
27
28 _625k-MCCarrierTableIndex            OBJECT-TYPE
29     SYNTAX INTEGER -- BaseStationCarrierType
30     MAX-ACCESS read-only
31     STATUS Current
32     DESCRIPTION "
33         Description for mibCtl Type 219 BaseStationCarrierType :
34         Base station carrier number.
35
36         Base station carriers are a contiguous set of carriers
37         that are used by the Base Station;
38         they are numbered from 0 to a current maximum of 32-1.
39         [Limits: 0 15 ]
40         Type derived from mibCtl Type 14 Unsigned32Type :
41         32 bit unsigned integer.
42         Type derived from mibCtl Type 11 Word32Type :
43         32 bits of raw opaque data.
44         Derived from basic 32 bit word type.
45     "
46     ::= { _625k-MCCarrierTableEntry 1 }
47
48
49
50 _625k-MCCarrierUsage                  OBJECT-TYPE
51     SYNTAX INTEGER -- CarrierUsageType
52     MAX-ACCESS read-only
53     STATUS Current
54     DESCRIPTION "Current assigned usage per base station carrier.
55
56
57
58
59     (From mibCtl ElementType 54 CarrierUsage)
60     Description for mibCtl Type 220 CarrierUsageType :
61     The assigned use of a radio carrier.
62
63     A radio carrier is a frequency band.
64     The assigned use of a carrier can be Reserved, Control
65     or Traffic.
66     Base station transmits control information on one of the time
67 slots
68     of given Control carrier.

```

```

1         Base station does not transmit anything on Reserved carriers.
2     Description for mibCtl CarrierUsageType 0 NotUse :
3         Not Use for this carrier.
4     Description for mibCtl CarrierUsageType 1 TCH :
5         All timeslots in this carrier are for traffic only.
6     Description for mibCtl CarrierUsageType 2 TCHBCH :
7         One timeslot in this carrier is for BCH, others for TCH.
8     "
9     ::= { _625k-MCCarrierTableEntry 2 }
10
11
12
13     _625k-MCBSCC                                OBJECT-TYPE
14     SYNTAX                Unsigned32 -- Unsigned32Type
15     MAX-ACCESS            read-write
16     STATUS                Current
17     DESCRIPTION
18         "Base Station Color Code.
19
20         (From mibCtl ElementType 61 BSCC)
21     Description for mibCtl Type 14 Unsigned32Type :
22         32 bit unsigned integer.
23     Type derived from mibCtl Type 11 Word32Type :
24         32 bits of raw opaque data.
25     Derived from basic 32 bit word type.
26     "
27     ::= { _625k-MCInterfaceRF 2 }
28
29
30
31     _625k-MCBSLowestCarrier                    OBJECT-TYPE
32     SYNTAX                OCTET STRING (SIZE(0..20)) -- TextType X 20
33     MAX-ACCESS            read-write
34     STATUS                Current
35     DESCRIPTION
36         "The lowest carrier of the base station operating band.
37
38         This is an extended carrier number
39         that identifies the lowest carrier of the bandwidth
40         to which the base station is tuned.
41         This value cannot be changed while the Base Station state
42         is Operating.
43
44         (From mibCtl ElementType 52 BSLowestCarrier)
45     Description for mibCtl Type 15 TextType :
46         ASCII or compatible text.
47     Type derived from mibCtl Type 12 OctetType :
48         8 bits of raw opaque data.
49     Derived from basic 8 bit word type.
50     "
51     ::= { _625k-MCInterfaceRF 3 }
52
53
54
55     _625k-MCBCHModuleAddress                    OBJECT-TYPE
56     SYNTAX                INTEGER -- ModuleAddressType
57     MAX-ACCESS            read-only
58     STATUS                Current
59     DESCRIPTION
60         "Which module is handling the broadcast channel.
61
62         (From mibCtl ElementType 57 BCHModuleAddress)
63     Description for mibCtl Type 202 ModuleAddressType :
64         Base station bus slot address.
65
66         Most components of the Base Station for which data can
67         be obtained are identified by a ModuleAddressType address
68         and possibly a subsidiary address.

```

```

1         [Limits: 0 7 ]
2         Type derived from mibCtl Type 14 Unsigned32Type :
3         32 bit unsigned integer.
4         Type derived from mibCtl Type 11 Word32Type :
5         32 bits of raw opaque data.
6         Derived from basic 32 bit word type.
7         "
8         ::= { _625k-MCInterfaceRF 4 }
9
10
11
12         _625k-MCBCHCarrierNumber                OBJECT-TYPE
13         SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
14         MAX-ACCESS      read-write
15         STATUS          Current
16         DESCRIPTION
17             "Number Of BCH Carrier.
18
19             (From mibCtl ElementType 58 BCHCarrierNumber)
20             Description for mibCtl Type 15 TextType :
21             ASCII or compatible text.
22             Type derived from mibCtl Type 12 OctetType :
23             8 bits of raw opaque data.
24             Derived from basic 8 bit word type.
25             "
26         ::= { _625k-MCInterfaceRF 5 }
27
28
29
30         _625k-MCRACHCarrierMask                OBJECT-TYPE
31         SYNTAX          Unsigned32 -- Unsigned32Type
32         MAX-ACCESS      read-write
33         STATUS          Current
34         DESCRIPTION
35             "RACH carrier mask.
36
37
38             (From mibCtl ElementType 73 RACHCarrierMask)
39             Description for mibCtl Type 14 Unsigned32Type :
40             32 bit unsigned integer.
41             Type derived from mibCtl Type 11 Word32Type :
42             32 bits of raw opaque data.
43             Derived from basic 32 bit word type.
44             "
45         ::= { _625k-MCInterfaceRF 6 }
46
47
48
49
50         _625k-MCRACHSlotMask                  OBJECT-TYPE
51         SYNTAX          Unsigned32 -- Unsigned32Type
52         MAX-ACCESS      read-write
53         STATUS          Current
54         DESCRIPTION
55             "RACH slot mask.
56
57
58             (From mibCtl ElementType 72 RACHSlotMask)
59             Description for mibCtl Type 14 Unsigned32Type :
60             32 bit unsigned integer.
61             Type derived from mibCtl Type 11 Word32Type :
62             32 bits of raw opaque data.
63             Derived from basic 32 bit word type.
64             "
65         ::= { _625k-MCInterfaceRF 7 }
66
67
68

```

```

1
2  _625k-MCCalibrationInterval          OBJECT-TYPE
3      SYNTAX          Unsigned32 -- Unsigned32Type
4      MAX-ACCESS      read-write
5      STATUS          Current
6      DESCRIPTION
7          "Calibration interval time.
8
9
10
11         (From mibCtl ElementType 75 CalibrationInterval)
12         Description for mibCtl Type 14 Unsigned32Type :
13             32 bit unsigned integer.
14         Type derived from mibCtl Type 11 Word32Type :
15             32 bits of raw opaque data.
16         Derived from basic 32 bit word type.
17         "
18     ::= { _625k-MCInterfaceRF 8 }
19
20
21
22  _625k-MCSpatialParameter            OBJECT-TYPE
23      SYNTAX          Unsigned32 -- Unsigned32Type
24      MAX-ACCESS      read-write
25      STATUS          Current
26      DESCRIPTION
27          "Spatial parameter.
28
29
30
31         (From mibCtl ElementType 78 SpatialParameter)
32         Description for mibCtl Type 14 Unsigned32Type :
33             32 bit unsigned integer.
34         Type derived from mibCtl Type 11 Word32Type :
35             32 bits of raw opaque data.
36         Derived from basic 32 bit word type.
37         "
38     ::= { _625k-MCInterfaceRF 9 }
39
40
41
42  _625k-MCCostCalcParameter           OBJECT-TYPE
43      SYNTAX          Unsigned32 -- Unsigned32Type
44      MAX-ACCESS      read-write
45      STATUS          Current
46      DESCRIPTION
47          "Cost calculation parameter.
48
49
50
51         (From mibCtl ElementType 79 CostCalcParameter)
52         Description for mibCtl Type 14 Unsigned32Type :
53             32 bit unsigned integer.
54         Type derived from mibCtl Type 11 Word32Type :
55             32 bits of raw opaque data.
56         Derived from basic 32 bit word type.
57         "
58     ::= { _625k-MCInterfaceRF 10 }
59
60
61
62  _625k-MCBSRegistrationCapacity       OBJECT-TYPE
63      SYNTAX          Unsigned32 -- Unsigned32Type
64      MAX-ACCESS      read-write
65      STATUS          Current
66      DESCRIPTION
67          "Capacity of UT registration on BS.
68

```

```

1
2
3      (From mibCtl ElementType 76 BSRegistrationCapacity)
4      Description for mibCtl Type 14 Unsigned32Type :
5          32 bit unsigned integer.
6      Type derived from mibCtl Type 11 Word32Type :
7          32 bits of raw opaque data.
8      Derived from basic 32 bit word type.
9      "
10     ::= { _625k-MCInterfaceRF 11 }
11
12
13
14     _625k-MCBSRegistrationTimer          OBJECT-TYPE
15     SYNTAX          Unsigned32 -- Unsigned32Type
16     MAX-ACCESS      read-write
17     STATUS          Current
18     DESCRIPTION
19         "Timer of keeping UT registration on BS.
20
21
22
23     (From mibCtl ElementType 77 BSRegistrationTimer)
24     Description for mibCtl Type 14 Unsigned32Type :
25         32 bit unsigned integer.
26     Type derived from mibCtl Type 11 Word32Type :
27         32 bits of raw opaque data.
28     Derived from basic 32 bit word type.
29     "
30     ::= { _625k-MCInterfaceRF 12 }
31
32
33
34     _625k-MCPCHFrequencyHopping          OBJECT-TYPE
35     SYNTAX          INTEGER -- BooleanType
36     MAX-ACCESS      read-write
37     STATUS          Current
38     DESCRIPTION
39         "Propriety of frequency hopping (PCH).
40
41
42
43     (From mibCtl ElementType 70 PCHFrequencyHopping)
44     Description for mibCtl Type 16 BooleanType :
45         Truth value, 0=FALSE, 1=TRUE.
46
47         This is a subset of TriStateType; no UNDEFINED value is provided.
48         [Limits: 0 1 ]
49     Description for mibCtl BooleanType 0 FALSE :
50         False.
51     Description for mibCtl BooleanType 1 TRUE :
52         True.
53     "
54     ::= { _625k-MCInterfaceRF 13 }
55
56
57
58     _625k-MCTCHFrequencyHopping          OBJECT-TYPE
59     SYNTAX          INTEGER -- BooleanType
60     MAX-ACCESS      read-write
61     STATUS          Current
62     DESCRIPTION
63         "Propriety of frequency hopping (TCH).
64
65
66
67     (From mibCtl ElementType 71 TCHFrequencyHopping)
68     Description for mibCtl Type 16 BooleanType :

```

