IEEE STANDARDS ASSOCIATION

IEEE Standard for

Local and metropolitan area networks—

Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility— Physical and Media Access Control Layer Specification

Amendment 1: Management Information Base Enhancements and Corrigenda Items

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

9 December 2010

IEEE Std 802.20a™-2010

(Amendment to
IEEE Std 802.20™-2008)

IEEE Std 802.20a™-2010
(Amendment to
IEEE Std 802.20™-2008)

IEEE Standard for Local and metropolitan area networks—

Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility— Physical and Media Access Control Layer Specification

Amendment 1: Management Information Base Enhancements and Corrigenda Items

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 8 December 2010

IEEE-SA Standards Board

Abstract: This amendment enhances IEEE 802.20 Management Information Base (MIB) clauses to include additional information and MIB documentation (i.e., a discussion of security impacts, and cross references to the text for appropriate parameters within the MIB, itself) to conform with the applicable parts of the IETF's 2008 guidelines for MIBs. The amendment also corrects any errors, inconsistencies, and ambiguities associated with the MIB clause of IEEE Std 802.20-2008.
Keywords: 802.20, Management Information Base, MIB

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6458-8 STD97017
Print: ISBN 978-0-7381-6459-5 STDPD97017

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 802.20a-2010, IEEE Standard for Local and metropolitan area networks—Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility—Physical and Media Access Control Layer Specification—Amendment 1: Management Information Base Enhancements and Corrigenda Items.

This amendment enhances the IEEE 802.20 Management Information Base (MIB) clauses to include additional information and MIB documentation (i.e., a discussion of Security impacts, and cross references to the text for appropriate parameters within the MIB, itself) to conform with the applicable parts of the IETF's 2008 guidelines for MIBs. The amendment also corrects any errors, inconsistencies, and ambiguities associated with the MIB clause of IEEE Std 802.20-2008.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this amendment may require use of subject matter covered by patent rights. By publication of this amendment, no position is taken with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses. Other Essential Patent Claims may exist for which a statement of assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this amendment are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this amendment was submitted to the IEEE-SA Standards Board for approval, the Working Group had the following membership:

Mark Klerer, *Chair*
Radhakrishna Canchi, *Vice Chair*

Nancy Bravin
Kazuhito Ishida
Francis O'Brien
James Ragsdale
Ajay Rajkumar
Jerry Upton

The following members of the individual balloting committee voted on this amendment. Balloters may have voted for approval, disapproval, or abstention.

Richard Alfvin
Butch Anton
Danilo Antonelli
William Byrd
Radhakrishna Canchi
Juan Carreon
Keith Chow
Charles Cook
Thomas Dineen
Sourav Dutta
C. Fitzgerald
Devon Gayle
Randall Groves
C. Guy
Robert F. Heile
Paul Isaacs
Akio Iso
Atsushi Ito
Raj Jain
Bobby Jose
Shinkyo Kaku
Masahiko Kaneko
Piotr Karocki
Stuart J. Kerry
Yongbum Kim
Mark Klerer
Gerald L. Kolbe
Thomas Kowalick
Jeremy Landt
William Lumpkins
G. Luri
Mark Maloney
Avygdor Moise
Michael S. Newman
Satoshi Oyama
Glenn Parsons
James Ragsdale
Robert Robinson
Benjamin Rolfe
Randall Safier
Bartien Sayogo
Gil Shultz
Kapil Sood
Amjad Soomro
Thomas Starai
Walter Struppler
Mark Sturza
James Tomcik
Mark-Rene Uchida
Prabodh Varshney
Joanne Wilson
Oren Yuen

When the IEEE-SA Standards Board approved this amendment on 8 December 2010, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Catherine Berger
IEEE Standards Project Editor

Michael Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for
Local and metropolitan area networks—

Part 20: Air Interface for Mobile Broadband Wireless Access Systems Supporting Vehicular Mobility— Physical and Media Access Control Layer Specification

Amendment 1: Management Information Base Enhancements and Corrigenda Items

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

Editorial Note

The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard. The editing instructions are shown in ***bold italics.*** Three editing instructions are used: ***change***, ***delete***, and ***insert***. ***Change*** is used to make a change to existing material. The editing instruction specifies the location of the change and describes what is being changed. Changes to existing text may be clarified using ~~strikeout~~ markings to indicate removal of old material, and underscore markings to indicate addition of new material). ***Delete*** removes existing material. ***Insert*** adds new material without changing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.

Changes are applied to the base text of IEEE Std 802.20-2008.

Replace Clause 17 of IEEE Std 802.20-2008 with the following text:

17. MAC and PHY MIB

17.1 Overview

This clause 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, IETF RFC 2578, IETF RFC 2579, and IETF 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 in Figure 2. The MIB object is composed of the following 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 that 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 eight objects used to set the frequency band of the transmitting base station. An administrator should take great care to include only authorized, licensed channel bands in the table. Failure to take these measures might cause a base station to violate local regulatory laws (e.g., FCC licensing in the USA) by transmitting power into unauthorized channels in the country where the base station is deployed.

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

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

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

There are no MIB objects that could be used to turn off or change the security parameter configuration of an IEEE 802.20 access node. The presence or absence of security (encryption, authentication) is controlled by the session state record for each individual user, and cannot be 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 whom 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 IETF RFC 3410, 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.

17.4 IANA considerations

No IANA actions are required by this clause.

17.5 Definition[1]

```
IEEE802dot20-MIB DEFINITIONS ::= BEGIN

IMPORTS
    ifIndex
        FROM IF-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    Counter32, Counter64, Integer32, MODULE-IDENTITY, OBJECT-IDENTITY,
    OBJECT-TYPE, transmission
        FROM SNMPv2-SMI
    RowPointer, RowStatus, TEXTUAL-CONVENTION, TruthValue
        FROM SNMPv2-TC
    ;

ieee802dot20 MODULE-IDENTITY
    LAST-UPDATED  "201010041622PDT" -- October 4, 2010
    ORGANIZATION
        "IEEE 802.20"
    CONTACT-INFO
      "WG URl: ieee802.org/20
        Contact: J. Tomcik         Postal:  5775 Morehouse Dr
        San Diego, CA, 92121, USA  Tel: 858-658-3231
         Fax: 858-658-3231
         E-mail: jtomcik@qualcomm.com  "
```