```

1         Truth value, 0=FALSE, 1=TRUE.
2
3         This is a subset of TriStateType; no UNDEFINED value is provided.
4         [Limits: 0 1 ]
5         Description for mibCtl BooleanType 0 FALSE :
6         False.
7         Description for mibCtl BooleanType 1 TRUE :
8         True.
9         "
10        ::= { _625k-MCInterfaceRF 14 }
11
12
13
14
15        _625k-MCRFStatusTable                OBJECT-TYPE
16        SYNTAX SEQUENCE OF _625k-MCRFStatusTableEntry
17        MAX-ACCESS not-accessible
18        STATUS Current
19        DESCRIPTION "RF Status Table"
20        ::= { _625k-MCInterfaceRF 15 }
21
22
23
24        _625k-MCRFStatusTableEntry            OBJECT-TYPE
25        SYNTAX _625k-MCRFStatusTableEntry
26        MAX-ACCESS not-accessible
27        STATUS Current
28        DESCRIPTION ""
29        INDEX { _625k-MCRFStatusTableIndex }
30        ::= { _625k-MCRFStatusTable 1 }
31
32        _625k-MCRFStatusTableEntry ::= SEQUENCE {
33        _625k-MCRFStatusTableIndex            INTEGER, -- MoNerdAddressType
34        _625k-MCBSAirBitRateUpLink           Unsigned32, -- Unsigned32Type
35        _625k-MCBSAirBitRateDownLink        Unsigned32, -- Unsigned32Type
36        _625k-MCBSActiveStream               Unsigned32, -- Unsigned32Type
37        _625k-MCBSActiveRegistration         Unsigned32 -- Unsigned32Type
38        }
39
40
41
42        _625k-MCRFStatusTableIndex            OBJECT-TYPE
43        SYNTAX INTEGER -- MoNerdAddressType
44        MAX-ACCESS read-only
45        STATUS Current
46        DESCRIPTION "
47        Description for mibCtl Type 204 MoNerdAddressType :
48        Base station network component address.
49
50        A network address is a subset of Base Station component addresses,
51        restricted to network components only.
52        Network components interface with a telephony switch or similar.
53        [Limits: 0 1 ]
54        Type derived from mibCtl Type 14 Unsigned32Type :
55        32 bit unsigned integer.
56        Type derived from mibCtl Type 11 Word32Type :
57        32 bits of raw opaque data.
58        Derived from basic 32 bit word type.
59        "
60        ::= { _625k-MCRFStatusTableEntry 1 }
61
62
63
64        _625k-MCBSAirBitRateUpLink            OBJECT-TYPE
65        SYNTAX Unsigned32 -- Unsigned32Type
66        MAX-ACCESS read-only
67        STATUS Current
68        DESCRIPTION

```



```

1         "Radio bit rate of up link per Modem control board.
2
3
4
5         (From mibCtl ElementType 4022 BSAirBitRateUpLink)
6         Description for mibCtl Type 14 Unsigned32Type :
7             32 bit unsigned integer.
8         Type derived from mibCtl Type 11 Word32Type :
9             32 bits of raw opaque data.
10        Derived from basic 32 bit word type.
11        "
12        ::= { _625k-MCRFStatusTableEntry 2 }
13
14
15
16        _625k-MCBSAirBitRateDownLink          OBJECT-TYPE
17        SYNTAX          Unsigned32 -- Unsigned32Type
18        MAX-ACCESS      read-only
19        STATUS          Current
20        DESCRIPTION
21            "Radio bit rate of down link per Modem control board.
22
23
24
25            (From mibCtl ElementType 4023 BSAirBitRateDownLink)
26            Description for mibCtl Type 14 Unsigned32Type :
27                32 bit unsigned integer.
28            Type derived from mibCtl Type 11 Word32Type :
29                32 bits of raw opaque data.
30            Derived from basic 32 bit word type.
31            "
32            ::= { _625k-MCRFStatusTableEntry 3 }
33
34
35
36        _625k-MCBSActiveStream                OBJECT-TYPE
37        SYNTAX          Unsigned32 -- Unsigned32Type
38        MAX-ACCESS      read-only
39        STATUS          Current
40        DESCRIPTION
41            "Number of streams currently connected in a base station.
42
43            Number of active streams.
44
45            (From mibCtl ElementType 4020 BSActiveStream)
46            Description for mibCtl Type 14 Unsigned32Type :
47                32 bit unsigned integer.
48            Type derived from mibCtl Type 11 Word32Type :
49                32 bits of raw opaque data.
50            Derived from basic 32 bit word type.
51            "
52            ::= { _625k-MCRFStatusTableEntry 4 }
53
54
55
56        _625k-MCBSActiveRegistration          OBJECT-TYPE
57        SYNTAX          Unsigned32 -- Unsigned32Type
58        MAX-ACCESS      read-only
59        STATUS          Current
60        DESCRIPTION
61            "Number of registrations currently existed in a base station.
62
63            The call capacity is determined by the available resources in a
64            base station.
65
66            (From mibCtl ElementType 4021 BSActiveRegistration)
67            Description for mibCtl Type 14 Unsigned32Type :
68                32 bit unsigned integer.

```

```

1         Type derived from mibCtl Type 11 Word32Type :
2         32 bits of raw opaque data.
3         Derived from basic 32 bit word type.
4         "
5         ::= { _625k-MCRFStatusTableEntry 5 }
6
7         _625k-MCSysScalars                OBJECT IDENTIFIER
8         -- DESCRIPTION                    "System Scalars"
9         ::= { _625k-MCSystem 4 }
10
11
12
13        _625k-MCBaseStationID              OBJECT-TYPE
14        SYNTAX                            OCTET STRING (SIZE(0..18)) -- TextType X 18
15        MAX-ACCESS                        read-write
16        STATUS                            Current
17        DESCRIPTION
18        "Base Station Identification Code.
19
20        This text string must represent in hexadecimal a 42 bit number
21        to be used as the Base Station Identification Code (BSID).
22        The BSID is used by the base station to identify itself to
23        subscriber units.
24        The BSID of a base station must at a minimum
25        differ from that of any other base station
26        where both would be within radio reception distance
27        of any subscriber unit.
28
29        This cannot be changed while the Base Station state is Operating.
30
31        (From mibCtl ElementType 60 BaseStationID)
32        Description for mibCtl Type 15 TextType :
33        ASCII or compatible text.
34        Type derived from mibCtl Type 12 OctetType :
35        8 bits of raw opaque data.
36        Derived from basic 8 bit word type.
37        "
38        ::= { _625k-MCSysScalars 1 }
39
40
41
42        _625k-MCBaseStationTypeID           OBJECT-TYPE
43        SYNTAX                            OCTET STRING (SIZE(0..20)) -- TextType X 20
44        MAX-ACCESS                        read-write
45        STATUS                            Current
46        DESCRIPTION
47        "Type ID of base station.
48
49
50
51        (From mibCtl ElementType 66 BaseStationTypeID)
52        Description for mibCtl Type 15 TextType :
53        ASCII or compatible text.
54        Type derived from mibCtl Type 12 OctetType :
55        8 bits of raw opaque data.
56        Derived from basic 8 bit word type.
57        "
58        ::= { _625k-MCSysScalars 2 }
59
60
61
62        _625k-MCBaseStationGroupID         OBJECT-TYPE
63        SYNTAX                            OCTET STRING (SIZE(0..20)) -- TextType X 20
64        MAX-ACCESS                        read-write
65        STATUS                            Current
66        DESCRIPTION
67        "Group ID of base station.
68

```

```

1
2
3      (From mibCtl ElementType 67 BaseStationGroupID)
4      Description for mibCtl Type 15 TextType :
5          ASCII or compatible text.
6      Type derived from mibCtl Type 12 OctetType :
7          8 bits of raw opaque data.
8      Derived from basic 8 bit word type.
9      "
10     ::= { _625k-MCSysScalars 3 }
11
12
13
14     _625k-MCBaseStationSubGroupID          OBJECT-TYPE
15     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
16     MAX-ACCESS      read-write
17     STATUS          Current
18     DESCRIPTION
19         "Sub group ID of base station.
20
21
22
23     (From mibCtl ElementType 68 BaseStationSubGroupID)
24     Description for mibCtl Type 15 TextType :
25         ASCII or compatible text.
26     Type derived from mibCtl Type 12 OctetType :
27         8 bits of raw opaque data.
28     Derived from basic 8 bit word type.
29     "
30     ::= { _625k-MCSysScalars 4 }
31
32
33
34     _625k-MCDesiredStateOfBaseStation      OBJECT-TYPE
35     SYNTAX          INTEGER -- ComponentStateType
36     MAX-ACCESS      read-write
37     STATUS          Current
38     DESCRIPTION
39         "Desired state of base station as a whole.
40
41         This indicates the Base Station state desired by the operator.
42         These desired states are currently supported:
43
44         Operating - for normal operation.
45
46         Ready - to avoid taking any new calls.
47         Existing calls will not be terminated except normally or by
48         command from the operator.
49         While existing calls remain, the base station state will
50         remain as Operating.
51
52         This information is permanently stored on the base station.
53
54     (From mibCtl ElementType 42 DesiredStateOfBaseStation)
55     Description for mibCtl Type 71 ComponentStateType :
56         Component operational state.
57
58         A component begins in the Unknown state.
59         If not detected, it enters and remains in the NotPresent state.
60         If detected, it enters the Uninitialized state, from where it
61         may go to the Testing and Initializing states and then to the
62         Standby or Operating state depending upon permissions.
63         Due to loss of permissions or resources, it may revert from
64         the Operating state to the Standby state.
65         Due to failure or loss of permission, it may revert to the
66         Uninitialized state, perhaps by way of the ShuttingDown state
67         depending on the device.
68         From the Uninitialized state it may return to more advanced

```

```

1         states depending upon permissions.
2         In case of a waiting period before (again) initializing,
3         the component is considered to be Initializing.
4
5         Permissions include administrative permissions (from the
6         operator); excessive failure restrictions; etc.
7         Description for mibCtl ComponentStateType 0 Unknown :
8         Component state not known.
9         Description for mibCtl ComponentStateType 1 NotPresent :
10        Component is not present.
11        Description for mibCtl ComponentStateType 2 PowerOff :
12        Component is present but powered off.
13        Description for mibCtl ComponentStateType 3 Uninitialized :
14        Component is present but not in use.
15
16        The power on/off state of the component is not specified in
17        this case.
18        Description for mibCtl ComponentStateType 4 Testing :
19        Component is being tested.
20        Description for mibCtl ComponentStateType 5 Initializing :
21        Component is being initialized.
22        Description for mibCtl ComponentStateType 6 Ready :
23        Component is ready but not operating.
24        Description for mibCtl ComponentStateType 7 Operating :
25        Component is operating for normal use without restriction.
26
27        The component is either in actual use or may be used at any time,
28        without restriction.
29        Description for mibCtl ComponentStateType 8 Abandoned :
30        Component state is not the desired state due to excessive errors.
31
32        The component state is not that desired, and the Base Station
33        software has abandoned attempts to place the component in
34        the desired state.
35        The actual state of the component is undefined.
36        The Base Station software will resume attempting to place the
37        component in the desired state if the appropriate Reinitialize
38        action element is written with the correct value.
39        Also, the software may resume attempts under other conditions,
40        not all of which may be documented.
41        Description for mibCtl ComponentStateType 9 InitialSetUp :
42        Component is initial set up..
43
44        Initial set up state.
45        Description for mibCtl ComponentStateType 10 Degrading :
46        Component is degrading..
47
48        Degrading state.
49        Description for mibCtl ComponentStateType 11 Restriction :
50        Component is restriction..
51
52        Restriction state.
53        "
54        ::= { _625k-MCSysScalars 5 }
55
56
57
58        _625k-MCTypeOfReboot          OBJECT-TYPE
59            SYNTAX          INTEGER -- RebootType
60            MAX-ACCESS      read-write
61            STATUS          Current
62            DESCRIPTION
63                "Type of reboot for base station.
64
65
66                (From mibCtl ElementType 63 TypeOfReboot)
67            Description for mibCtl Type 250 RebootType :
68

```

```

1       Reboot Type.
2       Description for mibCtl RebootType 0 Force :
3         Force mode.
4       Description for mibCtl RebootType 1 Graceful :
5         Graceful mode.
6       "
7       ::= { _625k-MCSysScalars 6 }
8
9
10
11      _625k-MCBaseStationRebootTime          OBJECT-TYPE
12      SYNTAX          Gauge -- AbsoluteTimeType
13      MAX-ACCESS      read-write
14      STATUS          Current
15      DESCRIPTION
16        "Time of base station reboot.
17
18        This is the base station reboot time (GPS time).
19
20        (From mibCtl ElementType 62 BaseStationRebootTime)
21      Description for mibCtl Type 801 AbsoluteTimeType :
22        Absolute time in GPS seconds.
23
24        GPS (Global Positioning System) time in seconds since Jan. 6,
25      1980.
26
27        Note that this differs from UTC (in addition to a possible
28        offset due to starting time) due to leap seconds; see
29        the GpsLeapSecond element.
30      Type derived from mibCtl Type 18 Gauge32Type :
31        32 bits of Gauge data.
32      Derived from basic 32 bit word type.
33      "
34      ::= { _625k-MCSysScalars 7 }
35
36
37      _625k-MCTypeOfBSDiagnosis              OBJECT-TYPE
38      SYNTAX          INTEGER -- DiagnosisType
39      MAX-ACCESS      read-write
40      STATUS          Current
41      DESCRIPTION
42        "Type of diagnosis for base station.
43
44
45
46        (From mibCtl ElementType 64 TypeOfBSDiagnosis)
47      Description for mibCtl Type 251 DiagnosisType :
48        Diagnosis Type.
49      Type derived from mibCtl Type 14 Unsigned32Type :
50        32 bit unsigned integer.
51      Type derived from mibCtl Type 11 Word32Type :
52        32 bits of raw opaque data.
53      Derived from basic 32 bit word type.
54      "
55      ::= { _625k-MCSysScalars 8 }
56
57
58
59      _625k-MCBSdiagnosisStatus              OBJECT-TYPE
60      SYNTAX          INTEGER -- DiagnosisStatusType
61      MAX-ACCESS      read-only
62      STATUS          Current
63      DESCRIPTION
64        "Diagnosis status for base station.
65
66
67
68        (From mibCtl ElementType 370 BSDiagnosisStatus)

```

```

1      Description for mibCtl Type 252 DiagnosisStatusType :
2          Diagnosis status Type.
3      Type derived from mibCtl Type 14 Unsigned32Type :
4          32 bit unsigned integer.
5      Type derived from mibCtl Type 11 Word32Type :
6          32 bits of raw opaque data.
7      Derived from basic 32 bit word type.
8      "
9      ::= { _625k-MCSysScalars 9 }
10
11
12
13     _625k-MCBSDiagnosisFailReason          OBJECT-TYPE
14     SYNTAX          INTEGER -- DiagFailReasonType
15     MAX-ACCESS      read-only
16     STATUS          Current
17     DESCRIPTION
18         "Diagnosis fail reason for base station.
19
20
21
22         (From mibCtl ElementType 371 BSDiagnosisFailReason)
23     Description for mibCtl Type 253 DiagFailReasonType :
24         Diagnosis fail reason Type.
25     Description for mibCtl DiagFailReasonType 1 PDSNPing :
26         Diagnosis fail reason is PDSN Ping.
27     Description for mibCtl DiagFailReasonType 2 Calibration :
28         Diagnosis fail reason is Calibration.
29     Description for mibCtl DiagFailReasonType 3 AntPath :
30         Diagnosis fail reason is TRx Antenna Path.
31     Description for mibCtl DiagFailReasonType 4 LOAlive :
32         Diagnosis fail reason is Local Oscilator DSP Alive.
33     Description for mibCtl DiagFailReasonType 5 GCLoopBack :
34         Diagnosis fail reason is GCLoopBack.
35     Description for mibCtl DiagFailReasonType 6 SlaveNM :
36         Diagnosis fail reason is Slave Modem control board.
37     Description for mibCtl DiagFailReasonType 7 GPSAnt :
38         Diagnosis fail reason is GPS Antenna.
39     Description for mibCtl DiagFailReasonType 8 SlotDSP :
40         Diagnosis fail reason is Modem control board DSP Alive.
41     Description for mibCtl DiagFailReasonType 9 ATMAlive :
42         Diagnosis fail reason is ATM Alive.
43     Description for mibCtl DiagFailReasonType 96 UndefineName :
44         Diagnosis fail reason is Undefine Diag Name.
45     Description for mibCtl DiagFailReasonType 97 TimeOut :
46         Diagnosis fail reason is Time Out.
47     Description for mibCtl DiagFailReasonType 98 InvalidStateExec :
48         Diagnosis fail reason is Invalid State Execute.
49     Description for mibCtl DiagFailReasonType 99 ExecFail :
50         Diagnosis fail reason is Execute Fail.
51     "
52     ::= { _625k-MCSysScalars 10 }
53
54
55
56     _625k-MCDiskDbUpdateSequence          OBJECT-TYPE
57     SYNTAX          Gauge -- Gauge32Type
58     MAX-ACCESS      read-only
59     STATUS          Current
60     DESCRIPTION
61         "Base station Flach update sequence number.
62
63         This number is incremented on disk every time any other
64         database element is actually changed on flash.
65         It is not incremented on redundant sets.
66         This number may also be set to a desired value.
67
68         (From mibCtl ElementType 6 DiskDbUpdateSequence)

```