[1] *Copyright release for MIB definitions:* Users of this standard may freely reproduce the MIB definitions in this annex so it can be used for its intended purpose.

```
    DESCRIPTION
        "The MIB module for IEEE 802.20 entities."

  ::= { iso (1) iso-identified-organization (3) ieee (111)
standardsassociation-numbered-series-standards (2) lan-man-stds (802)
ieee802dot20(20) ieee802dot20mibs(1) ieee802dot20widebandMIB(1)}

Dot20AnChannelBandsEntry ::= SEQUENCE
{
    dot20AnChannelBandIndex          Integer32,
    dot20AnSystemType                Integer32,
    dot20AnBandClass                 Integer32,
    dot20AnChannelNumber             Integer32,
    dot20AnHalfDuplexSupported       TruthValue,
    dot20AnReverseChannelBandClass   Integer32,
    dot20AnReverseChannelNumber      Integer32,
    dot20AnCyclicPrefixLength        Integer32,
    dot20AnFFTSize                   Integer32,
    dot20AnCBNumGuardSubcarriers     Integer32,
    dot20AnChannelBandShortId        Integer32,
    dot20AnChannelBandAccessHashMask Integer32,
    dot20AnChannelBandStatus         RowStatus
}

Dot20AnIdleStateStatsEntry ::= SEQUENCE
{
    dot20AnAccessAttemptCounts     Counter32,
    dot20AnAccessAttemptFailCounts Counter32,
    dot20AnPageAttemptCounts       Counter32,
    dot20AnPageFailureCounts       Counter32
}

Dot20AnNeighborListEntry ::= SEQUENCE
{
    dot20AnNeighborIndex         Integer32,
    dot20AnNeighborSectorPointer RowPointer,
    dot20AnNeighborRowStatus     RowStatus
}

Dot20AnNeighborSectorsEntry ::= SEQUENCE
{
    dot20AnNeighborSectorIndex       Integer32,
    dot20AnNeighborPilotID           Integer32,
    dot20AnNeighborEffTransmitPower  Integer32,
    dot20AnNeighborChannelBandRef    Integer32,
    dot20AnNeighborChannelShortID    Integer32,
    dot20AnNeighborSameANAsPrimSect  TruthValue,
    dot20AnNeighborSectorPilotGrpId  Integer32,
    dot20AnNeighborSynchGroupId      Integer32,
    dot20AnNeighborSectorCellGroupId Integer32,
    dot20AnNeighborSectorStatus      RowStatus
}

Dot20AnOtherTechNghbrsEntry ::= SEQUENCE
{
    dot20AnOtherTechnologyIndex    Integer32,
    dot20AnTechnologyType          Integer32,
    dot20AnTechNghbrListLength     Integer32,
    dot20AnTechnologyNeighborList  OCTET STRING,
    dot20AnOtherTechNghbrRowStatus RowStatus
}

Dot20AnSecondaryRegZoneCodeEntry ::= SEQUENCE
{
    dot20AnSecondaryRegZoneCodeIndex Integer32,
    dot20AnSecRegZoneCode             Integer32,
    dot20AnSecondaryRegZoneRowStatus RowStatus
}

Dot20AnSectorCdmaSubSegEntry ::= SEQUENCE
{
    dot20AnInterlaceId                Integer32,
    dot20AnCdmaSubSegmentNum          Integer32,
    dot20AnSectorCdmaSubSegRowStatus RowStatus
}

Dot20AnSectorConfigEntry ::= SEQUENCE
{
    dot20AnTotalNumSubcarriers      Integer32,
    dot20AnNumGuardSubcarriers      Integer32,
    dot20AnFlSubzoneSize            Integer32,
    dot20AnResourceChannelMuxMode   Integer32,
    dot20AnNumDRCHSubzones          Integer32,
    dot20AnFLReservedInterlaces     INTEGER,
    dot20AnNumFLReservedSubzones    Integer32,
    dot20AnCpichHoppingMode         Integer32,
    dot20AnNumEffectiveAntennas     Integer32,
    dot20AnNumCommonSegmentHopPorts Integer32,
    dot20AnNumLABSegments           Integer32,
    dot20AnMinScchResourceIndex     Integer32,
    dot20AnSinglePAForXCarriers     Integer32,
    dot20AnFlSdmaNumSubtrees        Integer32,
    dot20AnFDPICHCodeOffsetSubtree0 Integer32,
    dot20AnFDPICHCodeOffsetSubtree1 Integer32,
    dot20AnFDPICHCodeOffsetSubtree2 Integer32,
    dot20AnFDPICHCodeOffsetSubtree3 Integer32,
    dot20AnNumCmnPilotTxAnt         Integer32,
    dot20AnModSymbolsPerQPSKLAB     Integer32,
    dot20AnUseDrchForFlcs           Integer32,
    dot20AnEnableExpandedQPCH       TruthValue,
    dot20AnSectorConfigRowStatus    RowStatus
}

Dot20AnSectorExtChanInfoEntry ::= SEQUENCE
{
    dot20AnPilotID                    Integer32,
    dot20AnHalfDuplexModeSupported   TruthValue,
    dot20AnFACKBandwidthFactor        Integer32,
    dot20AnSFNCellID                  Integer32,
    dot20AnCellNullID                 Integer32,
    dot20AnMaxNumSharedLABs           Integer32,
    dot20AnMaxNumLABs                 Integer32,
    dot20AnMax16QamScchBlocks         Integer32,
    dot20AnPdCabResSharingEnabled     TruthValue,
    dot20AnNumAckableLABs             Integer32,
    dot20An16QamScchT2PRatio          INTEGER,
    dot20AnEffectiveTransmitPower     Integer32,
    dot20AnAssignmentAckHARQTx        Integer32,
    dot20AnCQIPilotTransmitPower      Integer32,
    dot20AnCommonPilotTransmitPower   Integer32,
    dot20AnCDMAInterlacesBitmap       Integer32,
    dot20AnNumOdcchReports            Integer32,
    dot20AnNumRLCdmaSubsegments       Integer32,
    dot20AnRackBandwidthFactor        Integer32,
    dot20AnRlNumSdmaDimensions        Integer32,
    dot20AnRlDpichCodeOffsetSubtree0 Integer32,
    dot20AnRlDpichCodeOffsetSubtree1 Integer32,
    dot20AnRlDpichCodeOffsetSubtree2 Integer32,
    dot20AnRlDpichCodeOffsetSubtree3 Integer32,
    dot20AnRlSubzoneSize              Integer32,
    dot20AnSilenceIntervalPeriod      Integer32,
    dot20AnSilenceIntervalDuration   Integer32,
    dot20AnNumSilenceIntervalSubzone Integer32,
    dot20AnAckInterferenceOffset     Integer32,
    dot20AnMacIdRange                INTEGER,
    dot20AnFlPcReportInterval        Integer32,
    dot20AnFlPqiReportInterval       Integer32,
    dot20AnFlIotReportInterval       Integer32,
    dot20AnFastIoTEnabled            TruthValue,
    dot20AnFastOSIEnabled            TruthValue,
    dot20AnRabEnabled                TruthValue,
    dot20AnOsiResponseMode           INTEGER,
    dot20AnSlowInterferenceOffset    Integer32,
    dot20AnCtrlAccessOffset          Integer32,
    dot20AnRlAuxPilotPower           Integer32,
    dot20AnReqQoSPowerBoost          Integer32,
    dot20AnErasureTargetCtoI0        Integer32,
    dot20AnErasureTargetCtoI1        Integer32,
    dot20AnErasureTargetCtoI2        Integer32,
    dot20AnErasureTargetCtoI3        Integer32,
    dot20AnAccessCycleDuration       Integer32,
    dot20AnMaxProbesPerSequence      Integer32,
    dot20AnProbeRampUpStepSize       Integer32,
    dot20AnPilotThreshold1           Integer32,
    dot20AnPilotThreshold2           Integer32,
    dot20AnOpenLoopAdjust            Integer32,
    dot20AnAccessRetryPersistance0   Integer32,
    dot20AnAccessRetryPersistence1   Integer32,
    dot20AnAccessRetryPersistence2   Integer32,
    dot20AnAccessRetryPersistence3   Integer32,
    dot20AnAccessRetryPersistence4   Integer32,
    dot20AnAccessRetryPersistence5   Integer32,
    dot20AnAccessRetryPersistence6   Integer32,
    dot20AnAccessRetryPersistence7   Integer32,
    dot20AnSectorExtChanRowStatus    RowStatus
}

Dot20AnSectorGrpResSetsEntry ::= SEQUENCE
{
    dot20AnResourceSetId             Integer32,
    dot20AnResourceSetBitmap         Integer32,
    dot20AnBRCHSubzoneCyclingEnabled TruthValue,
    dot20AnResourceSetSubZoneSpacing Integer32,
    dot20AnNumResourceSubzones       Integer32,
    dot20AnResourceSubzoneOffset     Integer32,
    dot20AnResourceSetRowStatus      RowStatus
}

Dot20AnSectorIpsiEntry ::= SEQUENCE
{
    dot20AnIpsiIndex     Integer32,
    dot20AnSupportedIpsi Integer32,
    dot20AnIpsiRowStatus RowStatus
}

Dot20AnSectorParamEntry ::= SEQUENCE
{
    dot20AnMobileCountryCode    Integer32,
    dot20AnMobileNetworkCode    Integer32,
    dot20AnSectorID             OCTET STRING,
    dot20AnChannelBandRef       Integer32,
    dot20AnLatitude             Integer32,
    dot20AnLongitude            Integer32,
    dot20AnLeapSeconds          Integer32,
    dot20AnLocalTimeOffset      Integer32,
    dot20AnPrimaryRegZoneCode   Integer32,
    dot20AnAnGroupId            Integer32,
    dot20AnPilotGroupId         Integer32,
    dot20AnSynchronousGroupId   Integer32,
    dot20AnCellGroupId          Integer32,
    dot20AnSectorParamRowStatus RowStatus
}

Dot20AnSectorToIfIndexEntry ::= SEQUENCE
{
    dot20AnIfChannelBandRef Integer32
}

Dot20CmnAuthStatsEntry ::= SEQUENCE
{
    dot20CmnAuthFailureCounts Counter64,
    dot20CmnAuthSuccessCounts Counter64
}

Dot20CmnLMACPacketStatsEntry ::= SEQUENCE
{
    dot20CmnPacketFormatIndex Integer32,
    dot20CmnARQAttemptsIndex  Integer32,
    dot20CmnFwdTxPacketCounts Counter64,
    dot20CmnRevRxPacketCounts Counter64
}

Dot20CmnLMACStatsEntry ::= SEQUENCE
{
    dot20CmnFLABCounts        Counter64,
    dot20CmnRLABCounts        Counter64,
    dot20CmnAccessGrantCounts Counter64
}

Dot20CmnQmpStatsEntry ::= SEQUENCE
{
    dot20CmnActiveReservationsCounts Counter64,
    dot20CmnIdleReservationsCounts   Counter64,
    dot20CmnReservationOpenCounts    Counter64,
    dot20CmnReservationCloseCounts   Counter64,
    dot20CmnReservationFailCounts    Counter64
}

Dot20CmnRlpStatsEntry ::= SEQUENCE
{
    dot20CmnStreamId          Integer32,
    dot20CmnRlpTxBytes        Counter64,
    dot20CmnRlpReTxBytes      Counter64,
    dot20CmnRlpTxDropBytes    Counter64,
    dot20CmnRlpTxStatus       Counter64,
    dot20CmnRlpRxBytes        Counter64,
    dot20CmnRlpRxStatus       Counter64,
    dot20CmnRlpTxPackets      Counter64,
    dot20CmnRlpReTxPackets    Counter64,
    dot20CmnRlpTxrDropPackets Counter64,
    dot20CmnRlpRxPackets      Counter64,
    dot20CmnRlpTxNAKTimeouts  Counter64,
    dot20CmnRlpTxACKTimeouts  Counter64
}

dot20An OBJECT-IDENTITY
    STATUS        current
    DESCRIPTION
        "AN specific configuration and statistics."
    ::= { ieee802dot20 1 }

dot20AnMac OBJECT-IDENTITY
    STATUS        current
    DESCRIPTION
        "MAC layer objects"
    ::= { dot20An 1 }

dot20AnConnectionControl OBJECT IDENTIFIER ::= { dot20AnMac 3 }

dot20AnIdleState OBJECT IDENTIFIER ::= { dot20AnConnectionControl 1 }

dot20AnIdleStateStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnIdleStateStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row of Idle State protocol statistics
         per 802.20 interface (i.e. sector for a specific ChannelBand)
         and carrier."
    ::= { dot20AnIdleState 1 }

dot20AnIdleStateStatsEntry OBJECT-TYPE
    SYNTAX        Dot20AnIdleStateStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the IdleStateStats table. This
         table is indexed by ifIndex and CarrierID(see 11.5.5.8). ifIndex:
Each IEEE
         802.20 interface (uniquely identified by SectorID) is
         represented by an ifEntry. In the case of a multicarrier
         Sector, the carrierID indentifies one specific carrier."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.4 (Access Channel MAC
         Protocol)"
    INDEX
        { ifIndex }
    ::= { dot20AnIdleStateStatsTable 1 }

dot20AnAccessAttemptCounts OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Access Attempts among all Terminals"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.4.5.5.2,
         (Access Channel MAC Protocol / AN Requirements)"
    ::= { dot20AnIdleStateStatsEntry 1 }

dot20AnAccessAttemptFailCounts OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Failed Access Attempts among all Terminals.
         Incremented when access RLAB is not used by a terminal."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.5.4.3.2 (BindATI), and
         Subclause 11.2.4.6.2.1 (issuing ConnectedState.Deactivate)"
    ::= { dot20AnIdleStateStatsEntry 2 }

dot20AnPageAttemptCounts OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Page Attempts"
    REFERENCE
```

```
        "IEEE Std. 802.20-2008, Subclause 8.3.5.8, and Table 208
(RouteOpenRequestReason encoding)"
    ::= { dot20AnIdleStateStatsEntry 3 }

dot20AnPageFailureCounts OBJECT-TYPE
    SYNTAX        Counter32
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Failed Page Attempts"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.3.5.8, and Table 358
(RouteOpenRequestReason encoding)"
    ::= { dot20AnIdleStateStatsEntry 4 }

dot20AnOverheadMessages OBJECT IDENTIFIER ::= { dot20AnConnectionControl 4 }

dot20AnSectorConfigTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorConfigEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 interface, i.e. sector
         for a specific ChannelBand. This table's attributes specify the
         configuration of the corresponding sector, and can be used to
         populate fields in SystemInfo block and QuickChannelInfo
         message, which are transmitted by the Overhead Messages Protocol."
    ::= { dot20AnOverheadMessages 1 }

dot20AnSectorConfigEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorConfigEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorConfig table. This
         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
         interface (uniquely identified by SectorID) is represented by
         an ifEntry."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6 (Overhead Messages Protocol)"
    INDEX
        { ifIndex }
    ::= { dot20AnSectorConfigTable 1 }

dot20AnTotalNumSubcarriers OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This parameter takes the value 2^(7+n), where n is the
         value of the 3 bit field. This field is not to be set to a
         value of 5 or above."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
    ::= { dot20AnSectorConfigEntry 29 }

dot20AnNumGuardSubcarriers OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the number of guard subcarriers
         as defined in 802.20 Physical layer clause."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
    ::= { dot20AnSectorConfigEntry 30 }
```

```
dot20AnFlSubzoneSize OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of subzones on the
         forward link. If n=0, this parameter is set to 64 and if
         n=1, this parameter is set to 128."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
    ::= { dot20AnSectorConfigEntry 31 }

dot20AnResourceChannelMuxMode OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of subzones on the
         forward link. If ResourceChannelMuxMode=0, this parameter is set to
64 and if
         ResourceChannelMuxMode=1, this parameter is set to 128."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 32 }

dot20AnNumDRCHSubzones OBJECT-TYPE
    SYNTAX        Integer32
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field takes values between 0 and N_FFT/64 - 1"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 33 }

dot20AnFLReservedInterlaces OBJECT-TYPE
    SYNTAX        INTEGER {
        zero(1),
        zeroToOne(2),
        zeroToTwo(3),
        zeroToThree(4),
        zeroToFour(5),
        zeroToFive(6),
        zeroToSix(7),
        zeroToSeven(8),
        zeroAndThree(9),
        zeroAndSix(10),
        zeroTwoAndFour(11),
        zeroTwoFourAndSix(12),
        reserved(13),
        reserved2(14),
        reserved3(15),
        none(16)
    }
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines which interlaces contain
         reserved bandwidth on the forward link."
    REFERENCE
        "IEEE Std. 802.20-2008, Table 285 (Interpretation of FL
         Reserved Interlaces), Subclause 11.6.5.2"
    ::= { dot20AnSectorConfigEntry 34 }

dot20AnNumFLReservedSubzones OBJECT-TYPE
```

```
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of subzones that are reserved
         on each interlace that contains reserved bandwidth"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.2 (SystemInfo Block)"
    ::= { dot20AnSectorConfigEntry 35 }

dot20AnCpichHoppingMode OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to 0 for deterministic, and 1 for
         random hopping"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 36 }

dot20AnNumEffectiveAntennas OBJECT-TYPE
    SYNTAX        Integer32 (1..8)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the number of
         effective antennas."
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 37 }

dot20AnNumCommonSegmentHopPorts OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the number of common segment
         hop ports."
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 38 }

dot20AnNumLABSegments OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field indicates the number of LABSegments."
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 39 }

dot20AnMinScchResourceIndex OBJECT-TYPE
    SYNTAX        Integer32 (0..31)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This parameter is in units of N_FFT/32 resources, and spans
         from 0 to N_FFT -1"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 40 }

dot20AnSinglePAForXCarriers OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
    "This field determines the structure of F-BPICH (SinglePAForMultipleChannelBands)"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 41 }

dot20AnFlSdmaNumSubtrees OBJECT-TYPE
    SYNTAX        Integer32 (1..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of sub-trees on the
         forward link. (FLNumSDMADimensions)"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 11.6.5.3 (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 42 }

dot20AnFLDPICHCodeOffsetSubtreeIndex0 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree
         0.  This subtree is always present, and is therefore not
         described in the overhead channels."
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
         Pilot Channel) and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorConfigEntry 43 }

dot20AnFLDPICHCodeOffsetSubtreeIndex1 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree
         1"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
         Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorConfigEntry 44 }

dot20AnFLDPICHCodeOffsetSubtree2 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree
         2"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
         Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorConfigEntry 45 }

dot20AnFLDPICHCodeOffsetSubtreeIndex3 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree
         3"
    REFERENCE
        "IEEE Std 802.20-2008 Subclause 9.4.1.2.3.2 (Forward Dedicated
         Pilot Channel), and Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorConfigEntry 46 }

dot20AnNumCmnPilotTxAnt OBJECT-TYPE
    SYNTAX        Integer32 (1..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute specifies the number of common pilot
         transmit antennas."
    REFERENCE
        "IEEE Std. 802.20-2008, Table 90 (NumEffectiveAntennas), Subclause
9.4.1.2.3.1.1 (Forward
         Common Pilot Channel Subcarriers), and Subclause 11.6.5.3
         (QuickChannelInfo Block)"
    ::= { dot20AnSectorConfigEntry 47 }

dot20AnModSymbolsPerQPSKLAB OBJECT-TYPE
    SYNTAX        Integer32 (0..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of modulation symbols
         for each block carried by the F-SCCH"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 287 (Interpretation of
         ModulationSymbolsPerQPSKLAB)"
    ::= { dot20AnSectorConfigEntry 48 }

dot20AnUseDrchForFlcs OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the hopping pattern on the FLCS.  It is set to
1 if the hopping pattern is DRCH on the FLCS, and is set to 0 otherwise"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo
         Block)"
    ::= { dot20AnSectorConfigEntry 49 }

dot20AnEnableExpandedQPCH OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of packets delivered to
         the Physical Layer by the MAC Layer"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.3 (QuickChannelInfo
         Block)"
    ::= { dot20AnSectorConfigEntry 50 }

dot20AnSectorConfigRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the SectorConfig Table. If
         the implementer of this MIB has chosen not to implement
         'dynamic assignment' of sectors, this attribute is not applicable
         and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnSectorConfigEntry 78 }
```

```
dot20AnSectorExtChanInfoTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorExtChanInfoEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 interface, i.e. sector
         for a specific ChannelBand. This table's attributes specify the
         configuration of the corresponding sector, and can be used to
         populate fields in extendedChannelInfo message."
    ::= { dot20AnOverheadMessages 2 }

dot20AnSectorExtChanInfoEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorExtChanInfoEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorExtChanInfo table. This
         table is indexed by IfIndex. ifIndex: Each IEEE 802.20
         interface (uniquely identified by SectorID) is represented by
         an ifEntry.  The Extended Channel Info is transmitted by the
         Overhead Messages Protocol."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4 (ExtendedChannelInfo)"
    INDEX
        { ifIndex }
    ::= { dot20AnSectorExtChanInfoTable 1 }

dot20AnPilotID OBJECT-TYPE
    SYNTAX        Integer32 (0..1023)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the PilotID of the sector."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 9.2.2.2.3 (PilotID and SectorSeed)"
    ::= { dot20AnSectorExtChanInfoEntry 1 }

dot20AnHalfDuplexModeSupported OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to True if the access network
         supports half duplex terminals, and is set to False
         otherwise. If half-duplex terminals are supported, the access
         network should assign MAC IDs and channel assignments in a
         manner that enables half-duplex terminal operation. A
         half-duplex access terminal is not required to monitor forward
         link transmissions on a PHY Frame where it is scheduled to make
         a reverse link transmission."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.4 (MACResourceAssignment)"
    ::= { dot20AnSectorExtChanInfoEntry 2 }

dot20AnFACKBandwidthFactor OBJECT-TYPE
    SYNTAX        Integer32 (1..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Forward Acknowledgement channel (FACK) bandwidth factor"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 3 }

dot20AnSFNCellID OBJECT-TYPE
    SYNTAX        Integer32 (0..511)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the ID of the single frequency network
         cell (for Broadcast Multicast Service)"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup),
         and Subclause 9.2.2.2.2 (SFNCellID and SFNPhase)"
    ::= { dot20AnSectorExtChanInfoEntry 5 }