```

1      Description for mibCtl Type 18 Gauge32Type :
2          32 bits of Gauge data.
3      Derived from basic 32 bit word type.
4      "
5      ::= { _625k-MCSysScalars 11 }
6
7
8
9      _625k-MCStateOfBaseStation          OBJECT-TYPE
10     SYNTAX          INTEGER -- ComponentStateType
11     MAX-ACCESS      read-only
12     STATUS          Current
13     DESCRIPTION
14         "State of base station as a whole.
15
16         This will not have values of Unknown or PowerOff since the
17         base station would be unable to report such values.
18
19         When sufficiently initialized, the state will be Operating if
20         accepting new calls (according to the desired state of the
21         base station) or continuing ongoing calls;
22         or Ready if the desired state is Ready and there are no
23         ongoing calls.
24
25         (From mibCtl ElementType 41 StateOfBaseStation)
26     Description for mibCtl Type 71 ComponentStateType :
27         Component operational state.
28
29         A component begins in the Unknown state.
30         If not detected, it enters and remains in the NotPresent state.
31         If detected, it enters the Uninitialized state, from where it
32         may go to the Testing and Initializing states and then to the
33         Standby or Operating state depending upon permissions.
34         Due to loss of permissions or resources, it may revert from
35         the Operating state to the Standby state.
36         Due to failure or loss of permission, it may revert to the
37         Uninitialized state, perhaps by way of the ShuttingDown state
38         depending on the device.
39         From the Uninitialized state it may return to more advanced
40         states depending upon permissions.
41         In case of a waiting period before (again) initializing,
42         the component is considered to be Initializing.
43
44         Permissions include administrative permissions (from the
45         operator); excessive failure restrictions; etc.
46     Description for mibCtl ComponentStateType 0 Unknown :
47         Component state not known.
48     Description for mibCtl ComponentStateType 1 NotPresent :
49         Component is not present.
50     Description for mibCtl ComponentStateType 2 PowerOff :
51         Component is present but powered off.
52     Description for mibCtl ComponentStateType 3 Uninitialized :
53         Component is present but not in use.
54
55         The power on/off state of the component is not specified in
56         this case.
57     Description for mibCtl ComponentStateType 4 Testing :
58         Component is being tested.
59     Description for mibCtl ComponentStateType 5 Initializing :
60         Component is being initialized.
61     Description for mibCtl ComponentStateType 6 Ready :
62         Component is ready but not operating.
63     Description for mibCtl ComponentStateType 7 Operating :
64         Component is operating for normal use without restriction.
65
66         The component is either in actual use or may be used at any time,
67         without restriction.
68     Description for mibCtl ComponentStateType 8 Abandoned :

```

```

1         Component state is not the desired state due to excessive errors.
2
3         The component state is not that desired, and the Base Station
4         software has abandoned attempts to place the component in
5         the desired state.
6         The actual state of the component is undefined.
7         The Base Station software will resume attempting to place the
8         component in the desired state if the appropriate Reinitialize
9         action element is written with the correct value.
10        Also, the software may resume attempts under other conditions,
11        not all of which may be documented.
12        Description for mibCtl ComponentStateType 9 InitialSetUp :
13        Component is initial set up..
14
15        Initial set up state.
16        Description for mibCtl ComponentStateType 10 Degrading :
17        Component is degrading..
18
19        Degrading state.
20        Description for mibCtl ComponentStateType 11 Restriction :
21        Component is restriction..
22
23        Restriction state.
24        "
25        ::= { _625k-MCSysScalars 12 }
26
27
28
29        _625k-MCBSTotalIndication          OBJECT-TYPE
30        SYNTAX          INTEGER -- IndicationType
31        MAX-ACCESS      read-only
32        STATUS          Current
33        DESCRIPTION
34        "Status of BS total indicator.
35
36        In the current implementation,
37        this indication is set to the value On by BS.
38
39        (From mibCtl ElementType 525 BSTotalIndication)
40        Description for mibCtl Type 230 IndicationType :
41        Hardware indication status (LEDs).
42
43
44        Description for mibCtl IndicationType 0 Off :
45        Off.
46        Description for mibCtl IndicationType 1 Amber :
47        Amber.
48        Description for mibCtl IndicationType 2 Red :
49        Red.
50        Description for mibCtl IndicationType 3 Green :
51        Green.
52        Description for mibCtl IndicationType 4 NotPresent :
53        Not present.
54        "
55        ::= { _625k-MCSysScalars 13 }
56
57
58
59        _625k-MCMasterAddress              OBJECT-TYPE
60        SYNTAX          INTEGER -- ModuleAddressType
61        MAX-ACCESS      read-only
62        STATUS          Current
63        DESCRIPTION
64        "Bus slot address of master Modem control board.
65
66        This indicates which Modem control board is master
67        of the base station.
68

```



```

1      (From mibCtl ElementType 32 MasterAddress)
2      Description for mibCtl Type 202 ModuleAddressType :
3          Base station bus slot address.
4
5          Most components of the Base Station for which data can
6          be obtained are identified by a ModuleAddressType address
7          and possibly a subsidiary address.
8          [Limits: 0 7 ]
9      Type derived from mibCtl Type 14 Unsigned32Type :
10         32 bit unsigned integer.
11      Type derived from mibCtl Type 11 Word32Type :
12         32 bits of raw opaque data.
13      Derived from basic 32 bit word type.
14      "
15      ::= { _625k-MCSysScalars 14 }
16
17
18
19      _625k-MCBSManufactureID          OBJECT-TYPE
20      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
21      MAX-ACCESS      read-only
22      STATUS          Current
23      DESCRIPTION
24          "Base Station manufacture identification number.
25
26          The manufacture identification assigned by Vendor
27
28          (From mibCtl ElementType 201 BSManufactureID)
29      Description for mibCtl Type 15 TextType :
30          ASCII or compatible text.
31      Type derived from mibCtl Type 12 OctetType :
32          8 bits of raw opaque data.
33      Derived from basic 8 bit word type.
34      "
35      ::= { _625k-MCSysScalars 15 }
36
37
38
39      _625k-MCBSSerialNumber          OBJECT-TYPE
40      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
41      MAX-ACCESS      read-only
42      STATUS          Current
43      DESCRIPTION
44          "Base Station serial number.
45
46          This is the character serial number of the base station.
47          This serial number will be unique among all base stations
48          of this type regardless of manufacturer.
49
50          (From mibCtl ElementType 203 BSSerialNumber)
51      Description for mibCtl Type 15 TextType :
52          ASCII or compatible text.
53      Type derived from mibCtl Type 12 OctetType :
54          8 bits of raw opaque data.
55      Derived from basic 8 bit word type.
56      "
57      ::= { _625k-MCSysScalars 16 }
58
59
60
61      _625k-MCDiagnosisBaseStation    OBJECT-TYPE
62      SYNTAX          INTEGER -- BooleanType
63      MAX-ACCESS      read-write -- REALLY: write-only
64      STATUS          Current
65      DESCRIPTION
66          "Diagnosis base station.
67
68          This is a write-only element; only a value of TRUE is valid.

```

```

1
2      (From mibCtl ElementType 47 DiagnosisBaseStation)
3      Description for mibCtl Type 16 BooleanType :
4          Truth value, 0=FALSE, 1=TRUE.
5
6          This is a subset of TriStateType; no UNDEFINED value is provided.
7          [Limits: 0 1 ]
8      Description for mibCtl BooleanType 0 FALSE :
9          False.
10     Description for mibCtl BooleanType 1 TRUE :
11         True.
12     "
13     ::= { _625k-MCSysScalars 17 }
14
15
16
17     _625k-MCRebootBaseStation                OBJECT-TYPE
18     SYNTAX                INTEGER -- BooleanType
19     MAX-ACCESS            read-write -- REALLY: write-only
20     STATUS                Current
21     DESCRIPTION
22         "Reboot base station.
23
24         This is a write-only element; only a value of TRUE is valid.
25         All existing calls will be terminated abruptly.
26         All components of the base station will be reinitialized
27         according to the permanent contents of the Base Station database.
28         The base station may be incommunicado for a period of time.
29
30         The reinitialization may be delayed by a few seconds to allow
31         for a clean shutdown.
32
33         (From mibCtl ElementType 44 RebootBaseStation)
34     Description for mibCtl Type 16 BooleanType :
35         Truth value, 0=FALSE, 1=TRUE.
36
37         This is a subset of TriStateType; no UNDEFINED value is provided.
38         [Limits: 0 1 ]
39     Description for mibCtl BooleanType 0 FALSE :
40         False.
41     Description for mibCtl BooleanType 1 TRUE :
42         True.
43     "
44     ::= { _625k-MCSysScalars 18 }
45
46
47
48     _625k-MCBSModelNumber                    OBJECT-TYPE
49     SYNTAX                OCTET STRING (SIZE(0..20)) -- TextType X 20
50     MAX-ACCESS            read-only
51     STATUS                Current
52     DESCRIPTION
53         "Base Station model number.
54
55         Base Station model number
56
57         (From mibCtl ElementType 204 BSModelNumber)
58     Description for mibCtl Type 15 TextType :
59         ASCII or compatible text.
60     Type derived from mibCtl Type 12 OctetType :
61         8 bits of raw opaque data.
62     Derived from basic 8 bit word type.
63     "
64     ::= { _625k-MCSysScalars 19 }
65
66
67
68     _625k-MCBSManufactureDate                OBJECT-TYPE

```

```

1      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
2      MAX-ACCESS      read-only
3      STATUS          Current
4      DESCRIPTION
5          "Base Station manufacture date.
6
7          Base Station manufacture date
8
9          (From mibCtl ElementType 205 BSManufactureDate)
10         Description for mibCtl Type 15 TextType :
11             ASCII or compatible text.
12         Type derived from mibCtl Type 12 OctetType :
13             8 bits of raw opaque data.
14         Derived from basic 8 bit word type.
15         "
16         ::= { _625k-MCSysScalars 20 }
17
18
19
20     _625k-MCBSHardwareRevision          OBJECT-TYPE
21     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
22     MAX-ACCESS      read-only
23     STATUS          Current
24     DESCRIPTION
25         "Base Station hardware revision.
26
27         Base Station hardware revision
28
29         (From mibCtl ElementType 206 BSHardwareRevision)
30         Description for mibCtl Type 15 TextType :
31             ASCII or compatible text.
32         Type derived from mibCtl Type 12 OctetType :
33             8 bits of raw opaque data.
34         Derived from basic 8 bit word type.
35         "
36         ::= { _625k-MCSysScalars 21 }
37
38
39
40     _625k-MCMiscComponents              OBJECT IDENTIFIER
41     -- DESCRIPTION          "Miscellaneous Component"
42     ::= { IEEE802dot20-625k-MC-MIB 2 }
43
44
45
46     _625k-MCAntenna                    OBJECT IDENTIFIER
47     -- DESCRIPTION          "Antenna"
48     ::= { _625k-MCMiscComponents 1 }
49
50
51
52     _625k-MCAntennaTable                OBJECT-TYPE
53     SYNTAX SEQUENCE OF _625k-MCAntennaTableEntry
54     MAX-ACCESS      not-accessible
55     STATUS          Current
56     DESCRIPTION    "Antenna Table"
57     ::= { _625k-MCAntenna 1 }
58
59
60
61     _625k-MCAntennaTableEntry          OBJECT-TYPE
62     SYNTAX          _625k-MCAntennaTableEntry
63     MAX-ACCESS      not-accessible
64     STATUS          Current
65     DESCRIPTION    ""
66     INDEX { _625k-MCAntennaTableIndex }
67     ::= { _625k-MCAntennaTable 1 }
68

```

```

1  _625k-MCAntennaTableEntry ::= SEQUENCE {
2      _625k-MCAntennaTableIndex          INTEGER, -- AntennaAddressType
3      _625k-MCStateOfAntenna            INTEGER -- ComponentStateType
4      }
5
6
7
8  _625k-MCAntennaTableIndex              OBJECT-TYPE
9      SYNTAX          INTEGER -- AntennaAddressType
10     MAX-ACCESS      read-only
11     STATUS          Current
12     DESCRIPTION    "
13         Description for mibCtl Type 210 AntennaAddressType :
14             Component antenna address.
15             [Limits: 0 11 ]
16         Type derived from mibCtl Type 14 Unsigned32Type :
17             32 bit unsigned integer.
18         Type derived from mibCtl Type 11 Word32Type :
19             32 bits of raw opaque data.
20         Derived from basic 32 bit word type.
21     "
22     ::= { _625k-MCAntennaTableEntry 1 }
23
24
25
26  _625k-MCStateOfAntenna                  OBJECT-TYPE
27     SYNTAX          INTEGER -- ComponentStateType
28     MAX-ACCESS      read-only
29     STATUS          Current
30     DESCRIPTION    "
31         "State of Antenna as a whole.
32
33
34         (From mibCtl ElementType 211 StateOfAntenna)
35         Description for mibCtl Type 71 ComponentStateType :
36             Component operational state.
37
38             A component begins in the Unknown state.
39             If not detected, it enters and remains in the NotPresent state.
40             If detected, it enters the Uninitialized state, from where it
41             may go to the Testing and Initializing states and then to the
42             Standby or Operating state depending upon permissions.
43             Due to loss of permissions or resources, it may revert from
44             the Operating state to the Standby state.
45             Due to failure or loss of permission, it may revert to the
46             Uninitialized state, perhaps by way of the ShuttingDown state
47             depending on the device.
48             From the Uninitialized state it may return to more advanced
49             states depending upon permissions.
50             In case of a waiting period before (again) initializing,
51             the component is considered to be Initializing.
52
53             Permissions include administrative permissions (from the
54             operator); excessive failure restrictions; etc.
55         Description for mibCtl ComponentStateType 0 Unknown :
56             Component state not known.
57         Description for mibCtl ComponentStateType 1 NotPresent :
58             Component is not present.
59         Description for mibCtl ComponentStateType 2 PowerOff :
60             Component is present but powered off.
61         Description for mibCtl ComponentStateType 3 Uninitialized :
62             Component is present but not in use.
63
64             The power on/off state of the component is not specified in
65             this case.
66         Description for mibCtl ComponentStateType 4 Testing :
67             Component is being tested.
68         Description for mibCtl ComponentStateType 5 Initializing :

```

```

1         Component is being initialized.
2     Description for mibCtl ComponentStateType 6 Ready :
3         Component is ready but not operating.
4     Description for mibCtl ComponentStateType 7 Operating :
5         Component is operating for normal use without restriction.
6
7         The component is either in actual use or may be used at any time,
8         without restriction.
9     Description for mibCtl ComponentStateType 8 Abandoned :
10        Component state is not the desired state due to excessive errors.
11
12        The component state is not that desired, and the Base Station
13        software has abandoned attempts to place the component in
14        the desired state.
15        The actual state of the component is undefined.
16        The Base Station software will resume attempting to place the
17        component in the desired state if the appropriate Reinitialize
18        action element is written with the correct value.
19        Also, the software may resume attempts under other conditions,
20        not all of which may be documented.
21     Description for mibCtl ComponentStateType 9 InitialSetUp :
22        Component is initial set up..
23
24        Initial set up state.
25     Description for mibCtl ComponentStateType 10 Degrading :
26        Component is degrading..
27
28        Degrading state.
29     Description for mibCtl ComponentStateType 11 Restriction :
30        Component is restriction..
31
32        Restriction state.
33     "
34     ::= { _625k-MCAntennaTableEntry 2 }
35
36
37
38     _625k-MCBSTemperatures                OBJECT IDENTIFIER
39     -- DESCRIPTION            "BS Temperature"
40     ::= { _625k-MCMiscComponents 3 }
41
42
43
44     _625k-MCBSTemperature                OBJECT-TYPE
45     SYNTAX                    OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
46     MAX-ACCESS                read-only
47     STATUS                    Current
48     DESCRIPTION
49         "The temperature of Base station (degrees Celsius).
50
51         The latest recorded temperature of a given BS.
52
53         (From mibCtl ElementType 536 BSTemperature)
54     Description for mibCtl Type 807 DegreesCelsiusType :
55         Temperature in degrees Celsius.
56     Type derived from mibCtl Type 15 TextType :
57         ASCII or compatible text.
58     Type derived from mibCtl Type 12 OctetType :
59         8 bits of raw opaque data.
60     Derived from basic 8 bit word type.
61     "
62     ::= { _625k-MCBSTemperatures 1 }
63
64
65
66     _625k-MCCableInfo                    OBJECT IDENTIFIER
67     -- DESCRIPTION            "Cable Info"
68     ::= { _625k-MCMiscComponents 4 }

```

```

1
2
3
4  _625k-MCCableLossValueForLoCal          OBJECT-TYPE
5  SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
6  MAX-ACCESS      read-only
7  STATUS          Current
8  DESCRIPTION
9      "Value of cable loss for Local Oscilator.
10
11     Value of cable loss.
12
13     (From mibCtl ElementType 801 CableLossValueForLoCal)
14     Description for mibCtl Type 15 TextType :
15         ASCII or compatible text.
16     Type derived from mibCtl Type 12 OctetType :
17         8 bits of raw opaque data.
18     Derived from basic 8 bit word type.
19     "
20     ::= { _625k-MCCableInfo 1 }
21
22
23
24
25  _625k-MCAntCableTable                    OBJECT-TYPE
26  SYNTAX SEQUENCE OF _625k-MCAntCableTableEntry
27  MAX-ACCESS      not-accessible
28  STATUS          Current
29  DESCRIPTION     "Antenna Cable"
30  ::= { _625k-MCCableInfo 2 }
31
32
33
34  _625k-MCAntCableTableEntry                OBJECT-TYPE
35  SYNTAX          _625k-MCAntCableTableEntry
36  MAX-ACCESS      not-accessible
37  STATUS          Current
38  DESCRIPTION     ""
39  INDEX { _625k-MCAntCableTableIndex }
40  ::= { _625k-MCAntCableTable 1 }
41
42  _625k-MCAntCableTableEntry ::= SEQUENCE {
43      _625k-MCAntCableTableIndex          INTEGER, -- AntennaAddressType
44      _625k-MCCableLossValueForAntenna    OCTET STRING (SIZE(0..20)) --
45      TextType X 20
46  }
47
48
49
50  _625k-MCAntCableTableIndex                OBJECT-TYPE
51  SYNTAX          INTEGER -- AntennaAddressType
52  MAX-ACCESS      read-only
53  STATUS          Current
54  DESCRIPTION     "
55      Description for mibCtl Type 210 AntennaAddressType :
56      Component antenna address.
57      [Limits: 0 11 ]
58      Type derived from mibCtl Type 14 Unsigned32Type :
59          32 bit unsigned integer.
60      Type derived from mibCtl Type 11 Word32Type :
61          32 bits of raw opaque data.
62      Derived from basic 32 bit word type.
63      "
64      ::= { _625k-MCAntCableTableEntry 1 }
65
66
67
68  _625k-MCCableLossValueForAntenna          OBJECT-TYPE

```

```

1      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
2      MAX-ACCESS      read-only
3      STATUS          Current
4      DESCRIPTION
5          "Value of cable loss for Antenna.
6
7          Value of cable loss.
8
9          (From mibCtl ElementType 802 CableLossValueForAntenna)
10         Description for mibCtl Type 15 TextType :
11             ASCII or compatible text.
12         Type derived from mibCtl Type 12 OctetType :
13             8 bits of raw opaque data.
14         Derived from basic 8 bit word type.
15         "
16         ::= { _625k-MCAntCableTableEntry 2 }
17
18
19
20
21     _625k-MCGPSCableTable          OBJECT-TYPE
22     SYNTAX SEQUENCE OF _625k-MCGPSCableTableEntry
23     MAX-ACCESS      not-accessible
24     STATUS          Current
25     DESCRIPTION    "GPS Cable"
26     ::= { _625k-MCCableInfo 10 }
27
28
29
30     _625k-MCGPSCableTableEntry     OBJECT-TYPE
31     SYNTAX          _625k-MCGPSCableTableEntry
32     MAX-ACCESS      not-accessible
33     STATUS          Current
34     DESCRIPTION    ""
35     INDEX { _625k-MCGPSCableTableIndex }
36     ::= { _625k-MCGPSCableTable 1 }
37
38     _625k-MCGPSCableTableEntry ::= SEQUENCE {
39         _625k-MCGPSCableTableIndex    INTEGER, -- GpsAddressType
40         _625k-MCCableLengthForGps     OCTET STRING (SIZE(0..20)) --
41         TextType X 20
42     }
43
44
45
46     _625k-MCGPSCableTableIndex     OBJECT-TYPE
47     SYNTAX          INTEGER -- GpsAddressType
48     MAX-ACCESS      read-only
49     STATUS          Current
50     DESCRIPTION    "
51         Description for mibCtl Type 209 GpsAddressType :
52         Base station GPS component address.
53
54
55         [Limits: 0 1 ]
56         Type derived from mibCtl Type 14 Unsigned32Type :
57             32 bit unsigned integer.
58         Type derived from mibCtl Type 11 Word32Type :
59             32 bits of raw opaque data.
60         Derived from basic 32 bit word type.
61         "
62         ::= { _625k-MCGPSCableTableEntry 1 }
63
64
65
66     _625k-MCCableLengthForGps      OBJECT-TYPE
67     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
68     MAX-ACCESS      read-only

```