dot20AnCellNullID OBJECT-TYPE
    SYNTAX        Integer32 (0..511)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Cell Null Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 6 }

dot20AnMaxNumSharedLABs OBJECT-TYPE
    SYNTAX        Integer32 (1..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the maximum number of shared LABs
         that are transmitted by this sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 7 }

dot20AnMaxNumLABs OBJECT-TYPE
    SYNTAX        Integer32 (0..63)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the Maximum number of LABs that can
         be transmitted by this sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 9 }

dot20AnMax16QamScchBlocks OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the maximum number of 16-QAM blocks
         that may be transmitted by the access network"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 10 }

dot20AnPdCabResSharingEnabled OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines if resource sharing using PDCABs is
         enabled"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 11 }

dot20AnNumAckableLABs OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the number of LABs on SCCH that the
         access terminal is to acknowledge"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 12 }

dot20An16QamScchT2PRatio OBJECT-TYPE
    SYNTAX        INTEGER {
        minusSevenDb(1),
        minusFourDb(2),
        zeroDb(3),
        minusTenDb(4)
    }
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "16 Qam Scch T2P Ratio"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 13 }

dot20AnEffectiveTransmitPower OBJECT-TYPE
    SYNTAX        Integer32 (0..63)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the effective transmit power of the
         sector in units of dBm"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 14 }

dot20AnAssignmentAckHARQTx OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The value 0 indicates that no ACK is sent in response to an
         assignment. The rules for interpreting other values of this
         field are provided in the Lower MAC Sublayer. The value 7 is
reserved"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 196, and Subclause 11.6.5.4.2
         (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 15 }

dot20AnCQIPilotTransmitPower OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The field determines the power spectral density of the F-CQIPICH
         relative to the reference transmit power density defined by the
Physical
         Layer. This parameter may take the value (-4 + n*0.5) dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 16 }

dot20AnCommonPilotTransmitPower OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The attribute's value (denoted n) determines the power
         spectral density of the F-CPICH during the FL PHY frame
         relative to the F-ACQCH. The pilot power density is equal
         to (-4 + n*0.5) dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.1 (ForwardChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 17 }

dot20AnCDMAInterlacesBitmap OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The j'th bit of this field is set to 1 if interlace i
         contains a Reverse Link CDMASegment. Here j is assumed to range
         from 0 through 7, and an interlace i is the set of PHY Frames
         that satisfy PHY Frame Index mod 8 = i"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
         (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 18 }

dot20AnNumOdcchReports OBJECT-TYPE
    SYNTAX        Integer32 (0..31)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Num ODCCH reports, specified in units of 16"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
         (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 27 }

dot20AnNumRLCdmaSubsegments OBJECT-TYPE
    SYNTAX        Integer32 (1..16)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of RLCdmaSubsegments on
         this sector."
    REFERENCE
        "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
         (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 28 }

dot20AnRackBandwidthFactor OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This parameter is set to 2^n, where n is the value of
         the two bit field."
    REFERENCE
        "IEEE Std. 802.20-2008, Table 137, and Subclause 11.6.5.4.2
         (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 30 }

dot20AnRlNumSdmaDimensions OBJECT-TYPE
    SYNTAX        Integer32 (1..4)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the number of spatial dimensions on
         the reverse link."
    REFERENCE
```

```
        "IEEE Std. 802.20-2008, Table 137  and
         Subclause 11.6.5.4.2 (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 31 }

dot20AnRLDpichCodeOffsetSubtreeIndex0 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree 0"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.2 (Reverse Channel Group)"
    ::= { dot20AnSectorExtChanInfoEntry 32 }

dot20AnRLDpichCodeOffsetSubtreeIndex1 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtreetree 1"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.1 (Reverse Channel Group)"
    ::= { dot20AnSectorExtChanInfoEntry 33 }

dot20AnRLDpichCodeOffsetSubtreeIndex2 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for tree 2"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.1 (Reverse Channel Group)"
    ::= { dot20AnSectorExtChanInfoEntry 34 }

dot20AnRLDpichCodeOffsetSubtreeIndex3 OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to the code offset for subtree 3"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.1 (Reverse Channel Group)"
    ::= { dot20AnSectorExtChanInfoEntry 35 }

dot20AnRLSubzoneSize OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the size of subzones on the reverse
         link. If n=0, this parameter takes the value 64 and if
         n=1, this parameter takes the value 128"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.2 (Reverse Channel Group)"
    ::= { dot20AnSectorExtChanInfoEntry 36 }
```

```
dot20AnSilenceIntervalPeriod OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the period in units of super frames
         when the silence interval repeats. The SilenceInterval takes
         a value of 2^n super frames, where n is the value of this parameter"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.2 (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 38 }

dot20AnSilenceIntervalDuration OBJECT-TYPE
    SYNTAX        Integer32 (1..8)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the duration silence interval in
         units of 8 OFDM symbols"
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.2 (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 39 }

dot20AnNumSilenceIntervalSubzone OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field specifies the set of subzones that are blanked
         during the silence interval."
    REFERENCE
        "IEEE Std. 802.20-2008, Table 185 (Physical Layer Numeric Constants
and Parameters), and
         Subclause 11.6.5.4.2 (ReverseChannelGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 40 }

dot20AnAckInterferenceOffset OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field may take values in units of dB"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 42 }

dot20AnMacIdRange OBJECT-TYPE
    SYNTAX        INTEGER {
        upTo63(1),
        upTo127(2),
        upTo255(3),
        upTo511(4),
        upTo1023(5),
        upTo2047(6),
        reserved(7),
        upTo31(8)
    }
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to indicate the range of assigned
         MACID values in the sector. For example, a MACIDRange of 63
         indicates that the sector has not assigned MACID values 64 and
         above"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 43 }

dot20AnFlPcReportInterval OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the periodicity at which power
         control commands are sent to the access terminal. This
         parameter can take the value 2^n, where n is the value of the
         three bit field."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 44 }

dot20AnFlPqiReportInterval OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the periodicity at which PQI
         reports commands are sent by this sector. This parameter
         takes the value 16*2^n, where n is the value of the three bit
         field"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 45 }

dot20AnFlIotReportInterval OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the periodicity at which IoT values
         are sent to the access terminal. This parameter takes the value 1, 8,
16, or 32, depending on the two bit field taking values 0, 1, 2, or 3.
 "
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 46 }

dot20AnIoTEnabled OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines if the sector transmits IOT"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 47 }

dot20AnFastOSIEnabled OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines if the access terminal is required
         to read OSI from this sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 48 }
```

```
dot20AnRabEnabled OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set to TRUE if this sector transmits RAB, and
         is set to FALSE otherwise"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 49 }

dot20AnOsiResponseMode OBJECT-TYPE
    SYNTAX        INTEGER {
        stochastic(1),
        deterministic(2)
    }
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the type of response to OSI modes. It is set to
0 for stochastic response and is set to 1 for deterministic response."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 50 }

dot20AnSlowInterferenceOffset OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set in units of dB"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 51 }

dot20AnCtrlAccessOffset OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field determines the initial gain of the R-CQICH over the
         R-ACH. The value of this parameter is -11+n dB, where n
         is the value of this parameter"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.3 (PowerControlGroup)"
    ::= { dot20AnSectorExtChanInfoEntry 52 }

dot20AnRlAuxPilotPower OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is determine the offset of R-AuxPICH with
         respect to R-PICH. This parameter may take the value 4+n."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 53 }

dot20AnReqQoSPowerBoost OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        " This parameter specifies a power boost value for r-reqch
transmissions on R-CDCCH for flows that allow request boost, and takes the
```

```
values 0, 1, 3, or 5 dB when the two-bit field is set to '00', '01', '10', or
'11', respectively.
 "
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.7.7.2.1 (PowerControl
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 54 }

dot20AnErasureTargetCtoI0 OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute's value (denoted n) determines the target C/I value of
erasure sequences for different assignment sizes. The
         transmit power is equal to n-12 dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 55 }

dot20AnErasureTargetCtoI1 OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute's value (denoted n) determines the target C/I value of
erasure sequences for different assignment sizes. The
         transmit power is equal to n-12 dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 56 }

dot20AnErasureTargetCtoI2 OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute's value (denoted n) determines the target C/I value of
erasure sequences for different assignment sizes. The
         transmit power is equal to n-12 dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 57 }

dot20AnErasureTargetCtoI3 OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute's value (denoted n) determines the target C/I value of
erasure sequences for different assignment sizes. The
         transmit power is equal to n-12 dB."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 8.8.9.2.3 (PowerParameters
         Attribute)"
    ::= { dot20AnSectorExtChanInfoEntry 58 }

dot20AnAccessCycleDuration OBJECT-TYPE
    SYNTAX        Integer32 (0..1)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the duration of the access
```

```
          cycle in units of Access Opportunities (as defined by the
          Physical Layer.)"
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 59 }

dot20AnMaxProbesPerSequence OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
         "This attribute (denoted n) determines the maximum number of probe
          sequences that can be part of one access sequence.  The
          number of probes is n+2"
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 60 }

dot20AnProbeRampUpStepSize OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
         "This attribute's value (denoted n) determines the power ramp
          up used for probes within a probe sequence and indicates
          a ramp up value of 2*(1+n) dB."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 61 }

dot20AnPilotThreshold1 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
         "This attribute's value (denoted n) determines
          PilotThreshold1 used by the Access Channel MAC Protocol. The
          value is -10 + 2n dB."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 62 }

dot20AnPilotThreshold2 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
         "This attribute's value (denoted n) determines
          PilotThreshold2 used by the Access Channel MAC Protocol. The
          value is -2n dB."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 63 }

dot20AnOpenLoopAdjust OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
         "This attribute's value (denoted n) determines the nominal
          power to be used by access terminal in the open loop power
          estimate. The value of nominal power is 70+n dB."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 64 }

dot20AnAccessRetryPersistance0 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal will set
         AccessRetryPersistence0 to 0."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 65 }

dot20AnAccessRetryPersistence1 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal will set
         AccessRetryPersistence1 to 0."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 66 }

dot20AnAccessRetryPersistence2 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal will set
         AccessRetryPersistence2 to 0."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 67 }

dot20AnAccessRetryPersistence3 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal sets
         AccessRetryPersistence3 to 0."
    REFERENCE
          "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
           Group)"
    ::= { dot20AnSectorExtChanInfoEntry 68 }
```

```
dot20AnAccessRetryPersistence4 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal sets
         AccessRetryPersistence4 to 0."
    REFERENCE
         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
          Group)"
    ::= { dot20AnSectorExtChanInfoEntry 69 }

dot20AnAccessRetryPersistence5 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal sets
         AccessRetryPersistence5 to 0."
    REFERENCE
         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
          Group)"
    ::= { dot20AnSectorExtChanInfoEntry 70 }

dot20AnAccessRetryPersistence6 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<7), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal sets
         AccessRetryPersistence6 to 0."
    REFERENCE
         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
          Group)"
    ::= { dot20AnSectorExtChanInfoEntry 71 }

dot20AnAccessRetryPersistence7 OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute determines the persistence probability for
         determining access sequence backoff. If this attribute's value
         is set to n (n<0), the access terminal will use 2^(-n/2) as the
         retry persistence.  For n=7, the access terminal sets
         AccessRetryPersistence7 to 0."
    REFERENCE
         "IEEE Std. 802.20-2008, Subclause 11.6.5.4.4 (AccessParameters
          Group)"
    ::= { dot20AnSectorExtChanInfoEntry 72 }

dot20AnSectorExtChanRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
```

```
         instances of the columnar objects in the SectorExtChanInfo
         Table. If the implementer of this MIB has chosen not to
         implement 'dynamic assignment' of sectors, this attribute is
         not useful and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnSectorExtChanInfoEntry 73 }

dot20AnSectorParamTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorParamEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 carrier of a sector for
         a specific ChannelBand. This table's attributes specify the
         configuration of the corresponding sector and can be used to
         populate fields in the SectorParameters message."
    ::= { dot20AnOverheadMessages 3 }

dot20AnSectorParamEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorParamEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the SectorParam table. This table
         is indexed by ifIndex. ifIndex: Each IEEE 802.20 interface
         (uniquely identified by SectorID) is represented by an
         ifEntry."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    INDEX
        { ifIndex }
    ::= { dot20AnSectorParamTable 1 }

dot20AnMobileCountryCode OBJECT-TYPE
    SYNTAX        Integer32 (0..4096)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the three digit Mobile Country
         Code associated with this sector (as specified in ITU-T
         Recommendation E.212, Identification Plan for Land Mobile
         Stations)."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 1 }

dot20AnMobileNetworkCode OBJECT-TYPE
    SYNTAX        Integer32 (0..4096)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This field is set three-digit BCD (binary coded
         decimal) encoded representation of the Mobile Network Code
         that has been assigned to the operator."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 2 }

dot20AnSectorID OBJECT-TYPE
    SYNTAX        OCTET STRING (SIZE(16))
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Sector Address Identifier. The access network sets the
         value of the SectorID according to the rules specified in 16.2.2.."
    REFERENCE
```

```
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters) and
Subclause 16.2.2 (SectorID Construction)"
    ::= { dot20AnSectorParamEntry 3 }

dot20AnChannelBandRef OBJECT-TYPE
    SYNTAX        Integer32
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The reference to the ChannelBand defined in ChannelBands table
         using this value as index (dot20AnChannelBandIndex)"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters,
         first instance), and Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnSectorParamEntry 4 }

dot20AnLatitude OBJECT-TYPE
    SYNTAX        Integer32 (-1296000..1296000)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The latitude of the sector. This attribute is set to
         this sector's latitude in units of 0.25 second, expressed as a
         two's complement signed number with positive numbers signifying
         North latitudes. Similarly, negative numbers signify South latitudes.
This attribute is set to a value in the
         range 1296000 to 1296000 inclusive (corresponding to a range of
         -90 to +90)."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 5 }

dot20AnLongitude OBJECT-TYPE
    SYNTAX        Integer32 (-2592000..2592000)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The longitude of the sector. This attribute is set to
         this sector's longitude in units of 0.25 second, expressed as a
         two's complement signed number with positive numbers signifying
         East longitude. Similarly, negative numbers signify West longitudes.
This attribute is set to a value in the
         range 2592000 to 2592000 inclusive (corresponding to a range of
         -180 degrees to +180 degrees)."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 6 }

dot20AnLeapSeconds OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The number of leap seconds that have occurred since the start
         of system time."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 7 }

dot20AnLocalTimeOffset OBJECT-TYPE
    SYNTAX        Integer32 (0..2047)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the offset of the local time
         from System Time. This value is in units of minutes,
         expressed as a two's complement signed number."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 8 }

dot20AnPrimaryRegZoneCode OBJECT-TYPE
    SYNTAX        Integer32
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The PrimaryRegistrationZoneCode for this sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 9 }

dot20AnAnGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Sector's AN Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.4.5.2.9 (AN Groups)"
    ::= { dot20AnSectorParamEntry 10 }

dot20AnPilotGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Sector's Pilot Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 11 }

dot20AnSynchronousGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Sector's Synchronous Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 12 }

dot20AnCellGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Sector's Cell Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorParamEntry 13 }

dot20AnSectorParamRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the SectorParam Table. If
         the implementer of this MIB has chosen not to implement
         'dynamic assignment' of sectors, this attribute is not useful
         and should return noSuchName upon SNMP request."
    DEFVAL          { active }
    ::= { dot20AnSectorParamEntry 14 }

dot20AnSectorGrpResSetsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorGrpResSetsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 sector and Forward