```

1      STATUS          Current
2      DESCRIPTION
3          "Cable length for Gps.
4
5          This cable is used for calibration.
6
7          (From mibCtl ElementType 803 CableLengthForGps)
8          Description for mibCtl Type 15 TextType :
9              ASCII or compatible text.
10         Type derived from mibCtl Type 12 OctetType :
11             8 bits of raw opaque data.
12         Derived from basic 8 bit word type.
13         "
14         ::= { _625k-MCGPSCableTableEntry 2 }
15
16
17
18     _625k-MCGPS          OBJECT IDENTIFIER
19         -- DESCRIPTION          "GPS"
20         ::= { _625k-MCMiscComponents 6 }
21
22
23
24     _625k-MCGPSTable    OBJECT-TYPE
25         SYNTAX SEQUENCE OF _625k-MCGPSTableEntry
26         MAX-ACCESS      not-accessible
27         STATUS          Current
28         DESCRIPTION    "GPS Table"
29         ::= { _625k-MCGPS 1 }
30
31
32
33     _625k-MCGPSTableEntry OBJECT-TYPE
34         SYNTAX          _625k-MCGPSTableEntry
35         MAX-ACCESS      not-accessible
36         STATUS          Current
37         DESCRIPTION    ""
38         INDEX           { _625k-MCGPSTableIndex }
39         ::= { _625k-MCGPSTable 1 }
40
41     _625k-MCGPSTableEntry ::= SEQUENCE {
42         _625k-MCGPSTableIndex      INTEGER, -- GpsAddressType
43         _625k-MCStateOfGps         INTEGER, -- ComponentStateType
44         _625k-MCGpsNumberOfSatelliteSeen Unsigned32, -- Unsigned32Type
45         _625k-MCGpsIndication      INTEGER, -- IndicationType
46         _625k-MCGpsSerialNumber    OCTET STRING (SIZE(0..20)) --
47         TextType X 20
48     }
49
50
51
52     _625k-MCGPSTableIndex OBJECT-TYPE
53         SYNTAX          INTEGER -- GpsAddressType
54         MAX-ACCESS      read-only
55         STATUS          Current
56         DESCRIPTION    "
57             Description for mibCtl Type 209 GpsAddressType :
58                 Base station GPS component address.
59
60
61                 [Limits: 0 1 ]
62             Type derived from mibCtl Type 14 Unsigned32Type :
63                 32 bit unsigned integer.
64             Type derived from mibCtl Type 11 Word32Type :
65                 32 bits of raw opaque data.
66             Derived from basic 32 bit word type.
67         "
68         ::= { _625k-MCGPSTableEntry 1 }

```



```

1
2
3
4  _625k-MCStateOfGps                                OBJECT-TYPE
5      SYNTAX                                         INTEGER -- ComponentStateType
6      MAX-ACCESS                                     read-only
7      STATUS                                         Current
8      DESCRIPTION
9          "GPS state.
10
11         The state of the GPS (Global Positioning System)
12         on the active local oscillator unit
13
14         (From mibCtl ElementType 420 StateOfGps)
15         Description for mibCtl Type 71 ComponentStateType :
16             Component operational state.
17
18         A component begins in the Unknown state.
19         If not detected, it enters and remains in the NotPresent state.
20         If detected, it enters the Uninitialized state, from where it
21         may go to the Testing and Initializing states and then to the
22         Standby or Operating state depending upon permissions.
23         Due to loss of permissions or resources, it may revert from
24         the Operating state to the Standby state.
25         Due to failure or loss of permission, it may revert to the
26         Uninitialized state, perhaps by way of the ShuttingDown state
27         depending on the device.
28         From the Uninitialized state it may return to more advanced
29         states depending upon permissions.
30         In case of a waiting period before (again) initializing,
31         the component is considered to be Initializing.
32
33         Permissions include administrative permissions (from the
34         operator); excessive failure restrictions; etc.
35         Description for mibCtl ComponentStateType 0 Unknown :
36             Component state not known.
37         Description for mibCtl ComponentStateType 1 NotPresent :
38             Component is not present.
39         Description for mibCtl ComponentStateType 2 PowerOff :
40             Component is present but powered off.
41         Description for mibCtl ComponentStateType 3 Uninitialized :
42             Component is present but not in use.
43
44         The power on/off state of the component is not specified in
45         this case.
46         Description for mibCtl ComponentStateType 4 Testing :
47             Component is being tested.
48         Description for mibCtl ComponentStateType 5 Initializing :
49             Component is being initialized.
50         Description for mibCtl ComponentStateType 6 Ready :
51             Component is ready but not operating.
52         Description for mibCtl ComponentStateType 7 Operating :
53             Component is operating for normal use without restriction.
54
55         The component is either in actual use or may be used at any time,
56         without restriction.
57         Description for mibCtl ComponentStateType 8 Abandoned :
58             Component state is not the desired state due to excessive errors.
59
60         The component state is not that desired, and the Base Station
61         software has abandoned attempts to place the component in
62         the desired state.
63         The actual state of the component is undefined.
64         The Base Station software will resume attempting to place the
65         component in the desired state if the appropriate Reinitialize
66         action element is written with the correct value.
67         Also, the software may resume attempts under other conditions,
68         not all of which may be documented.

```

```

1      Description for mibCtl ComponentStateType 9 InitialSetUp :
2          Component is initial set up..
3
4          Initial set up state.
5      Description for mibCtl ComponentStateType 10 Degrading :
6          Component is degrading..
7
8          Degrading state.
9      Description for mibCtl ComponentStateType 11 Restriction :
10         Component is restriction..
11
12         Restriction state.
13     "
14     ::= { _625k-MCGPSTableEntry 2 }
15
16
17
18     _625k-MCGpsNumberOfSatelliteSeen          OBJECT-TYPE
19     SYNTAX          Unsigned32 -- Unsigned32Type
20     MAX-ACCESS      read-only
21     STATUS          Current
22     DESCRIPTION
23         "Number of satellites seen by GPS.
24
25         The number of satellites seen by the GPS (Global Positioning System)
26         on the active local oscillator unit
27
28         (From mibCtl ElementType 421 GpsNumberOfSatelliteSeen)
29     Description for mibCtl Type 14 Unsigned32Type :
30         32 bit unsigned integer.
31     Type derived from mibCtl Type 11 Word32Type :
32         32 bits of raw opaque data.
33     Derived from basic 32 bit word type.
34     "
35     ::= { _625k-MCGPSTableEntry 3 }
36
37
38
39     _625k-MCGpsIndication                      OBJECT-TYPE
40     SYNTAX          INTEGER -- IndicationType
41     MAX-ACCESS      read-only
42     STATUS          Current
43     DESCRIPTION
44         "Status of GPS indicator.
45
46
47         (From mibCtl ElementType 530 GpsIndication)
48     Description for mibCtl Type 230 IndicationType :
49         Hardware indication status (LEDs).
50
51
52
53     Description for mibCtl IndicationType 0 Off :
54         Off.
55     Description for mibCtl IndicationType 1 Amber :
56         Amber.
57     Description for mibCtl IndicationType 2 Red :
58         Red.
59     Description for mibCtl IndicationType 3 Green :
60         Green.
61     Description for mibCtl IndicationType 4 NotPresent :
62         Not present.
63     "
64     ::= { _625k-MCGPSTableEntry 4 }
65
66
67
68     _625k-MCGpsSerialNumber                    OBJECT-TYPE

```