         Channel group resource set (see ExtendedChannelInfo)."
    ::= { dot20AnOverheadMessages 4 }

dot20AnSectorGrpResSetsEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorGrpResSetsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the
         AnSectorFwdChanGrpResourceSets table. This table is indexed
         by ifIndex and resourceSetId ifIndex: Each IEEE 802.20
         interface (uniquely identified by SectorID) is represented by
         an ifEntry."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    INDEX
        { ifIndex, dot20AnResourceSetId }
    ::= { dot20AnSectorGrpResSetsTable 1 }

dot20AnResourceSetId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Index of the forward channel group resource set for a
         particular sector."
    ::= { dot20AnSectorGrpResSetsEntry 1 }

dot20AnResourceSetBitmap OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The j'th bit of this field is set to 1 if a frame with
         frame index mod InterlaceDepth = j contains a subzone that
         corresponds to this resource set. If the InterlaceDepth = 6,
         the last two bits of this field is set to 0"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    ::= { dot20AnSectorGrpResSetsEntry 2 }

dot20AnBRCHSubzoneCyclingEnabled OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This parameter is set to TRUE if BRCHSubzoneCycling is enabled
         on this sector. For BRCH resource set with  BRCHSubzoneCycling
         disabled or DRCH resource set, the first subzone offset on all
         interlaces where this resource set is present is set to
         the ResourceSubzoneOffset. For BRCH resource sets with
         BRCHSubzoneCycling enabled, the offset of the first subzone
         over each interlace is shifted cyclically. Since the offset of
         first subzone over the lowest indexed interlace is defined by
         ResourceSubzoneOffset, the offset of the first subzone in the
         next interlace, where the resource set is present, is increased
         by 1 mod NumBRCHSubzones"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    ::= { dot20AnSectorGrpResSetsEntry 3 }

dot20AnResourceSetSubZoneSpacing OBJECT-TYPE
    SYNTAX         Integer32 (0..3)
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
        "This field indicates the spacing between subzones in a
         resource set. Subzones belonging to a resource group on an
         interlace is equally spaced, where the first subzone is
         defined by ResourceSubzoneOffset and
         BRCHSubzoneCyclingEnabled"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    ::= { dot20AnSectorGrpResSetsEntry 4 }

dot20AnNumResourceSubzones OBJECT-TYPE
    SYNTAX         Integer32 (0..31)
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
        "This field determines the number of subzones in each
         interlace where the resource set is present. An interlace is
         defined as the set of frames that have the same Frame Index mod
         InterlaceDepth, where InterlaceDepth is defined by
         ResourceSetInterlace. This parameter takes the value n+1."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    ::= { dot20AnSectorGrpResSetsEntry 5 }

dot20AnResourceSubzoneOffset OBJECT-TYPE
    SYNTAX         Integer32 (0..31)
    MAX-ACCESS     read-write
    STATUS         current
    DESCRIPTION
        "This field is set to the first subzone on the lowest
         indexed interlace that is part of a resource set. "
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.7.5.3
         (SupplementalConfigAssignment)"
    ::= { dot20AnSectorGrpResSetsEntry 6 }

dot20AnResourceSetRowStatus OBJECT-TYPE
    SYNTAX         RowStatus
    MAX-ACCESS     read-create
    STATUS         current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the
         SectorFwdChanGrpResourceSet Table. If the implementor of this
         MIB has chosen not to implement 'dynamic assignment' of
         sectors, this attribute is not useful and should return
         noSuchName upon SNMP request."
    DEFVAL          { active }
    ::= { dot20AnSectorGrpResSetsEntry 7 }

dot20AnSecondaryRegZoneCodeTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF Dot20AnSecondaryRegZoneCodeEntry
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "This table provides one row per 802.20 interface and per
         secondary registration zone code."
    ::= { dot20AnOverheadMessages 5 }

dot20AnSecondaryRegZoneCodeEntry OBJECT-TYPE
    SYNTAX        Dot20AnSecondaryRegZoneCodeEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the SecondaryRegZoneCode table,
         which is used to trigger registration for paging.  This table
         is indexed by IfIndex and dot20AnSecondaryRegZoneCodeIndex.
         ifIndex: Each IEEE 802.20 interface (uniquely identified by
         SectorID) is represented by an ifEntry."

    INDEX
        { ifIndex, dot20AnSecondaryRegZoneCodeIndex }
    ::= { dot20AnSecondaryRegZoneCodeTable 1 }

dot20AnSecondaryRegZoneCodeIndex OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Index of the secondary registration zone code for a particular
         sector."
    ::= { dot20AnSecondaryRegZoneCodeEntry 1 }

dot20AnSecRegZoneCode OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "One of the SecondaryRegistrationZoneCode for this sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSecondaryRegZoneCodeEntry 2 }

dot20AnSecondaryRegZoneRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the SecondaryRegZoneCode
         Table. If the implementor of this MIB has chosen not to
         implement 'dynamic assignment' of sectors, this attribute is
         not useful and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnSecondaryRegZoneCodeEntry 3 }

dot20AnSectorIpsiTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorIpsiEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 interface and per
         IPSI."
    ::= { dot20AnOverheadMessages 6 }

dot20AnSectorIpsiEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorIpsiEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
```

```
        "An Entry (conceptual row) in the SectorIpsi table, which is a
         list of personalities supported by the given sector. This table
         is indexed by IfIndex and dot20AnIpsiIndex. ifIndex: Each IEEE
         802.20 interface (uniquely identified by SectorID) is
         represented by an ifEntry."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    INDEX
        { ifIndex }
    ::= { dot20AnSectorIpsiTable 1 }

dot20AnIpsiIndex OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Index of an Ipsi supported by a particular sector."
    ::= { dot20AnSectorIpsiEntry 1 }

dot20AnSupportedIpsi OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "IPSI supported by a particular sector"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnSectorIpsiEntry 2 }

dot20AnIpsiRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the SectorIpsi Table. If
         the implementor of this MIB has chosen not to implement
         'dynamic assignment' of sectors, this attribute is not useful
         and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnSectorIpsiEntry 3 }

dot20AnSectorCdmaSubSegTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorCdmaSubSegEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 sector, interlace and
         Reverse Channel group CDMA Sub segment (see ExtendedChannelInfo
         message in AIS)."
    ::= { dot20AnOverheadMessages 8 }

dot20AnSectorCdmaSubSegEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorCdmaSubSegEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the AnSectorCdmaSubSeg table.
         This table is indexed by ifIndex, interlaceId and
         CDMASubSegmentId."

    INDEX
        { ifIndex, dot20AnInterlaceId }
    ::= { dot20AnSectorCdmaSubSegTable 1 }

dot20AnInterlaceId OBJECT-TYPE
```

```
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Interlace Id"
    ::= { dot20AnSectorCdmaSubSegEntry 1 }

dot20AnCdmaSubSegmentNum OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Number of reverse channel CDMA Sub segment within an interlace
         for a particular sector."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.4.2 (ReverseChannelGroup)"
    ::= { dot20AnSectorCdmaSubSegEntry 2 }

dot20AnSectorCdmaSubSegRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the SectorCdmaSubSeg
         Table. If the implementor of this MIB has chosen not to
         implement 'dynamic assignment' of sectors, this attribute is
         not useful and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnSectorCdmaSubSegEntry 3 }

dot20AnChannelBandsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnChannelBandsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 ChannelBand. This
         table's attributes specify the ChannelBand record of a
         particular ChannelBand which may be used for a sector defined
         in the SectorConfig table, or by a member of the neighbor list
         defined in NeighborSectorsTable."
    ::= { dot20AnOverheadMessages 9 }

dot20AnChannelBandsEntry OBJECT-TYPE
    SYNTAX        Dot20AnChannelBandsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the ChannelBands table. The
         Channel Bands table is referenced by the NeighborSectorsTable
         or Sector Table.  This table is indexed by ChannelBandIndex."

    INDEX
        { dot20AnChannelBandIndex }
    ::= { dot20AnChannelBandsTable 1 }

dot20AnChannelBandIndex OBJECT-TYPE
    SYNTAX        Integer32 (1..2147483647)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Index of the ChannelBand within the ChannelBands table."
    ::= { dot20AnChannelBandsEntry 1 }

dot20AnSystemType OBJECT-TYPE
    SYNTAX        Integer32 (0..2)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute discriminates between the different ChannelBand
         Records."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 2 }

dot20AnBandClass OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the band class number
         corresponding to the frequency assignment of the ChannelBand
         specified by this record."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 3 }

dot20AnChannelNumber OBJECT-TYPE
    SYNTAX        Integer32 (0..65535)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the Channel number
         corresponding to the frequency assignment of the ChannelBand
         specified by this record."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 4 }

dot20AnHalfDuplexSupported OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to TRUE if half duplex operation
         is supported in this system."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 5 }
<Edited to here - 2008-11-07 1539>
dot20AnReverseChannelBandClass OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the band class number
         corresponding to the frequency assignment of the reverse
         ChannelBand specified by this record."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters),
         and Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 6 }

dot20AnReverseChannelNumber OBJECT-TYPE
    SYNTAX        Integer32 (0..65535)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the Channel number
         corresponding to the frequency assignment of the Reverse
         ChannelBand specified by this record."
    REFERENCE
```

```
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters),
         and Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 7 }

dot20AnCyclicPrefixLength OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the cyclic prefix length,
         i.e. it is set to the quantity (N_CP-1) from the Physical
         Layer."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record),
         and Table 165 (Specification for the u Parameter)"
    ::= { dot20AnChannelBandsEntry 8 }

dot20AnFFTSize OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to log_2(N_FFT/128)."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 9 }

dot20AnCBNumGuardSubcarriers OBJECT-TYPE
    SYNTAX        Integer32 (0..63)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the number of guard subcarriers
         for the system on the forward channel."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnChannelBandsEntry 10 }

dot20AnChannelBandShortId OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute identifies the two bit index that identifies
         this channel in beacon transmissions"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnChannelBandsEntry 11 }

dot20AnChannelBandAccessHashMask OBJECT-TYPE
    SYNTAX        Integer32 (0..65536)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "this attribute is set to the AccessHashingChannelMask for this
         channel"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnChannelBandsEntry 12 }

dot20AnChannelBandStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
```

```
         instances of the columnar objects in the  ChannelBands Table.
         If the implementor of this MIB has chosen not to implement
         'dynamic assignment' of ChannelBands, this attribute is not
         useful and should return noSuchName upon SNMP request."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    DEFVAL         { active }
    ::= { dot20AnChannelBandsEntry 13 }

dot20AnNeighborSectorsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnNeighborSectorsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per 802.20 neighbor sector. This
         table's attributes specify the sector parameters of a
         particular neighbor sector which may be used as a neighbor to
         one sector defined in the SectorConfig table."
    ::= { dot20AnOverheadMessages 10 }

dot20AnNeighborSectorsEntry OBJECT-TYPE
    SYNTAX        Dot20AnNeighborSectorsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the AnNeighborSectors table. This
         table is indexed by ChannelBandIndex, NeighborSectorIndex."
    INDEX
        { dot20AnChannelBandIndex, dot20AnNeighborSectorIndex }
    ::= { dot20AnNeighborSectorsTable 1 }

dot20AnNeighborSectorIndex OBJECT-TYPE
    SYNTAX        Integer32 (1..2147483647)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Index of the Neighbor Sector for this Neighbor Carrier within
         the ChannelBand."
    ::= { dot20AnNeighborSectorsEntry 1 }

dot20AnNeighborPilotID OBJECT-TYPE
    SYNTAX        Integer32 (0..1023)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the PilotID of a neighboring
         sector that the access terminal should add to its Neighbor
         Set."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 5.3.2.1 (PilotPN and PilotPhase)"
    ::= { dot20AnNeighborSectorsEntry 2 }

dot20AnNeighborEffTransmitPower OBJECT-TYPE
    SYNTAX        Integer32 (0..63)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the transmit power of the
         sector in units of dBm."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 3 }

dot20AnNeighborChannelBandRef OBJECT-TYPE
    SYNTAX        Integer32
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The reference to the ChannelBand defined in ChannelBands table
         (dot20AnChannelBandIndex)"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 4 }

dot20AnNeighborChannelShortID OBJECT-TYPE
    SYNTAX        Integer32 (0..3)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Neighbor Sector's short Channel ID"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 5 }

dot20AnNeighborSameANAsPrimSect OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Set true if same access network as primary sector."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 6 }

dot20AnNeighborSectorPilotGrpId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Neighbor Sector's Pilot Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 7 }

dot20AnNeighborSynchGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Neighbor Sector's Synchronous Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 8 }

dot20AnNeighborSectorCellGroupId OBJECT-TYPE
    SYNTAX        Integer32 (0..7)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "Neighbor Sector's Cell Group Id"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnNeighborSectorsEntry 9 }

dot20AnNeighborSectorStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the NeighborSectors
         Table. If the implementor of this MIB has chosen not to
         implement 'dynamic assignment' of neighbor sectors this
         attribute is not useful and should return noSuchName upon SNMP
         request."
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    DEFVAL         { active }
    ::= { dot20AnNeighborSectorsEntry 10 }

dot20AnOtherTechNghbrsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnOtherTechNghbrsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row per other technology neighbor
         channel. This table's attributes specify the technology type
         and neighbor list of a particular neighbor channel which may be
         used by one sector defined in the SectorConfig table for
         inter-technology handoff."
    ::= { dot20AnOverheadMessages 11 }

dot20AnOtherTechNghbrsEntry OBJECT-TYPE
    SYNTAX        Dot20AnOtherTechNghbrsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the AnOtherTechNghbrs table. This
         table is indexed by Sector (ifIndex) and OtherTechnologyIndex"
    INDEX
        { ifIndex, dot20AnOtherTechnologyIndex }
    ::= { dot20AnOtherTechNghbrsTable 1 }

dot20AnOtherTechnologyIndex OBJECT-TYPE
    SYNTAX        Integer32 (1..2147483647)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "The neighbor other technology entry index"
    ::= { dot20AnOtherTechNghbrsEntry 1 }

dot20AnTechnologyType OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the type of other technology.
         Interpretation for its value should as defined in the AIS
         spec."
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnOtherTechNghbrsEntry 2 }

dot20AnTechNghbrListLength OBJECT-TYPE
    SYNTAX        Integer32 (0..255)
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set the length, in bytes, of the
         neighbor list information for the other technology."
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnOtherTechNghbrsEntry 3 }

dot20AnTechnologyNeighborList OBJECT-TYPE
    SYNTAX        OCTET STRING (SIZE(256))
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "This attribute is set to the neighbor list information
         for the other technology."
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 11.6.5.5 (SectorParameters)"
    ::= { dot20AnOtherTechNghbrsEntry 4 }

dot20AnOtherTechNghbrRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the OtherTechNghbrs Table.
         If the implementor of this MIB has chosen not to implement
         'dynamic assignment' of other technology neighbors, this
         attribute is not useful and should return noSuchName upon SNMP
         request."
    DEFVAL         { active }
    ::= { dot20AnOtherTechNghbrsEntry 5 }

dot20AnNeighborListTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnNeighborListEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table defines the neighbor lists for the sectors defined
         in the SectorConfig table. Each row in this table indexed per
         sector (ifIndex) specifies a pointer to a neighbor sector of
         this sector."
    ::= { dot20AnOverheadMessages 12 }

dot20AnNeighborListEntry OBJECT-TYPE
    SYNTAX        Dot20AnNeighborListEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the AnNeighborList table. This
         table is indexed by Sector (ifIndex) and NeighborIndex indexing
         each neighbor sector for a particular Sector."
    INDEX
        { ifIndex, dot20AnNeighborIndex }
    ::= { dot20AnNeighborListTable 1 }

dot20AnNeighborIndex OBJECT-TYPE
    SYNTAX        Integer32 (1..32)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This index identifies one neighbor sector for a Sector."
    ::= { dot20AnNeighborListEntry 1 }

dot20AnNeighborSectorPointer OBJECT-TYPE
    SYNTAX        RowPointer
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "This attribute points to an instance of sector in SectorConfig
         table or in NeighborSectors table. This sector is defined as a
         neighbor of the sector identified by the ifIndex of this
         attribute's entry."
    ::= { dot20AnNeighborListEntry 2 }

dot20AnNeighborRowStatus OBJECT-TYPE
    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "The status column used for creating, modifying, and deleting
         instances of the columnar objects in the  NeighborList Table.
         If the implementor of this MIB has chosen not to implement
         'dynamic assignment' of neighbor list entries this attribute is
         not useful and should return noSuchName upon SNMP request."
    DEFVAL         { active }
    ::= { dot20AnNeighborListEntry 3 }

dot20AnSectorToIfIndexTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20AnSectorToIfIndexEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table can be used to find the ifIndex of an 802.20
         interface based on its SectorID and ChannelBand information
         (reverse mapping of the Sector Config table)."
    ::= { dot20An 2 }

dot20AnSectorToIfIndexEntry OBJECT-TYPE
    SYNTAX        Dot20AnSectorToIfIndexEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the AnSectorToIfIndex table."
    INDEX
        { dot20AnSectorID, ifIndex }
    ::= { dot20AnSectorToIfIndexTable 1 }

dot20AnIfChannelBandRef OBJECT-TYPE
    SYNTAX        Integer32
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The reference to the ChannelBand defined in ChannelBands table
         (dot20AnChannelBandIndex)"
    REFERENCE
        "IEEE Std. 802.20-2008, Subclause 11.6.5.5 (SectorParameters,
         first instance), and Subclause 15.2.1 (ChannelBand Record)"
    ::= { dot20AnSectorToIfIndexEntry 1 }

dot20Cmn OBJECT-IDENTITY
    STATUS        current
    DESCRIPTION
        "Common configuration and statistics."
    ::= { ieee802dot20 2 }

dot20CmnMac OBJECT-IDENTITY
    STATUS        current
    DESCRIPTION
        "MAC layer objects"
    ::= { dot20Cmn 1 }

dot20CmnSessionControl OBJECT IDENTIFIER ::= { dot20CmnMac 1 }

dot20CmnSessionMgtProtocol OBJECT IDENTIFIER ::= { dot20CmnSessionControl 1 }