```

1      SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
2      MAX-ACCESS      read-only
3      STATUS          Current
4      DESCRIPTION
5          "GPS serial number text.
6
7          Factory set uniquely for each component.
8
9          (From mibCtl ElementType 570 GpsSerialNumber)
10         Description for mibCtl Type 15 TextType :
11             ASCII or compatible text.
12         Type derived from mibCtl Type 12 OctetType :
13             8 bits of raw opaque data.
14         Derived from basic 8 bit word type.
15         "
16         ::= { _625k-MCGPSTableEntry 5 }
17
18
19
20     _625k-MCPowerAmplifier          OBJECT IDENTIFIER
21     -- DESCRIPTION          "Power Amplifier"
22     ::= { _625k-MCMiscComponents 8 }
23
24
25
26
27     _625k-MCPAUnitTable          OBJECT-TYPE
28     SYNTAX SEQUENCE OF _625k-MCPAUnitTableEntry
29     MAX-ACCESS not-accessible
30     STATUS Current
31     DESCRIPTION "PA Table"
32     ::= { _625k-MCPowerAmplifier 1 }
33
34
35
36     _625k-MCPAUnitTableEntry      OBJECT-TYPE
37     SYNTAX          _625k-MCPAUnitTableEntry
38     MAX-ACCESS      not-accessible
39     STATUS          Current
40     DESCRIPTION      ""
41     INDEX           { _625k-MCPAUnitTableIndex }
42     ::= { _625k-MCPAUnitTable 1 }
43
44     _625k-MCPAUnitTableEntry ::= SEQUENCE {
45         _625k-MCPAUnitTableIndex      INTEGER, -- PAUnitAddressType
46         _625k-MCRebootPAUnit          INTEGER, -- BooleanType
47         _625k-MCStateOfPAUnit         INTEGER, -- ComponentStateType
48         _625k-MCPAUnitIndication      INTEGER, -- IndicationType
49         _625k-MCPAUnitSerialNumber    OCTET STRING (SIZE(0..20)), --
50         TextType X 20
51         _625k-MCPAUnitModelNumber     OCTET STRING (SIZE(0..20)), --
52         TextType X 20
53         _625k-MCPAUnitManufactureDate OCTET STRING (SIZE(0..20)), --
54         TextType X 20
55         _625k-MCPAUnitHardwareRevision OCTET STRING (SIZE(0..20)), --
56         TextType X 20
57         _625k-MCPAUnitManufactureID   OCTET STRING (SIZE(0..20)), --
58         TextType X 20
59         _625k-MCPAUnitTemperature     OCTET STRING (SIZE(0..4)) --
60         DegreesCelsiusType
61     }
62
63
64
65     _625k-MCPAUnitTableIndex      OBJECT-TYPE
66     SYNTAX          INTEGER -- PAUnitAddressType
67     MAX-ACCESS      read-only
68     STATUS          Current

```

```

1 DESCRIPTION      "
2   Description for mibCtl Type 207 PAUnitAddressType :
3     Base station power amplifier component unit address.
4
5     A power amplifier unit address is a subset of Base Station
6     component addresses,
7     restricted to power amplifier components only.
8     Power amplifiers boost radio frequency
9     signal levels.
10    [Limits: 0 3 ]
11    Type derived from mibCtl Type 14 Unsigned32Type :
12    32 bit unsigned integer.
13    Type derived from mibCtl Type 11 Word32Type :
14    32 bits of raw opaque data.
15    Derived from basic 32 bit word type.
16  "
17  ::= { _625k-MCPAUnitTableEntry 1 }
18
19
20
21  _625k-MCRebootPAUnit      OBJECT-TYPE
22  SYNTAX                    INTEGER -- BooleanType
23  MAX-ACCESS                read-write -- REALLY: write-only
24  STATUS                    Current
25  DESCRIPTION
26    "Action to reboot a PA unit.
27
28    This is a write-only element; only a value of TRUE is valid.
29
30    (From mibCtl ElementType 506 RebootPAUnit)
31  Description for mibCtl Type 16 BooleanType :
32    Truth value, 0=FALSE, 1=TRUE.
33
34    This is a subset of TriStateType; no UNDEFINED value is provided.
35    [Limits: 0 1 ]
36  Description for mibCtl BooleanType 0 FALSE :
37    False.
38  Description for mibCtl BooleanType 1 TRUE :
39    True.
40  "
41  ::= { _625k-MCPAUnitTableEntry 2 }
42
43
44
45  _625k-MCStateOfPAUnit     OBJECT-TYPE
46  SYNTAX                    INTEGER -- ComponentStateType
47  MAX-ACCESS                read-only
48  STATUS                    Current
49  DESCRIPTION
50    "State of PA as a whole.
51
52
53    (From mibCtl ElementType 212 StateOfPAUnit)
54  Description for mibCtl Type 71 ComponentStateType :
55    Component operational state.
56
57    A component begins in the Unknown state.
58    If not detected, it enters and remains in the NotPresent state.
59    If detected, it enters the Uninitialized state, from where it
60    may go to the Testing and Initializing states and then to the
61    Standby or Operating state depending upon permissions.
62    Due to loss of permissions or resources, it may revert from
63    the Operating state to the Standby state.
64    Due to failure or loss of permission, it may revert to the
65    Uninitialized state, perhaps by way of the ShuttingDown state
66    depending on the device.
67    From the Uninitialized state it may return to more advanced
68    states depending upon permissions.

```

```

1         In case of a waiting period before (again) initializing,
2         the component is considered to be Initializing.
3
4         Permissions include administrative permissions (from the
5         operator); excessive failure restrictions; etc.
6     Description for mibCtl ComponentStateType 0 Unknown :
7         Component state not known.
8     Description for mibCtl ComponentStateType 1 NotPresent :
9         Component is not present.
10    Description for mibCtl ComponentStateType 2 PowerOff :
11        Component is present but powered off.
12    Description for mibCtl ComponentStateType 3 Uninitialized :
13        Component is present but not in use.
14
15        The power on/off state of the component is not specified in
16        this case.
17    Description for mibCtl ComponentStateType 4 Testing :
18        Component is being tested.
19    Description for mibCtl ComponentStateType 5 Initializing :
20        Component is being initialized.
21    Description for mibCtl ComponentStateType 6 Ready :
22        Component is ready but not operating.
23    Description for mibCtl ComponentStateType 7 Operating :
24        Component is operating for normal use without restriction.
25
26        The component is either in actual use or may be used at any time,
27        without restriction.
28    Description for mibCtl ComponentStateType 8 Abandoned :
29        Component state is not the desired state due to excessive errors.
30
31        The component state is not that desired, and the Base Station
32        software has abandoned attempts to place the component in
33        the desired state.
34        The actual state of the component is undefined.
35        The Base Station software will resume attempting to place the
36        component in the desired state if the appropriate Reinitialize
37        action element is written with the correct value.
38        Also, the software may resume attempts under other conditions,
39        not all of which may be documented.
40    Description for mibCtl ComponentStateType 9 InitialSetUp :
41        Component is initial set up..
42
43        Initial set up state.
44    Description for mibCtl ComponentStateType 10 Degrading :
45        Component is degrading..
46
47        Degrading state.
48    Description for mibCtl ComponentStateType 11 Restriction :
49        Component is restriction..
50
51        Restriction state.
52    "
53    ::= { _625k-MCPAUnitTableEntry 3 }
54
55
56
57    _625k-MCPAUnitIndication          OBJECT-TYPE
58        SYNTAX          INTEGER -- IndicationType
59        MAX-ACCESS      read-only
60        STATUS          Current
61        DESCRIPTION
62            "Status of PA Unit indicator.
63
64
65
66        (From mibCtl ElementType 526 PAUnitIndication)
67    Description for mibCtl Type 230 IndicationType :
68        Hardware indication status (LEDs).

```

```

1
2
3      Description for mibCtl IndicationType 0 Off :
4          Off.
5      Description for mibCtl IndicationType 1 Amber :
6          Amber.
7      Description for mibCtl IndicationType 2 Red :
8          Red.
9      Description for mibCtl IndicationType 3 Green :
10         Green.
11     Description for mibCtl IndicationType 4 NotPresent :
12         Not present.
13     "
14     ::= { _625k-MCPAUnitTableEntry 4 }
15
16
17
18     _625k-MCPAUnitSerialNumber          OBJECT-TYPE
19     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
20     MAX-ACCESS      read-only
21     STATUS          Current
22     DESCRIPTION
23         "PA unit serial number text.
24
25         Factory set uniquely for each component.
26
27         (From mibCtl ElementType 560 PAUnitSerialNumber)
28     Description for mibCtl Type 15 TextType :
29         ASCII or compatible text.
30     Type derived from mibCtl Type 12 OctetType :
31         8 bits of raw opaque data.
32     Derived from basic 8 bit word type.
33     "
34     ::= { _625k-MCPAUnitTableEntry 5 }
35
36
37
38     _625k-MCPAUnitModelNumber          OBJECT-TYPE
39     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
40     MAX-ACCESS      read-only
41     STATUS          Current
42     DESCRIPTION
43         "PA unit model number.
44
45         Factory set with description of component type, including
46         the major revision level.
47
48         (From mibCtl ElementType 561 PAUnitModelNumber)
49     Description for mibCtl Type 15 TextType :
50         ASCII or compatible text.
51     Type derived from mibCtl Type 12 OctetType :
52         8 bits of raw opaque data.
53     Derived from basic 8 bit word type.
54     "
55     ::= { _625k-MCPAUnitTableEntry 6 }
56
57
58
59     _625k-MCPAUnitManufactureDate      OBJECT-TYPE
60     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
61     MAX-ACCESS      read-only
62     STATUS          Current
63     DESCRIPTION
64         "PA unit manufacture date.
65
66         Factory set to month and date of manufacture date of the module.
67
68         (From mibCtl ElementType 562 PAUnitManufactureDate)

```

```

1      Description for mibCtl Type 15 TextType :
2          ASCII or compatible text.
3      Type derived from mibCtl Type 12 OctetType :
4          8 bits of raw opaque data.
5      Derived from basic 8 bit word type.
6      "
7      ::= { _625k-MCPAUnitTableEntry 7 }
8
9
10
11     _625k-MCPAUnitHardwareRevision          OBJECT-TYPE
12     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
13     MAX-ACCESS      read-only
14     STATUS          Current
15     DESCRIPTION
16         "PA unit hardware revision name.
17
18         Set at the factory to indicate the minor hardware revision
19         level of the module.
20
21         (From mibCtl ElementType 563 PAUnitHardwareRevision)
22     Description for mibCtl Type 15 TextType :
23         ASCII or compatible text.
24     Type derived from mibCtl Type 12 OctetType :
25         8 bits of raw opaque data.
26     Derived from basic 8 bit word type.
27     "
28     ::= { _625k-MCPAUnitTableEntry 8 }
29
30
31
32     _625k-MCPAUnitManufactureID            OBJECT-TYPE
33     SYNTAX          OCTET STRING (SIZE(0..20)) -- TextType X 20
34     MAX-ACCESS      read-only
35     STATUS          Current
36     DESCRIPTION
37         "PA unit manufacture ID.
38
39
40
41         (From mibCtl ElementType 564 PAUnitManufactureID)
42     Description for mibCtl Type 15 TextType :
43         ASCII or compatible text.
44     Type derived from mibCtl Type 12 OctetType :
45         8 bits of raw opaque data.
46     Derived from basic 8 bit word type.
47     "
48     ::= { _625k-MCPAUnitTableEntry 9 }
49
50
51
52     _625k-MCPAUnitTemperature              OBJECT-TYPE
53     SYNTAX          OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
54     MAX-ACCESS      read-only
55     STATUS          Current
56     DESCRIPTION
57         "The temperature of PAUnit (degrees Celsius).
58
59         The latest recorded temperature of a given PAUnit.
60
61         (From mibCtl ElementType 539 PAUnitTemperature)
62     Description for mibCtl Type 807 DegreesCelsiusType :
63         Temperature in degrees Celsius.
64     Type derived from mibCtl Type 15 TextType :
65         ASCII or compatible text.
66     Type derived from mibCtl Type 12 OctetType :
67         8 bits of raw opaque data.
68     Derived from basic 8 bit word type.

```