dot20CmnSessionOpenCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of sessions opened"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
        Protocol State Diagram (Access Network))"
    ::= { dot20CmnSessionMgtProtocol 1 }

dot20CmnSessionCloseCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of sessions closed"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
        Protocol State Diagram (Access Network))"
    ::= { dot20CmnSessionMgtProtocol 2 }

dot20CmnSessionFailureCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of session open/close failures"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 159 (Basic Session Control
        Protocol State Diagram (Access Network))"
    ::= { dot20CmnSessionMgtProtocol 3 }

dot20CmnConnectionControl OBJECT IDENTIFIER ::= { dot20CmnMac 3 }

dot20CmnConnectedState OBJECT IDENTIFIER ::= { dot20CmnConnectionControl 1 }

dot20CmnActiveConnectionCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of current active connections (in Open state.)"
    REFERENCE
        "IEEE Std 802.20-2008, Figures 152 and 153"
    ::= { dot20CmnConnectedState 1 }

dot20CmnConnectionAttemptCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of connection attempts (i.e. that reached BindATI state.)"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 152 (Basic Connected State
         Protocol State Diagram (AT)) and Figure 153 (Basic Connected
         State Protocol State Diagram (AN))"
    ::= { dot20CmnConnectedState 2 }

dot20CmnConnectionFailureCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of connection failures during connection attempt (i.e.
         That reached BindATI state without reaching Open state,
         through timeout or deactivation"
    REFERENCE
        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
         State Diagram (AT)) and 153 (Connected State Protocol State
         Diagram (AN))"
    ::= { dot20CmnConnectedState 3 }

dot20CmnConnectionDropCounts OBJECT-TYPE
```

```
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of dropped connections (via a command of
         ConnectedState.Close) after a connection has been established."
    REFERENCE
        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
         State Diagram (AT)) and 153 (Connected State Protocol State
         Diagram (AN))"
    ::= { dot20CmnConnectedState 4 }

dot20CmnConnectionReleaseCounts OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of connection release (Tx ConnectionClose or
         Rx ConnectionClose) after a connection has been established."
    REFERENCE
        "IEEE Std 802.20-2008, Figures 152 (Connect State Protocol
         State Diagram (AT)) and 153 (Connected State Protocol State
         Diagram (AN))"
    ::= { dot20CmnConnectedState 5 }

dot20CmnRadioLink OBJECT IDENTIFIER ::= { dot20CmnMac 4 }

dot20CmnRlp OBJECT IDENTIFIER ::= { dot20CmnRadioLink 2 }

dot20CmnRlpStatsTable OBJECT-TYPE
    SYNTAX         SEQUENCE OF Dot20CmnRlpStatsEntry
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "This table provides one row of Radio Link Protocol statistics
         per 802.20 interface"
    ::= { dot20CmnRlp 1 }

dot20CmnRlpStatsEntry OBJECT-TYPE
    SYNTAX         Dot20CmnRlpStatsEntry
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "An Entry (conceptual row) in the RlpStats table. This table is
         indexed by IfIndex and dot20StreamId."
    INDEX
        { ifIndex, dot20CmnStreamId }
    ::= { dot20CmnRlpStatsTable 1 }

dot20CmnStreamId OBJECT-TYPE
    SYNTAX         Integer32 (0 .. 31)
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "Stream Id"
    ::= { dot20CmnRlpStatsEntry 1 }

dot20CmnRlpTxBytes OBJECT-TYPE
    SYNTAX         Counter64
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Number of RLP bytes of payload transmitted"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 2 }

dot20CmnRlpReTxBytes OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP bytes of payload retransmitted"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 3 }

dot20CmnRlpTxDropBytes OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP bytes of dropped before transmission"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 4 }

dot20CmnRlpTxStatus OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP ReceiverStatus messages transmitted"
    REFERENCE
        "IEEE 802.20-2008, Subclause 7.3.4.3.3.5 (ATReceiverStatus),
         and Subclause 7.3.4.3.3.7 (ANReceiverStatus)"
    ::= { dot20CmnRlpStatsEntry 5 }

dot20CmnRlpRxBytes OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP bytes of payload received"
    ::= { dot20CmnRlpStatsEntry 6 }

dot20CmnRlpRxStatus OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP ReceiverStatus messages received"
    REFERENCE
        "IEEE 802.20-2008, Subclause 7.3.4.3.3.5 (ATReceiverStatus),
         and Subclause 7.3.4.3.3.7 (ANReceiverStatus)"
    ::= { dot20CmnRlpStatsEntry 7 }

dot20CmnRlpTxPackets OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP Packets transmitted"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 8 }

dot20CmnRlpReTxPackets OBJECT-TYPE
```

```
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP Packets retransmitted"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 9 }

dot20CmnRlpTxrDropPackets OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP Packets dropped before transmission"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 10 }

dot20CmnRlpRxPackets OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of RLP Packets received"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.3 (RLP Receive
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 11 }

dot20CmnRlpTxNAKTimeouts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of NAK Timeouts"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.3 (RLP Receive
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 12 }

dot20CmnRlpTxACKTimeouts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of ACK Timeouts"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.3.3.4.2 (RLP Transmit
         Procedures)"
    ::= { dot20CmnRlpStatsEntry 13 }

dot20CmnQmp OBJECT-IDENTITY
    STATUS        current
    DESCRIPTION
        "Qos Management Protocol"
    ::= { dot20CmnRadioLink 3 }

dot20CmnQmpStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20CmnQmpStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row of QMP statistics per 802.20
```

```
         interface"
    ::= { dot20CmnQmp 2 }

dot20CmnQmpStatsEntry OBJECT-TYPE
    SYNTAX        Dot20CmnQmpStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the QmpStats table. This table is
         indexed by IfIndex. ifIndex: Each IEEE 802.20 interface is
         represented by an ifEntry."
    INDEX
        { ifIndex }
    ::= { dot20CmnQmpStatsTable 1 }

dot20CmnActiveReservationsCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Active (Open State) Reservations"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
         State Diagram (AT)), and Figure 22 (Forward Link Reservation State
         Diagram (AN)"
    ::= { dot20CmnQmpStatsEntry 1 }

dot20CmnIdleReservationsCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Idle (Close State) Reservations"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
         State Diagram (AT)), and Figure 22 (Forward Link Reservation State
         Diagram (AN)"
    ::= { dot20CmnQmpStatsEntry 2 }

dot20CmnReservationOpenCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Reservations Open requests"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
         State Diagram (AT)), Figure 22 (Forward Link Reservation State
         Diagram (AN), Subclause 7.2.3.3.1 (ReservationOnRequest), and
         Subclause 7.2.3.3.6 (RevReservationOn)"
    ::= { dot20CmnQmpStatsEntry 3 }

dot20CmnReservationCloseCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Reservations Close requests"
    REFERENCE
        "IEEE Std 802.20-2008, Figure 21 (Reverse Link Reservation
         State Diagram (AT)), Figure 22 (Forward Link Reservation State
         Diagram (AN), Subclause 7.2.3.3.2 (ReservationOffRequest),
         and Subclause 7.2.3.3.7 (RevReservationOn)"
    ::= { dot20CmnQmpStatsEntry 4 }

dot20CmnReservationFailCounts OBJECT-TYPE
```

```
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Failed Reservations requests"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 7.2.3.3.5 (ReservationReject)"
    ::= { dot20CmnQmpStatsEntry 5 }

dot20CmnSecurity OBJECT IDENTIFIER ::= { dot20CmnMac 5 }

dot20CmnKeyExchangeProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 1 }

dot20CmnKeyExchangeAttemptCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of key exchanges attempts"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 10.4.5.2.1 (KeyRequest)"
    ::= { dot20CmnKeyExchangeProtocol 1 }

dot20CmnKeyExchangeFailureCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of key exchanges failures"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 10.4.5.2.4 (KeyReject)"
    ::= { dot20CmnKeyExchangeProtocol 2 }

dot20CmnMessageIntegrityProtocol OBJECT IDENTIFIER ::= { dot20CmnSecurity 2 }

dot20CmnAuthStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20CmnAuthStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row of Authentication statistics per
         802.20 interface (i.e. sector for a specific ChannelBand.)"
    ::= { dot20CmnMessageIntegrityProtocol 1 }

dot20CmnAuthStatsEntry OBJECT-TYPE
    SYNTAX        Dot20CmnAuthStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Authentication statistics per 802.20 interfaces"
    INDEX
        { ifIndex }
    ::= { dot20CmnAuthStatsTable 1 }

dot20CmnAuthFailureCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Authentication failures (i.e. failure code 0x03 for
         RouteOpenReject.)"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 13.2.6.2.1
         (RouteOpenRequest), and Subclause 13.2.6.12
         (RouteOpenReject)"
    ::= { dot20CmnAuthStatsEntry 1 }
```

```
dot20CmnAuthSuccessCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of successful Authentications"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 13.2.6.2.1
         (RouteOpenRequest), and Subclause 13.2.6.3
         (RouteOpenAccept)"
    ::= { dot20CmnAuthStatsEntry 2 }

dot20CmnLowerMAC OBJECT IDENTIFIER ::= { dot20CmnMac 6 }

dot20CmnLMACPacketStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20CmnLMACPacketStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row of Lower MAC protocol statistics
         per 802.20 interface, packet format and nb of ARQ attempts
         needed in order to successfully transmit/receive a packet."
    ::= { dot20CmnLowerMAC 1 }

dot20CmnLMACPacketStatsEntry OBJECT-TYPE
    SYNTAX        Dot20CmnLMACPacketStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the LMACPacketStats table. This
         table is indexed by IfIndex, PacketFormatIndex and
         ARQAttemptsIndex."
    INDEX
        { ifIndex, dot20CmnPacketFormatIndex, dot20CmnARQAttemptsIndex
        }
    ::= { dot20CmnLMACPacketStatsTable 1 }

dot20CmnPacketFormatIndex OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "The packet format index as defined in 802.20 AIS spec."
    ::= { dot20CmnLMACPacketStatsEntry 1 }

dot20CmnARQAttemptsIndex OBJECT-TYPE
    SYNTAX        Integer32 (0..15)
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "Number of ARQ attempts that were needed in order to transmit
         or receive a packet. Index 0 means that the packets failed to
         be transmitted/received."
    ::= { dot20CmnLMACPacketStatsEntry 2 }


dot20CmnFwdTxPacketCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of transmitted packets"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 8.6.5.5.2.2 (F-DCH TX Associated
         with Persistent Assignments), Subclause 8.6.5.5.2.3 (F-DCH TX
         Associated with Non-Persistent Assignments and Residual Resource
         Assignments), and Subclause 8.6.5.5.2.4 (F-DCH TX Associated with
         Group Resource Assignments)"
    ::= { dot20CmnLMACPacketStatsEntry 3 }

dot20CmnRevRxPacketCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of received packets"
    REFERENCE
        "IEEE Std 802.20-2008,
         Subclause 8.6.5.5.1.2.2 (AT Processing for Non-Persistent
Assignments),
         Subclause 8.6.5.5.1.2.3 (AT Processing for Residual Resource
Assignments),
         Subclause 8.6.5.5.1.2.4 (AT Processing for Group Resource
Assignments)"
    ::= { dot20CmnLMACPacketStatsEntry 4 }

dot20CmnLMACStatsTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF Dot20CmnLMACStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "This table provides one row of Lower MAC protocol statistics
         per 802.20 interface and packet formats."
    ::= { dot20CmnLowerMAC 2 }

dot20CmnLMACStatsEntry OBJECT-TYPE
    SYNTAX        Dot20CmnLMACStatsEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "An Entry (conceptual row) in the LMACStats table. This table
         is indexed by IfIndex, PacketFormatIndex."
    INDEX
        { ifIndex, dot20CmnPacketFormatIndex }
    ::= { dot20CmnLMACStatsTable 1 }

dot20CmnFLABCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Forward Link Assignment Blocks"
    REFERENCE
        "IEEE Std 802.20-2008, Table 44 (F-SCCH Blocks), and Subclause
         8.5.5.4.1.2 (Framing of F-SCCH Blocks)"
    ::= { dot20CmnLMACStatsEntry 1 }

dot20CmnRLABCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Reverse Link Assignment Block"
    REFERENCE
        "IEEE Std 802.20-2008, Table 44 (F-SCCH Blocks), and Subclause
         8.5.5.4.1.2 (Framing of F-SCCH Blocks), and Subclause
         8.5.5.3.1.1.3.3 (RLAB)"
    ::= { dot20CmnLMACStatsEntry 2 }

dot20CmnAccessGrantCounts OBJECT-TYPE
    SYNTAX        Counter64
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Number of Access Grants (the number of times the indication
         ForwardLinkControlSegmentMAC.AccessGrantSent is raised)"
    REFERENCE
        "IEEE Std 802.20-2008, Subclause 8.5.5.4.1.1.3.1.1 (Procedures
         for Sending an Access Grant)"
    ::= { dot20CmnLMACStatsEntry 3 }

dot20Conformance OBJECT IDENTIFIER ::= { ieee802dot20 4 }

dot20Groups OBJECT IDENTIFIER ::= { dot20Conformance 1 }

dot20CmnSessionMgtPGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnSessionCloseCounts, dot20CmnSessionFailureCounts,
        dot20CmnSessionOpenCounts }
    STATUS        current
    DESCRIPTION
        "The session management protocol statistics"
    ::= { dot20Groups 1 }

dot20CmnKeyExchangePGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnKeyExchangeAttemptCounts,
        dot20CmnKeyExchangeFailureCounts }
    STATUS        current
    DESCRIPTION
        "The key exchange protocol statistics"
     ::= { dot20Groups 4 }

dot20CmnConnectedStatePGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnActiveConnectionCounts,
        dot20CmnConnectionAttemptCounts, dot20CmnConnectionDropCounts,
        dot20CmnConnectionFailureCounts, dot20CmnConnectionReleaseCounts
        }
    STATUS        current
    DESCRIPTION
         "The connected state protocol statistics"
    ::= { dot20Groups 5 }

dot20CmnRadioLinkGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnActiveReservationsCounts,
        dot20CmnIdleReservationsCounts, dot20CmnReservationCloseCounts,
        dot20CmnReservationFailCounts, dot20CmnReservationOpenCounts,
        dot20CmnRevRxPacketCounts, dot20CmnRlpReTxBytes,
        dot20CmnRlpReTxPackets, dot20CmnRlpRxBytes,
        dot20CmnRlpRxPackets, dot20CmnRlpRxStatus,
        dot20CmnRlpTxACKTimeouts, dot20CmnRlpTxBytes,
        dot20CmnRlpTxDropBytes, dot20CmnRlpTxNAKTimeouts,
        dot20CmnRlpTxPackets, dot20CmnRlpTxStatus,
        dot20CmnRlpTxrDropPackets }
    STATUS        current
    DESCRIPTION
         "The radio link layer statistics"
    ::= { dot20Groups 7 }

dot20CmnAuthGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnAuthFailureCounts, dot20CmnAuthSuccessCounts }
    STATUS        current
    DESCRIPTION
        "The authentication protocol statistics"
    ::= { dot20Groups 8 }

dot20CmnLowerMACGroup OBJECT-GROUP
    OBJECTS
        { dot20CmnAccessGrantCounts, dot20CmnFLABCounts,
        dot20CmnFwdTxPacketCounts, dot20CmnRLABCounts,
        dot20CmnRevRxPacketCounts }
    STATUS        current
    DESCRIPTION
        "The lower mac sublayer statistics"
    ::= { dot20Groups 9 }

dot20AnIdleStatePGroup OBJECT-GROUP
    OBJECTS
        { dot20AnAccessAttemptCounts, dot20AnAccessAttemptFailCounts,
        dot20AnPageAttemptCounts, dot20AnPageFailureCounts }
    STATUS        current
    DESCRIPTION
        "The An idle state protocol statistics"
    ::= { dot20Groups 10 }

dot20AnOverheadGroup OBJECT-GROUP
    OBJECTS
        { dot20An16QamScchT2PRatio, dot20AnAccessCycleDuration,
        dot20AnAccessRetryPersistance0, dot20AnAccessRetryPersistence1,
        dot20AnAccessRetryPersistence2, dot20AnAccessRetryPersistence3,
        dot20AnAccessRetryPersistence4, dot20AnAccessRetryPersistence5,
        dot20AnAccessRetryPersistence6, dot20AnAccessRetryPersistence7,
        dot20AnAckInterferenceOffset, dot20AnAnGroupId,
        dot20AnAssignmentAckHARQTx, dot20AnBRCHSubzoneCyclingEnabled,
        dot20AnBandClass, dot20AnCBNumGuardSubcarriers,
        dot20AnCDMAInterlacesBitmap, dot20AnCQIPilotTransmitPower,
        dot20AnCdmaSubSegmentNum, dot20AnCellGroupId, dot20AnCellNullID,
        dot20AnChannelBandAccessHashMask, dot20AnChannelBandRef,
        dot20AnChannelBandShortId, dot20AnChannelNumber,
        dot20AnCommonPilotTransmitPower, dot20AnCpichHoppingMode,
        dot20AnCtrlAccessOffset, dot20AnCyclicPrefixLength,
        dot20AnEffectiveTransmitPower, dot20AnEnableExpandedQPCH,
        dot20AnErasureTargetCtoI0, dot20AnErasureTargetCtoI1,
        dot20AnErasureTargetCtoI2, dot20AnErasureTargetCtoI3,
        dot20AnFACKBandwidthFactor, dot20AnFFTSize,
        dot20AnFDPICHCodeOffsetSubtree0, dot20AnFDPICHCodeOffsetSubtree1,
        dot20AnFDPICHCodeOffsetSubtree2, dot20AnFDPICHCodeOffsetSubtree3,
        dot20AnFLReservedInterlaces, dot20AnFastIoTEnabled,
        dot20AnFastOSIEnabled, dot20AnFlIotReportInterval,
        dot20AnFlPcReportInterval, dot20AnFlPqiReportInterval,
        dot20AnFlSdmaNumSubtrees, dot20AnFlSubzoneSize,
        dot20AnHalfDuplexModeSupported, dot20AnHalfDuplexSupported,
        dot20AnIfChannelBandRef, dot20AnLatitude, dot20AnLeapSeconds,
        dot20AnLocalTimeOffset, dot20AnLongitude, dot20AnMacIdRange,
        dot20AnMax16QamScchBlocks, dot20AnMaxNumLABs, dot20AnMaxNumSharedLABs,
        dot20AnMaxProbesPerSequence, dot20AnMinScchResourceIndex,
        dot20AnMobileCountryCode, dot20AnMobileNetworkCode,
        dot20AnNeighborPilotID, dot20AnNeighborChannelShortID,
        dot20AnNeighborSameANAsPrimSect, dot20AnNeighborSectorCellGroupId,
        dot20AnNeighborSectorPilotGrpId, dot20AnNeighborChannelBandRef,
        dot20AnNeighborSectorPointer, dot20AnNeighborSynchGroupId,
        dot20AnNeighborEffTransmitPower, dot20AnNumAckableLABs,
        dot20AnNumCmnPilotTxAnt, dot20AnNumCommonSegmentHopPorts,
        dot20AnNumDRCHSubzones, dot20AnNumEffectiveAntennas,
        dot20AnNumFLReservedSubzones,
        dot20AnNumGuardSubcarriers, dot20AnNumLABSegments,
        dot20AnNumOdcchReports, dot20AnNumRLCdmaSubsegments,
        dot20AnNumResourceSubzones, dot20AnNumSilenceIntervalSubzone,
        dot20AnOpenLoopAdjust, dot20AnOsiResponseMode,
        dot20AnPdCabResSharingEnabled, dot20AnPilotGroupId, dot20AnPilotID,
        dot20AnPilotThreshold1, dot20AnPilotThreshold2,
        dot20AnPrimaryRegZoneCode, dot20AnProbeRampUpStepSize,
        dot20AnRabEnabled, dot20AnRackBandwidthFactor,
        dot20AnReqQoSPowerBoost, dot20AnResourceChannelMuxMode,
        dot20AnResourceSetBitmap, dot20AnResourceSetSubZoneSpacing,
        dot20AnResourceSubzoneOffset, dot20AnReverseChannelBandClass,
        dot20AnReverseChannelNumber, dot20AnRlAuxPilotPower,
        dot20AnRlDpichCodeOffsetSubtree0, dot20AnRlDpichCodeOffsetSubtree1,
        dot20AnRlDpichCodeOffsetSubtree2, dot20AnRlDpichCodeOffsetSubtree3,
        dot20AnRlNumSdmaDimensions, dot20AnModSymbolsPerQPSKLAB,
        dot20AnSFNCellID, dot20AnSecRegZoneCode, dot20AnSectorID,
        dot20AnSilenceIntervalDuration, dot20AnSilenceIntervalPeriod,
        dot20AnSinglePAForXCarriers, dot20AnSlowInterferenceOffset,
        dot20AnSupportedIpsi, dot20AnSynchronousGroupId, dot20AnSystemType,
        dot20AnTechNghbrListLength, dot20AnTechnologyNeighborList,
        dot20AnTechnologyType, dot20AnTotalNumSubcarriers,
        dot20AnUseDrchForFlcs, dot20AnRlSubzoneSize }
    STATUS        current
    DESCRIPTION "The overhead messages protocol configuration"
    ::= { dot20Groups 11 }

dot20AnOverheadGroup2 OBJECT-GROUP
    OBJECTS
        { dot20AnChannelBandStatus, dot20AnIpsiRowStatus,
        dot20AnNeighborRowStatus, dot20AnNeighborSectorStatus,
        dot20AnOtherTechNghbrRowStatus, dot20AnResourceSetRowStatus,
        dot20AnSecondaryRegZoneRowStatus,
        dot20AnSectorCdmaSubSegRowStatus, dot20AnSectorConfigRowStatus,
        dot20AnSectorExtChanRowStatus, dot20AnSectorParamRowStatus }
    STATUS        current
    DESCRIPTION
        "If the MIB is created with pre-configured sector list tables and
         neighbor list tables, this Overhead Group is unnecessary.  Otherwise,
         these items are used to add rows to these tables in the MIB, so
         that additional sectors and/or neighbors can be added after MIB
         creation, through SNMPv2."
    ::= { dot20Groups 12 }

dot20Compliances OBJECT IDENTIFIER ::= { dot20Conformance 2 }

dot20AnCompliance MODULE-COMPLIANCE
    STATUS         current
    DESCRIPTION
        "The compliance statement for SNMPv2 entities that implement
         the IEEE 802.20 MIB for the An."
    MODULE         IEEE802dot20-MIB
        MANDATORY-GROUPS
            { dot20AnIdleStatePGroup, dot20AnOverheadGroup,
            dot20CmnAuthGroup, dot20CmnConnectedStatePGroup,
            dot20CmnKeyExchangePGroup, dot20CmnLowerMACGroup,
            dot20CmnRadioLinkGroup, dot20CmnSessionMgtPGroup }
        GROUP          dot20AnOverheadGroup2
        DESCRIPTION
            "This group is required only if 'dynamic assignment' of
             rows in the OverheadGroup tables is supported."
    ::= { dot20Compliances 1 }

END
```

Replace Clause 31 of IEEE Std 802.20-2008 with the following text:

31. 625k-MC OA & M Radio Network Quality Monitor and Control Enhancement

This clause is added to the baseline specification ATIS – 0700004.2005.

625k-MC network systems provide radio network quality monitoring and control functionality. The MIB of 625k-MC mode comprises of the managed objects, attributes, actions, and notifications required to manage a BS. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in 31.1.5.

31.1 625k-MC Mode MIB

31.1.1 Overview

This clause defines a Management Information Base (MIB) module for managing the 625k-MC mode. 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).The objects in this MIB are defined using the mechanisms specified in the Structure of Management Information (SMI). The MIB module specified is compliant to SMIv2, which is described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

31.1.2 625k-MC MIB structure

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

31.1.3 Security considerations

This MIB relates to a system that 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.

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

There are no MIB objects that could be used to turn off or change the security parameter configuration of an IEEE 802.20 base station. The presence or absence of security (encryption, authentication) is controlled for each individual user and cannot be 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 whom 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 IETF RFC 3410, 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 clause.

31.1.5 Definition[2]

```
IEEE802dot20-625k-MC-MIB  DEFINITIONS ::= BEGIN

IMPORTS
    ifIndex
             FROM IF-MIB
    MODULE-COMLIANCE, OBJECT-GROUP
             FROM SNMPv2-CONF
    Counter32, Counter64, Interger32, Unsigned32, MODULE-IDENTITY,
    OBJECT-IDENTITY, OBJECT-TYPE, transmission
       FROM SNMPv2-SMI
    TruthValue, Gauge, Counter, Gauge32       FROM SNMPv2-TC
   ;

IEEE802dot20-625k-MC-MIB MODULE-IDENTITY
   LAST-UPDATED        "201010047200OZ" - October 4, 2010
   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@kyocera-wireless.com"

    DESCRIPTION
        "The MIB module for IEEE802.20 625k-MC mode entities"
    ::= { iso (1) iso-identified-organization (3) ieee (111)
standardsassociation-numbered-series-standards (2) lan-man-stds (802)
ieee802dot20(20) ieee802dot20mibs(1) ieee802dot20625kMIB(2)}

dot20m625kmcSystem                        OBJECT IDENTIFIER    --
DESCRIPTION       "System Elements"
 ::= { IEEE802dot20-625k-MC-MIB 1 }


dot20m625kmcSysAlarms                     OBJECT IDENTIFIER    --
DESCRIPTION       "Alarms"
 ::= { dot20m625kmcSystem 1}



dot20m625kmcAlarmScalars                  OBJECT IDENTIFIER    --
DESCRIPTION       "Alarm Scalars"
 ::= { dot20m625kmcSysAlarms 1 }
```