```

1      "
2      ::= { _625k-MCPAUnitTableEntry 10 }
3
4
5
6
7      _625k-MCPAModuleTable                                OBJECT-TYPE
8      SYNTAX SEQUENCE OF _625k-MCPAModuleTableEntry
9      MAX-ACCESS not-accessible
10     STATUS Current
11     DESCRIPTION "PA Module Table"
12     ::= { _625k-MCPowerAmplifier 2 }
13
14
15
16     _625k-MCPAModuleTableEntry                            OBJECT-TYPE
17     SYNTAX _625k-MCPAModuleTableEntry
18     MAX-ACCESS not-accessible
19     STATUS Current
20     DESCRIPTION ""
21     INDEX { _625k-MCPAModuleTableIndex1, _625k-MCPAModuleTableIndex2 }
22     ::= { _625k-MCPAModuleTable 1 }
23
24     _625k-MCPAModuleTableEntry ::= SEQUENCE {
25         _625k-MCPAModuleTableIndex1    INTEGER, -- PAUnitAddressType
26         _625k-MCPAModuleTableIndex2    INTEGER -- PAModuleAddressType
27     }
28
29
30
31     _625k-MCPAModuleTableIndex1                          OBJECT-TYPE
32     SYNTAX INTEGER -- PAUnitAddressType
33     MAX-ACCESS read-only
34     STATUS Current
35     DESCRIPTION "
36         Description for mibCtl Type 207 PAUnitAddressType :
37         Base station power amplifier component unit address.
38
39         A power amplifier unit address is a subset of Base Station
40         component addresses,
41         restricted to power amplifier components only.
42         Power amplifiers boost radio frequency
43         signal levels.
44         [Limits: 0 3 ]
45         Type derived from mibCtl Type 14 Unsigned32Type :
46         32 bit unsigned integer.
47         Type derived from mibCtl Type 11 Word32Type :
48         32 bits of raw opaque data.
49         Derived from basic 32 bit word type.
50     "
51     ::= { _625k-MCPAModuleTableEntry 1 }
52
53
54
55     _625k-MCPAModuleTableIndex2                          OBJECT-TYPE
56     SYNTAX INTEGER -- PAModuleAddressType
57     MAX-ACCESS read-only
58     STATUS Current
59     DESCRIPTION "
60         Description for mibCtl Type 208 PAModuleAddressType :
61         Base station power amplifier component module address.
62
63         A power amplifier module address is a subset of Base Station
64         component addresses,
65         [Limits: 0 2 ]
66         Type derived from mibCtl Type 14 Unsigned32Type :
67         32 bit unsigned integer.
68         Type derived from mibCtl Type 11 Word32Type :

```



```

1         32 bits of raw opaque data.
2         Derived from basic 32 bit word type.
3     "
4     ::= { _625k-MCPAModuleTableEntry 2 }
5
6
7
8     _625k-MCPowerSupply          OBJECT IDENTIFIER
9     -- DESCRIPTION               "Power supply"
10    ::= { _625k-MCMiscComponents 9 }
11
12
13
14
15    _625k-MCPowerSupplyTable      OBJECT-TYPE
16    SYNTAX SEQUENCE OF _625k-MCPowerSupplyTableEntry
17    MAX-ACCESS not-accessible
18    STATUS Current
19    DESCRIPTION "Power supply Table"
20    ::= { _625k-MCPowerSupply 1 }
21
22
23
24    _625k-MCPowerSupplyTableEntry OBJECT-TYPE
25    SYNTAX _625k-MCPowerSupplyTableEntry
26    MAX-ACCESS not-accessible
27    STATUS Current
28    DESCRIPTION ""
29    INDEX { _625k-MCPowerSupplyTableIndex }
30    ::= { _625k-MCPowerSupplyTable 1 }
31
32    _625k-MCPowerSupplyTableEntry ::= SEQUENCE {
33        _625k-MCPowerSupplyTableIndex INTEGER, -- PowerAddressType
34        _625k-MCStateOfPowerSupply   INTEGER, -- ComponentStateType
35        _625k-MCPowerSupplyIndication INTEGER -- IndicationType
36    }
37
38
39
40    _625k-MCPowerSupplyTableIndex OBJECT-TYPE
41    SYNTAX INTEGER -- PowerAddressType
42    MAX-ACCESS read-only
43    STATUS Current
44    DESCRIPTION "
45        Description for mibCtl Type 211 PowerAddressType :
46        Component power supply address.
47        [Limits: 0 2 ]
48        Type derived from mibCtl Type 14 Unsigned32Type :
49        32 bit unsigned integer.
50        Type derived from mibCtl Type 11 Word32Type :
51        32 bits of raw opaque data.
52        Derived from basic 32 bit word type.
53    "
54    ::= { _625k-MCPowerSupplyTableEntry 1 }
55
56
57
58    _625k-MCStateOfPowerSupply     OBJECT-TYPE
59    SYNTAX INTEGER -- ComponentStateType
60    MAX-ACCESS read-only
61    STATUS Current
62    DESCRIPTION
63        "State of Power supply as a whole.
64
65        (From mibCtl ElementType 213 StateOfPowerSupply)
66        Description for mibCtl Type 71 ComponentStateType :
67        Component operational state.
68

```

```

1
2     A component begins in the Unknown state.
3     If not detected, it enters and remains in the NotPresent state.
4     If detected, it enters the Uninitialized state, from where it
5     may go to the Testing and Initializing states and then to the
6     Standby or Operating state depending upon permissions.
7     Due to loss of permissions or resources, it may revert from
8     the Operating state to the Standby state.
9     Due to failure or loss of permission, it may revert to the
10    Uninitialized state, perhaps by way of the ShuttingDown state
11    depending on the device.
12    From the Uninitialized state it may return to more advanced
13    states depending upon permissions.
14    In case of a waiting period before (again) initializing,
15    the component is considered to be Initializing.
16
17    Permissions include administrative permissions (from the
18    operator); excessive failure restrictions; etc.
19    Description for mibCtl ComponentStateType 0 Unknown :
20        Component state not known.
21    Description for mibCtl ComponentStateType 1 NotPresent :
22        Component is not present.
23    Description for mibCtl ComponentStateType 2 PowerOff :
24        Component is present but powered off.
25    Description for mibCtl ComponentStateType 3 Uninitialized :
26        Component is present but not in use.
27
28        The power on/off state of the component is not specified in
29        this case.
30    Description for mibCtl ComponentStateType 4 Testing :
31        Component is being tested.
32    Description for mibCtl ComponentStateType 5 Initializing :
33        Component is being initialized.
34    Description for mibCtl ComponentStateType 6 Ready :
35        Component is ready but not operating.
36    Description for mibCtl ComponentStateType 7 Operating :
37        Component is operating for normal use without restriction.
38
39        The component is either in actual use or may be used at any time,
40        without restriction.
41    Description for mibCtl ComponentStateType 8 Abandoned :
42        Component state is not the desired state due to excessive errors.
43
44        The component state is not that desired, and the Base Station
45        software has abandoned attempts to place the component in
46        the desired state.
47        The actual state of the component is undefined.
48        The Base Station software will resume attempting to place the
49        component in the desired state if the appropriate Reinitialize
50        action element is written with the correct value.
51        Also, the software may resume attempts under other conditions,
52        not all of which may be documented.
53    Description for mibCtl ComponentStateType 9 InitialSetUp :
54        Component is initial set up..
55
56        Initial set up state.
57    Description for mibCtl ComponentStateType 10 Degrading :
58        Component is degrading..
59
60        Degrading state.
61    Description for mibCtl ComponentStateType 11 Restriction :
62        Component is restriction..
63
64        Restriction state.
65    "
66 ::= { _625k-MCPowerSupplyTableEntry 2 }
67
68

```

```

1
2  _625k-MCPowerSupplyIndication          OBJECT-TYPE
3      SYNTAX          INTEGER -- IndicationType
4      MAX-ACCESS      read-only
5      STATUS          Current
6      DESCRIPTION
7          "Status of Power Supply indicator.
8
9
10
11         (From mibCtl ElementType 527 PowerSupplyIndication)
12         Description for mibCtl Type 230 IndicationType :
13             Hardware indication status (LEDs).
14
15
16         Description for mibCtl IndicationType 0 Off :
17             Off.
18         Description for mibCtl IndicationType 1 Amber :
19             Amber.
20         Description for mibCtl IndicationType 2 Red :
21             Red.
22         Description for mibCtl IndicationType 3 Green :
23             Green.
24         Description for mibCtl IndicationType 4 NotPresent :
25             Not present.
26         "
27 ::= { _625k-MCPowerSupplyTableEntry 3 }
28
29
30
31 END

```


1
2