[2] *Copyright release for MIB definitions:* Users of this standard may freely reproduce the MIB definitions in this annex so it can be used for its intended purpose.

```
dot20m625kmcCommonAlarmStatus               OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Common alarm status.



        (From mibCtl ElementType 16 CommonAlarmStatus)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcAlarmScalars 1 }



dot20m625kmcFailReasonForAlarm              OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Fail reason for alarm.



        (From mibCtl ElementType 15 FailReasonForAlarm)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcAlarmScalars 2 }




dot20m625kmcAlarmSummaryTable               OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcAlarmSummaryTableEntry
    MAX-ACCESS            not-accessible
    STATUS            Current
    DESCRIPTION       "Alarm Summary Table"
    ::= { dot20m625kmcSysAlarms 2 }



dot20m625kmcAlarmSummaryTableEntry          OBJECT-TYPE
    SYNTAX           dot20m625kmcAlarmSummaryTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcAlarmSummaryTableIndex }
    ::= { dot20m625kmcAlarmSummaryTable 1 }

dot20m625kmcAlarmSummaryTableEntry ::= SEQUENCE {
   dot20m625kmcAlarmSummaryTableIndex         INTEGER, -- AlarmEventType
   dot20m625kmcAlarmSummary                   INTEGER -- AlarmStateType
    }



dot20m625kmcAlarmSummaryTableIndex          OBJECT-TYPE
```

```
    SYNTAX            INTEGER -- AlarmEventType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 85 AlarmEventType :
            Enumeration of alarm event types.

            Defines semantics of events that are also alarms.
            All alarm events are enumerated first in the list of event types.
            The highest alarm event index will never be more than 255.
            [Limits: 0 255 ]
        Type derived from mibCtl Type 3 EventType :
            Enumeration of event types.

            Defines semantics of events.
            An event is re. an event log message.
            [Limits: 0 255 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcAlarmSummaryTableEntry 1 }



dot20m625kmcAlarmSummary                    OBJECT-TYPE
    SYNTAX            INTEGER -- AlarmStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Summary of all alarms generated by the base station.

        Each element contains the summary of a type of alarm (e.g. module
        over temperature).  When the management station sees that alarm
        summary is SET, it can, for example, query AlarmModuleOverTemp
        to see which module(s) is over temperature.

        (From mibCtl ElementType 5210 AlarmSummary)
        Description for mibCtl Type 80 AlarmStateType :
            Current state of an alarm.

            This value is CLEARED when
            the conditions which caused the alarm to occur are taken care of
            and no longer exist.
            The value is SET when due to some conditions, the Base Station
            software decides that an alarm is necessary.
            Typically (though this may not be true for all alarms
            or if the alarm changes state too frequently)
            an event is logged when an alarm is SET and then again when it
            is CLEARED.
            [Limits: 0 1 ]
        Description for mibCtl AlarmStateType 0 CLEARED :
            No alarm.
        Description for mibCtl AlarmStateType 1 SET :
            Alarm is set.
        "
    ::= { dot20m625kmcAlarmSummaryTableEntry 2 }



dot20m625kmcSysFiles                        OBJECT IDENTIFIER    --
DESCRIPTION     "Files"
    ::= { dot20m625kmcSystem 2}
```

```
dot20m625kmcStatsFiles                     OBJECT IDENTIFIER    --
DESCRIPTION       "Statistics file"
    ::= { dot20m625kmcSysFiles 1 }



dot20m625kmcStatsUploadURL                 OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..64)) -- URLType
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
        "EMS location to upload BS statistics file.



        (From mibCtl ElementType 2831 StatsUploadURL)
        Description for mibCtl Type 401 URLType :
            Universal Resource Locator (URL).

            A Universal Resource Locator (URL) is a text string
            that specifies a network location for a file.
            The general format for a URL consists of 2 parts:

            1. Protocol name: lower case letters, followed by a colon.
            See below for supported protocols.
            This field may be omitted, to default to the file: protocol.

            2. Additional information, depending on the protocol.
            For many protocols, a host name is required, which
            consists of a dotted numerical Internet Protocol (IP) address,
            or a dotted symbolic name with alphanumerical components,
            where supported.

            Supported protocols are:

            tftp: is the Trivial File Transfer Protocol.
            The additional information should begin with two slashes (//)
            followed by a host name, a slash (/) and a file path.
            The file path is interpreted by the host system,frequently
            relative to a special directory set up for this purpose.

            file: is the plain old file protocol.
            The additional information consists of a file path, which
            should begin with a slash (/).
            This is only useful if Base Station has been configured
            to be an Network File System (NFS) client of the host.
            The filepath is interpreted on the Base Station, so it
            must begin with the mount name specified in the NFS configuration.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcStatsFiles 1 }



dot20m625kmcStatsUploadStatus              OBJECT-TYPE
    SYNTAX            INTEGER -- FileUploadStatusType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Stats file upload status.
        (From mibCtl ElementType 2832 StatsUploadStatus)
        Description for mibCtl Type 403 FileUploadStatusType :
            File upload status.


        Description for mibCtl FileUploadStatusType 0 Unknown :
            File upload status is unknown.
        Description for mibCtl FileUploadStatusType 1 Missing :
            File is missing or invalid.
        Description for mibCtl FileUploadStatusType 2 PartialUpload :
            File is in the process of being upload to EMS.
        Description for mibCtl FileUploadStatusType 4 Complete :
            File is completely uploaded to EMS.
        Description for mibCtl FileUploadStatusType 5 Failure :
            Upload process is failure.
        Description for mibCtl FileUploadStatusType 6 NotManaged :
            File upload is not being managed.
        "
    ::= { dot20m625kmcStatsFiles 2 }



dot20m625kmcStatsUploadFailReason           OBJECT-TYPE
    SYNTAX            INTEGER -- FileUploadFailReasonType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Reason for last stats file upload failure.



        (From mibCtl ElementType 2833 StatsUploadFailReason)
        Description for mibCtl Type 406 FileUploadFailReasonType :
            Reason for failure to upload a file.


        Description for mibCtl FileUploadFailReasonType 0 NoFailure :
            File upload in progress or completed without problem.
        Description for mibCtl FileUploadFailReasonType 1 BadPathSpecified :
            File upload failed because network path not found.
        Description for mibCtl FileUploadFailReasonType 2 FlashDiskReadError :
            File upload failed because of flash disk read error.
        Description for mibCtl FileUploadFailReasonType 3 Aborted :
            File upload aborted due to change of specification.
        Description for mibCtl FileUploadFailReasonType 4 WriteError :
            Error in putting a file.
        "
    ::= { dot20m625kmcStatsFiles 3 }



dot20m625kmcStatsUploadBytes                OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Upload size of BS stats file in bytes.



        (From mibCtl ElementType 2834 StatsUploadBytes)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcStatsFiles 4 }



dot20m625kmcStatsUploadDate                 OBJECT-TYPE
    SYNTAX            Gauge32 -- AbsoluteTimeType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "BS Stats File upload complete time.



        (From mibCtl ElementType 2835 StatsUploadDate)
        Description for mibCtl Type 801 AbsoluteTimeType :
            Absolute time in GPS seconds.

            GPS (Global Positioning System) time in seconds since Jan. 6,
1980.
            Note that this differs from UTC (in addition to a possible
            offset due to starting time) due to leap seconds; see
            the GpsLeapSecond element.
        Type derived from mibCtl Type 18 Gauge32Type :
            32 bits of Gauge data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcStatsFiles 5 }



dot20m625kmcUploadStatsFile                 OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write -- REALLY: write-only
    STATUS            current
    DESCRIPTION
        "Upload Stats file.



        (From mibCtl ElementType 2836 UploadStatsFile)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcStatsFiles 6 }


dot20m625kmcSysInterfaces                   OBJECT IDENTIFIER    --
DESCRIPTION       "System Interfaces"
    ::= { dot20m625kmcSystem 3 }



dot20m625kmcInterfaceNetwork                OBJECT IDENTIFIER    --
DESCRIPTION       "Network Interfaces"
    ::= { dot20m625kmcSysInterfaces 1 }



dot20m625kmcTypeOfNetworkProtocol           OBJECT-TYPE
```

```
    SYNTAX            INTEGER -- NetworkProtocolType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Type of Network Protocol used with the Network.

        Type of Network Protocol is Ethernet or ATM.

        (From mibCtl ElementType 3002 TypeOfNetworkProtocol)
        Description for mibCtl Type 214 NetworkProtocolType :
            Network Protocol type.
        Description for mibCtl NetworkProtocolType 0 Unknown :
            Network protocol type is unknown.
        Description for mibCtl NetworkProtocolType 1 Ethernet :
            Ethernet interface.
        Description for mibCtl NetworkProtocolType 2 ATM :
            ATM interface.
        "
    ::= { dot20m625kmcInterfaceNetwork 1 }



dot20m625kmcMgmtNetConfigTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcMgmtNetConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Mgmt Network Configuration"
    ::= { dot20m625kmcInterfaceNetwork 2 }



dot20m625kmcMgmtNetConfigTableEntry         OBJECT-TYPE
    SYNTAX           dot20m625kmcMgmtNetConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcMgmtNetConfigTableIndex }
    ::= { dot20m625kmcMgmtNetConfigTable 1 }

dot20m625kmcMgmtNetConfigTableEntry ::= SEQUENCE {
   dot20m625kmcMgmtNetConfigTableIndex        INTEGER, -- MoNerdAddressType
   dot20m625kmcEthernetIPAddress              OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmcEthernetIPLocalBits            OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmcEthernetHostName               OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcMgmtNetConfigTableIndex         OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcMgmtNetConfigTableEntry 1 }



dot20m625kmcEthernetIPAddress               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Internet Protocol (IP) address for ethernet port of Module.

        This is the actual IP address in use for the ethernet port
        of a given Module.
        If IP is not being used on the ethernet port, or there is
        no ethernet port, then an empty string is provided for this element.

        (From mibCtl ElementType 2811 EthernetIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcMgmtNetConfigTableEntry 2 }



dot20m625kmcEthernetIPLocalBits             OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Ethernet IP (Internet Protocol) local routing bit count.

        This indicates how many of the low-order bits of
        the IP address of the ethernet connection are used
        within the local network.
        The remaining (high-order) bits are the same for all
        hosts on the local network.
        This is used as the first part of the routing algorithm.
        IP addresses that do not share the upper bits of the ethernet
        IP address and which are not otherwise resolved will be sent
        through the gateway, if defined.

        For example, 255.255.255.0

        (From mibCtl ElementType 2812 EthernetIPLocalBits)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcMgmtNetConfigTableEntry 3 }



dot20m625kmcEthernetHostName                OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Ethernet IP host name for module.

        (From mibCtl ElementType 2813 EthernetHostName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcMgmtNetConfigTableEntry 4 }




dot20m625kmcUserNetConfigTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcUserNetConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "User Network Configuration"
    ::= { dot20m625kmcInterfaceNetwork 3 }



dot20m625kmcUserNetConfigTableEntry         OBJECT-TYPE
    SYNTAX           dot20m625kmcUserNetConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcUserNetConfigTableIndex }
    ::= { dot20m625kmcUserNetConfigTable 1 }

dot20m625kmcUserNetConfigTableEntry ::= SEQUENCE {
   dot20m625kmcUserNetConfigTableIndex        INTEGER, -- MoNerdAddressType
   dot20m625kmcUserEthernetIPAddress          OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmcUserEthernetIPLocalBits        OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmcUserEthernetHostName           OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcUserNetConfigTableIndex         OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcUserNetConfigTableEntry 1 }



dot20m625kmcUserEthernetIPAddress           OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Internet Protocol (IP) address for user ethernet port of Module.

        This is the actual IP address in use for the ethernet port
        of a given Module.
        If IP is not being used on the ethernet port, or there is
        no ethernet port, then an empty string is provided for this element.

        (From mibCtl ElementType 2817 UserEthernetIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcUserNetConfigTableEntry 2 }



dot20m625kmcUserEthernetIPLocalBits         OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Ethernet IP subnet mask for user network.

        Ethernet IP subnet mask for user network

        (From mibCtl ElementType 2818 UserEthernetIPLocalBits)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcUserNetConfigTableEntry 3 }



dot20m625kmcUserEthernetHostName            OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "User ethernet IP host name for module.

        (From mibCtl ElementType 2819 UserEthernetHostName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcUserNetConfigTableEntry 4 }



dot20m625kmcUserNetStatusTable              OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcUserNetStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Network Status"
    ::= { _625K-MCInterfaceNetwork 4 }



dot20m625kmcUserNetStatusTableEntry         OBJECT-TYPE
    SYNTAX           dot20m625kmcUserNetStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcUserNetStatusTableIndex }
    ::= { dot20m625kmcUserNetStatusTable 1 }

dot20m625kmcUserNetStatusTableEntry ::= SEQUENCE {
   dot20m625kmcUserNetStatusTableIndex        INTEGER, -- MoNerdAddressType
   dot20m625kmcNetworkInOctets                Counter, -- Counter32Type
   dot20m625kmcNetworkOutOctets               Counter -- Counter32Type
    }



dot20m625kmcUserNetStatusTableIndex         OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcUserNetStatusTableEntry 1 }



dot20m625kmcNetworkInOctets                 OBJECT-TYPE
    SYNTAX            Counter32 -- Counter32Type
    MAX-ACCESS            read-only
    STATUS            Currenturrent
    DESCRIPTION
        "In octets user data of network.



        (From mibCtl ElementType 1000 NetworkInOctets)
        Description for mibCtl Type 19 Counter32Type :
            32 bits of Counter data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcUserNetStatusTableEntry 2 }



dot20m625kmcNetworkOutOctets                OBJECT-TYPE
    SYNTAX            Counter32 -- Counter32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Out octets user data of network.



        (From mibCtl ElementType 1001 NetworkOutOctets)
        Description for mibCtl Type 19 Counter32Type :
            32 bits of Counter data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcUserNetStatusTableEntry 3 }




dot20m625kmcL2TPConfigTable                 OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcL2TPConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "L2TP Configuration Table"
    ::= { dot20m625kmcInterfaceNetwork 5 }



dot20m625kmcL2TPConfigTableEntry            OBJECT-TYPE
    SYNTAX           dot20m625kmcL2TPConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcL2TPConfigTableIndex }
    ::= { dot20m625kmcL2TPConfigTable 1 }

dot20m625kmcL2TPConfigTableEntry ::= SEQUENCE {
   dot20m625kmcL2TPConfigTableIndex           INTEGER, -- MoNerdAddressType
   dot20m625kmcL2TPPeerName                   OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcL2TPPeerIPAddress              OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmcL2TPAVPHostName                OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcL2TPAVPChallAndRes             OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcL2TPConfigTableIndex            OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcL2TPConfigTableEntry 1 }



dot20m625kmcL2TPPeerName                    OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "L2TP peer name.

        Tunnel switch host name

        (From mibCtl ElementType 2000 L2TPPeerName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcL2TPConfigTableEntry 2 }



dot20m625kmcL2TPPeerIPAddress               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "L2TP peer IP Address.

        Tunnel switch IP Address

        (From mibCtl ElementType 2001 L2TPPeerIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcL2TPConfigTableEntry 3 }


dot20m625kmcL2TPAVPHostName                 OBJECT-TYPE
```

```
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "BS host name using L2TP.



        (From mibCtl ElementType 2008 L2TPAVPHostName)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcL2TPConfigTableEntry 4 }



dot20m625kmcL2TPAVPChallAndRes              OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "AVP challenge and response name.



        (From mibCtl ElementType 2012 L2TPAVPChallAndRes)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcL2TPConfigTableEntry 5 }




dot20m625kmcL2TPStatusTable                 OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcL2TPStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "L2TP Status Table"
    ::= { dot20m625kmcInterfaceNetwork 6 }



dot20m625kmcL2TPStatusTableEntry            OBJECT-TYPE
    SYNTAX           dot20m625kmcL2TPStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcL2TPStatusTableIndex }
    ::= { dot20m625kmcL2TPStatusTable 1 }

dot20m625kmcL2TPStatusTableEntry ::= SEQUENCE {
   dot20m625kmcL2TPStatusTableIndex           INTEGER, -- MoNerdAddressType
   dot20m625kmcL2TPActiveSession              Unsigned32, -- Unsigned32Type
   dot20m625kmcL2TPActiveTunnel               Unsigned32 -- Unsigned32Type
    }



dot20m625kmcL2TPStatusTableIndex            OBJECT-TYPE
```

```
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcL2TPStatusTableEntry 1 }



dot20m625kmcL2TPActiveSession               OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "L2TP active session.



        (From mibCtl ElementType 2013 L2TPActiveSession)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcL2TPStatusTableEntry 2 }



dot20m625kmcL2TPActiveTunnel                OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "L2TP active tunnel.



        (From mibCtl ElementType 2014 L2TPActiveTunnel)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcL2TPStatusTableEntry 3 }




dot20m625kmcATMConfigTable                  OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcATMConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "ATM Configuration Table"
    ::= { dot20m625kmcInterfaceNetwork 7 }



dot20m625kmcATMConfigTableEntry             OBJECT-TYPE
    SYNTAX           dot20m625kmcATMConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcATMConfigTableIndex }
    ::= { dot20m625kmcATMConfigTable 1 }

dot20m625kmcATMConfigTableEntry ::= SEQUENCE {
   dot20m625kmcATMConfigTableIndex            INTEGER, -- MoNerdAddressType
   dot20m625kmcAtmAddress                     OCTET STRING (SIZE(0..40)),
-- TextType X 40
   dot20m625kmcAtmVCTypes                     INTEGER, -- AtmVCType
   dot20m625kmcAtmFrameTypes                  INTEGER, -- AtmFrameType
   dot20m625kmcAtmUNIVersion                  INTEGER, -- AtmUNIVersionType
   dot20m625kmcAtmLineStatus                  INTEGER, -- LineStatusType
   dot20m625kmcAtmParameterFailReason         Unsigned32, -- Unsigned32Type
   dot20m625kmcMCAtmOpenChannelFailReason       Unsigned32, -- Unsigned32Type
   dot20m625kmcAtmChannelNumber               Unsigned32, -- Unsigned32Type
   dot20m625kmcAtmAlarmCauseRegister          Unsigned32, -- Unsigned32Type
   dot20m625kmcAtmPHYIntrCauseRegister        Unsigned32 -- Unsigned32Type
    }



dot20m625kmcATMConfigTableIndex             OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcATMConfigTableEntry 1 }



dot20m625kmcAtmAddress                      OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..40)) -- TextType X 40
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Atm Address.



        (From mibCtl ElementType 1950 AtmAddress)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 2 }



dot20m625kmcAtmVCTypes                      OBJECT-TYPE
    SYNTAX            INTEGER -- AtmVCType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Atm VC Type.



        (From mibCtl ElementType 1951 AtmVCTypes)
        Description for mibCtl Type 300 AtmVCType :
            ATM VC Type.
        Description for mibCtl AtmVCType 0 Unknown :
            ATM VC Type is Unknown.
        Description for mibCtl AtmVCType 1 PVC :
            ATM VC Type is PVC.
        Description for mibCtl AtmVCType 2 SVC :
            ATM VC Type is SVC.
        Description for mibCtl AtmVCType 3 PVCSVC :
            ATM VC Type is PVC & SVC.
        "
    ::= { dot20m625kmcATMConfigTableEntry 3 }



dot20m625kmcAtmFrameTypes                   OBJECT-TYPE
    SYNTAX            INTEGER -- AtmFrameType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Atm Frame Type.



        (From mibCtl ElementType 1952 AtmFrameTypes)
        Description for mibCtl Type 301 AtmFrameType :
            ATM Frame Type.
        Description for mibCtl AtmFrameType 0 Unknown :
            ATM Frame Type is Unknown.
        Description for mibCtl AtmFrameType 1 OC48 :
            ATM Frame Type is OC48.
        Description for mibCtl AtmFrameType 2 OC36 :
            ATM Frame Type is OC36.
        Description for mibCtl AtmFrameType 3 OC24 :
            ATM Frame Type is OC24.
        Description for mibCtl AtmFrameType 4 OC18 :
            ATM Frame Type is OC18.
        Description for mibCtl AtmFrameType 5 OC12 :
            ATM Frame Type is OC12.
        Description for mibCtl AtmFrameType 6 OC9 :
            ATM Frame Type is OC9.
        Description for mibCtl AtmFrameType 7 OC3 :
            ATM Frame Type is OC3.
        Description for mibCtl AtmFrameType 8 OC1 :
            ATM Frame Type is OC1.
        Description for mibCtl AtmFrameType 9 STM16 :
            ATM Frame Type is STM16.
        Description for mibCtl AtmFrameType 10 STM4 :
            ATM Frame Type is STM4.
        Description for mibCtl AtmFrameType 11 STM1 :
            ATM Frame Type is STM1.
        Description for mibCtl AtmFrameType 12 DS3 :
            ATM Frame Type is DS3.
        Description for mibCtl AtmFrameType 13 DS2 :
            ATM Frame Type is DS2.
        Description for mibCtl AtmFrameType 14 DS1 :
            ATM Frame Type is DS1.
        Description for mibCtl AtmFrameType 15 DS0 :
            ATM Frame Type is DS0.
        Description for mibCtl AtmFrameType 16 E3 :
            ATM Frame Type is E3.
        Description for mibCtl AtmFrameType 17 E2 :
            ATM Frame Type is E2.
        Description for mibCtl AtmFrameType 18 E1 :
            ATM Frame Type is E1.
        Description for mibCtl AtmFrameType 19 E0 :
            ATM Frame Type is E0.
        "
    ::= { dot20m625kmcATMConfigTableEntry 4 }



dot20m625kmcAtmUNIVersion                  OBJECT-TYPE
    SYNTAX            INTEGER -- AtmUNIVersionType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Atm UNI version.



        (From mibCtl ElementType 1953 AtmUNIVersion)
        Description for mibCtl Type 302 AtmUNIVersionType :
            ATM UNI Version Type.
        Description for mibCtl AtmUNIVersionType 0 Unknown :
            ATM UNI Version Type is Unknown.
        Description for mibCtl AtmUNIVersionType 1 V30 :
            ATM UNI Version Type is 3.0.
        Description for mibCtl AtmUNIVersionType 2 V31 :
            ATM UNI Version Type is 3.1.
        Description for mibCtl AtmUNIVersionType 3 V40 :
            ATM UNI Version Type is 4.0.
        "
    ::= { dot20m625kmcATMConfigTableEntry 5 }



dot20m625kmcAtmLineStatus                  OBJECT-TYPE
    SYNTAX            INTEGER -- LineStatusType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm line status.

        ATM line status

        (From mibCtl ElementType 1956 AtmLineStatus)
        Description for mibCtl Type 72 LineStatusType :
            Line status type.
        Description for mibCtl LineStatusType 0 LinkUp :
            Line status is link up.
        Description for mibCtl LineStatusType 1 LinkDown :
            Line status is link down.
        "
    ::= { dot20m625kmcATMConfigTableEntry 6 }
```

```
dot20m625kmcAtmParameterFailReason          OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm parameter fail reason.

        ATM parameter fail reason

        (From mibCtl ElementType 1957 AtmParameterFailReason)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 7 }



dot20m625kmcAtmOpenChannelFailReason        OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm open channel fail reason.

        ATM open channel fail reason

        (From mibCtl ElementType 1958 AtmOpenChannelFailReason)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 8 }



dot20m625kmcAtmChannelNumber                OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm open channel fail reason.

        ATM open channel fail reason

        (From mibCtl ElementType 1959 AtmChannelNumber)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 9 }



dot20m625kmcAtmAlarmCauseRegister           OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm alarm cause register.

        ATM alarm cause register

        (From mibCtl ElementType 1960 AtmAlarmCauseRegister)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 10 }



dot20m625kmcAtmPHYIntrCauseRegister         OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Atm PHY Interrupt cause register.

        ATM PHY interrupt cause register

        (From mibCtl ElementType 1961 AtmPHYIntrCauseRegister)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcATMConfigTableEntry 11 }




dot20m625kmcA10ConfigTable                  OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcA10ConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "A10 Configuration Table"
    ::= { dot20m625kmcInterfaceNetwork 9 }



dot20m625kmcA10ConfigTableEntry             OBJECT-TYPE
    SYNTAX           dot20m625kmcA10ConfigTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcA10ConfigTableIndex }
    ::= { dot20m625kmcA10ConfigTable 1 }

dot20m625kmcA10ConfigTableEntry ::= SEQUENCE {
   dot20m625kmcA10ConfigTableIndex            INTEGER, -- MoNerdAddressType
   dot20m625kmc3GPP2PriPDSNIPAddress          OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmc3GPP2PriPDSNSharedSecret       OCTET STRING (SIZE(0..64)),
-- TextType X 64
   dot20m625kmc3GPP2PriPDSNSPI                Unsigned32, -- Unsigned32Type
   dot20m625kmc3GPP2SecPDSNIPAddress          OCTET STRING (SIZE(0..15)),
-- IPAddressTextType
   dot20m625kmc3GPP2SecPDSNSharedSecret       OCTET STRING (SIZE(0..64)),
-- TextType X 64
   dot20m625kmc3GPP2SecPDSNSPI                Unsigned32 -- Unsigned32Type
```

```
    }



dot20m625kmcA10ConfigTableIndex             OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcA10ConfigTableEntry 1 }



dot20m625kmc3GPP2PriPDSNIPAddress           OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 primary PDSN IP address.

        3GPP2 Primary PDSN IP Address

        (From mibCtl ElementType 2100 3GPP2PriPDSNIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 2 }



dot20m625kmc3GPP2PriPDSNSharedSecret        OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..64)) -- TextType X 64
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 primary PDSN shared secret.

        3GPP2 primary PDSN shared secret

        (From mibCtl ElementType 2101 3GPP2PriPDSNSharedSecret)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 3 }



dot20m625kmc3GPP2PriPDSNSPI                 OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 primary PDSN SPI.

        3GPP2 primary PDSN SPI

        (From mibCtl ElementType 2102 3GPP2PriPDSNSPI)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 4 }



dot20m625kmc3GPP2SecPDSNIPAddress           OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 secondary PDSN IP address.

        3GPP2 secondary PDSN IP address

        (From mibCtl ElementType 2103 3GPP2SecPDSNIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 5 }



dot20m625kmc3GPP2SecPDSNSharedSecret        OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..64)) -- TextType X 64
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 secondary PDSN shared secret.

        3GPP2 secondary PDSN shared secret

        (From mibCtl ElementType 2104 3GPP2SecPDSNSharedSecret)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 6 }



dot20m625kmc3GPP2SecPDSNSPI                 OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "3GPP2 secondary PDSN SPI.

        3GPP2 secondary PDSN SPI

        (From mibCtl ElementType 2105 3GPP2SecPDSNSPI)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcA10ConfigTableEntry 7 }




dot20m625kmcA10StatusTable                  OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcA10StatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "A10 Status Table"
    ::= { dot20m625kmcInterfaceNetwork 10 }



dot20m625kmcA10StatusTableEntry             OBJECT-TYPE
    SYNTAX           dot20m625kmcA10StatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcA10StatusTableIndex }
    ::= { dot20m625kmcA10StatusTable 1 }

dot20m625kmcA10StatusTableEntry ::= SEQUENCE {
   dot20m625kmcA10StatusTableIndex            INTEGER, -- MoNerdAddressType
   dot20m625kmc3GPP2PDSNIPAddress             OCTET STRING (SIZE(0..15))
-- IPAddressTextType
    }



dot20m625kmcA10StatusTableIndex             OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcA10StatusTableEntry 1 }



dot20m625kmc3GPP2PDSNIPAddress              OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..15)) -- IPAddressTextType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "3GPP2 PDSN IP Address.

        Current main using PDSN IP address

        (From mibCtl ElementType 2113 3GPP2PDSNIPAddress)
        Description for mibCtl Type 420 IPAddressTextType :
            Internet Protocol Address (Text).

            This text must currently be in the dotted abc.def.ghi.jkl format.
            In the future, hostnames might be allowed.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcA10StatusTableEntry 2 }



dot20m625kmcInterfaceRF                     OBJECT IDENTIFIER    --
DESCRIPTION       "Radio Frequency Interfaces"
    ::= { dot20m625kmcSysInterfaces 2 }



dot20m625kmcCarrierTable                    OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcCarrierTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Carrier Frequency Table"
    ::= { dot20m625kmcInterfaceRF 1 }



dot20m625kmcCarrierTableEntry               OBJECT-TYPE
    SYNTAX           dot20m625kmcCarrierTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcCarrierTableIndex }
    ::= { dot20m625kmcCarrierTable 1 }

dot20m625kmcCarrierTableEntry ::= SEQUENCE {
   dot20m625kmcCarrierTableIndex             INTEGER,-- BaseStationCarrierType
   dot20m625kmcCarrierUsage                  INTEGER -- CarrierUsageType
    }



dot20m625kmcCarrierTableIndex               OBJECT-TYPE
    SYNTAX            INTEGER -- BaseStationCarrierType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
```

```
        Description for mibCtl Type 219 BaseStationCarrierType :
            Base station carrier number.

            Base station carriers are a contiguous set of carriers
            that are used by the Base Station;
            they are numbered from 0 to a current maximum of 32-1.
            [Limits: 0 15 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcCarrierTableEntry 1 }



dot20m625kmcCarrierUsage                    OBJECT-TYPE
    SYNTAX            INTEGER -- CarrierUsageType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Current assigned usage per base station carrier.



        (From mibCtl ElementType 54 CarrierUsage)
        Description for mibCtl Type 220 CarrierUsageType :
            The assigned use of a radio carrier.

            A radio carrier is a frequency band.
            The assigned use of a carrier can be Reserved, Control
            or Traffic.
            Base station transmits control information on one of the time
slots
            of given Control carrier.
            Base station does not transmit anything on Reserved carriers.
        Description for mibCtl CarrierUsageType 0 NotUse :
            Not Use for this carrier.
        Description for mibCtl CarrierUsageType 1 TCH :
            All timeslots in this carrier are for traffic only.
        Description for mibCtl CarrierUsageType 2 TCHBCH :
            One timeslot in this carrier is for BCH, others for TCH.
        "
    ::= { dot20m625kmcCarrierTableEntry 2 }



dot20m625kmcBSCC                            OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Base Station Color Code.

        (From mibCtl ElementType 61 BSCC)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 2 }
```

```
dot20m625kmcBSLowestCarrier                 OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "The lowest carrier of the base station operating band.

        This is an extended carrier number
        that identifies the lowest carrier of the bandwidth
        to which the base station is tuned.
        This value cannot be changed while the Base Station state
        is Operating.

        (From mibCtl ElementType 52 BSLowestCarrier)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 3 }



dot20m625kmcBCHModuleAddress                OBJECT-TYPE
    SYNTAX            INTEGER -- ModuleAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Which module is handling the broadcast channel.

        (From mibCtl ElementType 57 BCHModuleAddress)
        Description for mibCtl Type 202 ModuleAddressType :
            Base station bus slot address.

            Most components of the Base Station for which data can
            be obtained are identified by a ModuleAddressType address
            and possibly a subsidiary address.
            [Limits: 0 7 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 4 }



dot20m625kmcBCHCarrierNumber                OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Number Of BCH Carrier.

        (From mibCtl ElementType 58 BCHCarrierNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 5 }
```

```
dot20m625kmcRACHCarrierMask                 OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "RACH carrier mask.



        (From mibCtl ElementType 73 RACHCarrierMask)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 6 }



dot20m625kmcRACHSlotMask                    OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "RACH slot mask.



        (From mibCtl ElementType 72 RACHSlotMask)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 7 }



dot20m625kmcCalibrationInterval             OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Calibration interval time.



        (From mibCtl ElementType 75 CalibrationInterval)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 8 }



dot20m625kmcSpatialParameter                OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
```

```
        "Spatial parameter.



        (From mibCtl ElementType 78 SpatialParameter)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 9 }



dot20m625kmcCostCalcParameter               OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Cost calculation parameter.



        (From mibCtl ElementType 79 CostCalcParameter)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 10 }



dot20m625kmcBSRegistrationCapacity          OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Capacity of UT registration on BS.



        (From mibCtl ElementType 76 BSRegistrationCapacity)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 11 }



dot20m625kmcBSRegistrationTimer             OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Timer of keeping UT registration on BS.



        (From mibCtl ElementType 77 BSRegistrationTimer)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcInterfaceRF 12 }



dot20m625kmcPCHFrequencyHopping             OBJECT-TYPE
    SYNTAX            INTEGER -- BooleanType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Propriety of frequency hopping (PCH).



        (From mibCtl ElementType 70 PCHFrequencyHopping)
        Description for mibCtl Type 16 BooleanType :
            Truth value, 0=FALSE, 1=TRUE.

            This is a subset of TriStateType; no UNDEFINED value is provided.
            [Limits: 0 1 ]
        Description for mibCtl BooleanType 0 FALSE :
            False.
        Description for mibCtl BooleanType 1 TRUE :
            True.
        "
    ::= { dot20m625kmcInterfaceRF 13 }



dot20m625kmcTCHFrequencyHopping             OBJECT-TYPE
    SYNTAX            INTEGER -- BooleanType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Propriety of frequency hopping (TCH).



        (From mibCtl ElementType 71 TCHFrequencyHopping)
        Description for mibCtl Type 16 BooleanType :
            Truth value, 0=FALSE, 1=TRUE.

            This is a subset of TriStateType; no UNDEFINED value is provided.
            [Limits: 0 1 ]
        Description for mibCtl BooleanType 0 FALSE :
            False.
        Description for mibCtl BooleanType 1 TRUE :
            True.
        "
    ::= { dot20m625kmcInterfaceRF 14 }




dot20m625kmcRFStatusTable                   OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcRFStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "RF Status Table"
    ::= { dot20m625kmcInterfaceRF 15 }
```

```
dot20m625kmcRFStatusTableEntry              OBJECT-TYPE
    SYNTAX           dot20m625kmcRFStatusTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcRFStatusTableIndex }
    ::= { dot20m625kmcRFStatusTable 1 }

dot20m625kmcRFStatusTableEntry ::= SEQUENCE {
   dot20m625kmcRFStatusTableIndex              INTEGER, -- MoNerdAddressType
   dot20m625kmcBSAirBitRateUpLink              Unsigned32, -- Unsigned32Type
   dot20m625kmcBSAirBitRateDownLink            Unsigned32, -- Unsigned32Type
   dot20m625kmcBSActiveStream                  Unsigned32, -- Unsigned32Type
   dot20m625kmcBSActiveRegistration            Unsigned32 -- Unsigned32Type
    }



dot20m625kmcRFStatusTableIndex              OBJECT-TYPE
    SYNTAX            INTEGER -- MoNerdAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 204 MoNerdAddressType :
            Base station network component address.

            A network address is a subset of Base Station component addresses,
            restricted to network components only.
            Network components interface with a telephony switch or similar.
            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcRFStatusTableEntry 1 }



dot20m625kmcBSAirBitRateUpLink              OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Radio bit rate of up link per Modem control board.



        (From mibCtl ElementType 4022 BSAirBitRateUpLink)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcRFStatusTableEntry 2 }



dot20m625kmcBSAirBitRateDownLink            OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
```

```
        "Radio bit rate of down link per Modem control board.



        (From mibCtl ElementType 4023 BSAirBitRateDownLink)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcRFStatusTableEntry 3 }



dot20m625kmcBSActiveStream                  OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Number of streams currently connected in a base station.

        Number of active streams.

        (From mibCtl ElementType 4020 BSActiveStream)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcRFStatusTableEntry 4 }



dot20m625kmcBSActiveRegistration            OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Number of registrations currently existed in a base station.

        The call capacity is determined by the available resources in a
        base station.

        (From mibCtl ElementType 4021 BSActiveRegistration)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcRFStatusTableEntry 5 }

dot20m625kmcSysScalars                      OBJECT IDENTIFIER    --
DESCRIPTION       "System Scalars"
    ::= { dot20m625kmcSystem 4 }



dot20m625kmcBaseStationID                   OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..18)) -- TextType X 18
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Base Station Identification Code.
        This text string must represent in hexadecimal a 42 bit number
        to be used as the Base Station Identification Code (BSID).
        The BSID is used by the base station to identify itself to
        subscriber units.
        The BSID of a base station must at a minimum
        differ from that of any other base station
        where both would be within radio reception distance
        of any subscriber unit.

        This cannot be changed while the Base Station state is Operating.

        (From mibCtl ElementType 60 BaseStationID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 1 }



dot20m625kmcBaseStationTypeID               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Type ID of base station.



        (From mibCtl ElementType 66 BaseStationTypeID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 2 }



dot20m625kmcBaseStationGroupID              OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Group ID of base station.



        (From mibCtl ElementType 67 BaseStationGroupID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 3 }



dot20m625kmcBaseStationSubGroupID           OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Sub group ID of base station.



        (From mibCtl ElementType 68 BaseStationSubGroupID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 4 }



dot20m625kmcDesiredStateOfBaseStation       OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Desired state of base station as a whole.

        This indicates the Base Station state desired by the operator.
        These desired states are currently supported:

        Operating -  for normal operation.

        Ready -  to avoid taking any new calls.
        Existing calls will not be terminated except normally or by
        command from the operator.
        While existing calls remain, the base station state will
        remain as Operating.

        This information is permanently stored on the base station.

        (From mibCtl ElementType 42 DesiredStateOfBaseStation)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.
            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.

            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up..

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading..

            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction..

            Restriction state.
        "
    ::= { dot20m625kmcSysScalars 5 }



dot20m625kmcTypeOfReboot                    OBJECT-TYPE
    SYNTAX            INTEGER -- RebootType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Type of reboot for base station.



        (From mibCtl ElementType 63 TypeOfReboot)
        Description for mibCtl Type 250 RebootType :
            Reboot Type.
        Description for mibCtl RebootType 0 Force :
            Force  mode.
        Description for mibCtl RebootType 1 Graceful :
            Graceful  mode.
        "
    ::= { dot20m625kmcSysScalars 6 }



dot20m625kmcBaseStationRebootTime           OBJECT-TYPE
    SYNTAX            Gauge -- AbsoluteTimeType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Time of base station reboot.

        This is the base station reboot time (GPS time).

        (From mibCtl ElementType 62 BaseStationRebootTime)
        Description for mibCtl Type 801 AbsoluteTimeType :
            Absolute time in GPS seconds.

            GPS (Global Positioning System) time in seconds since Jan. 6,
1980.
            Note that this differs from UTC (in addition to a possible
            offset due to starting time) due to leap seconds; see
            the GpsLeapSecond element.
        Type derived from mibCtl Type 18 Gauge32Type :
            32 bits of Gauge data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcSysScalars 7 }



dot20m625kmcTypeOfBSDiagnosis               OBJECT-TYPE
    SYNTAX            INTEGER -- DiagnosisType
    MAX-ACCESS            read-write
    STATUS            current
    DESCRIPTION
        "Type of diagnosis for base station.



        (From mibCtl ElementType 64 TypeOfBSDiagnosis)
        Description for mibCtl Type 251 DiagnosisType :
            Diagnosis Type.
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcSysScalars 8 }



dot20m625kmcBSDiagnosisStatus               OBJECT-TYPE
    SYNTAX            INTEGER -- DiagnosisStatusType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Diagnosis status for base station.



        (From mibCtl ElementType 370 BSDiagnosisStatus)
        Description for mibCtl Type 252 DiagnosisStatusType :
            Diagnosis status Type.
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcSysScalars 9 }
```

```
dot20m625kmcBSDiagnosisFailReason           OBJECT-TYPE
    SYNTAX            INTEGER -- DiagFailReasonType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Diagnosis fail reason for base station.



        (From mibCtl ElementType 371 BSDiagnosisFailReason)
        Description for mibCtl Type 253 DiagFailReasonType :
            Diagnosis fail reason Type.
        Description for mibCtl DiagFailReasonType 1 PDSNPing :
            Diagnosis fail reason is PDSN Ping.
        Description for mibCtl DiagFailReasonType 2 Calibration :
            Diagnosis fail reason is Calibration.
        Description for mibCtl DiagFailReasonType 3 AntPath :
            Diagnosis fail reason is TRx Antenna Path.
        Description for mibCtl DiagFailReasonType 4 LOAlive :
            Diagnosis fail reason is Local Oscilator DSP Alive.
        Description for mibCtl DiagFailReasonType 5 GCLoopBack :
            Diagnosis fail reason is GCLoopBack.
        Description for mibCtl DiagFailReasonType 6 SlaveNM :
            Diagnosis fail reason is Slave Modem control board.
        Description for mibCtl DiagFailReasonType 7 GPSAnt :
            Diagnosis fail reason is GPS Antenna.
        Description for mibCtl DiagFailReasonType 8 SlotDSP :
            Diagnosis fail reason is Modem control board DSP Alive.
        Description for mibCtl DiagFailReasonType 9 ATMAlive :
            Diagnosis fail reason is ATM Alive.
        Description for mibCtl DiagFailReasonType 96 UndefineName :
            Diagnosis fail reason is Undefine Diag Name.
        Description for mibCtl DiagFailReasonType 97 TimeOut :
            Diagnosis fail reason is Time Out.
        Description for mibCtl DiagFailReasonType 98 InvalidStateExec :
            Diagnosis fail reason is Invalid State Execute.
        Description for mibCtl DiagFailReasonType 99 ExecFail :
            Diagnosis fail reason is Execute Fail.
        "
    ::= { dot20m625kmcSysScalars 10 }



dot20m625kmcDiskDbUpdateSequence            OBJECT-TYPE
    SYNTAX            Gauge -- Gauge32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base station Flach update sequence number.

        This number is incremented on disk every time any other
        database element is actually changed on flash.
        It is not incremented on redundant sets.
        This number may also be set to a desired value.

        (From mibCtl ElementType 6 DiskDbUpdateSequence)
        Description for mibCtl Type 18 Gauge32Type :
            32 bits of Gauge data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcSysScalars 11 }
```

```
dot20m625kmcStateOfBaseStation              OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "State of base station as a whole.

        This will not have values of Unknown or PowerOff since the
        base station would be unable to report such values.

        When sufficiently initialized, the state will be Operating if
        accepting new calls (according to the desired state of the
        base station) or continuing ongoing calls;
        or Ready if the desired state is Ready and their are no
        ongoing calls.

        (From mibCtl ElementType 41 StateOfBaseStation)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.

            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.

            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up.

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading.

            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction.

            Restriction state.
        "
    ::= { dot20m625kmcSysScalars 12 }



dot20m625kmcBSTotalIndication               OBJECT-TYPE
    SYNTAX            INTEGER -- IndicationType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Status of BS total indicator.

        In the current implementation,
        this indication is set to the value On by BS.

        (From mibCtl ElementType 525 BSTotalIndication)
        Description for mibCtl Type 230 IndicationType :
            Hardware indication status (LEDs).


        Description for mibCtl IndicationType 0 Off :
            Off.
        Description for mibCtl IndicationType 1 Amber :
            Amber.
        Description for mibCtl IndicationType 2 Red :
            Red.
        Description for mibCtl IndicationType 3 Green :
            Green.
        Description for mibCtl IndicationType 4 NotPresent :
            Not present.
        "
    ::= { dot20m625kmcSysScalars 13 }



dot20m625kmcMasterAddress                   OBJECT-TYPE
    SYNTAX            INTEGER -- ModuleAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Bus slot address of master Modem control board.

        This indicates which Modem control board is master
        of the base station.

        (From mibCtl ElementType 32 MasterAddress)
        Description for mibCtl Type 202 ModuleAddressType :
            Base station bus slot address.
            Most components of the Base Station for which data can
            be obtained are identified by a ModuleAddressType address
            and possibly a subsidiary address.
            [Limits: 0 7 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcSysScalars 14 }



dot20m625kmcBSManufactureID                 OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base Station manufacture identification number.

        The manufacture identification assigned by Vendor

        (From mibCtl ElementType 201 BSManufactureID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 15 }



dot20m625kmcBSSerialNumber                  OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base Station serial number.

        This is the character serial number of the base station.
        This serial number will be unique among all base stations
        of this type regardless of manufacturer.

        (From mibCtl ElementType 203 BSSerialNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 16 }



dot20m625kmcDiagnosisBaseStation            OBJECT-TYPE
    SYNTAX            INTEGER -- BooleanType
    MAX-ACCESS            read-write -- REALLY: write-only
    STATUS            current
    DESCRIPTION
        "Diagnosis base station.

        This is a write-only element; only a value of TRUE is valid.

        (From mibCtl ElementType 47 DiagnosisBaseStation)
        Description for mibCtl Type 16 BooleanType :
            Truth value, 0=FALSE, 1=TRUE.

            This is a subset of TriStateType; no UNDEFINED value is provided.
            [Limits: 0 1 ]
        Description for mibCtl BooleanType 0 FALSE :
            False.
        Description for mibCtl BooleanType 1 TRUE :
            True.
        "
    ::= { dot20m625kmcSysScalars 17 }



dot20m625kmcRebootBaseStation               OBJECT-TYPE
    SYNTAX            INTEGER -- BooleanType
    MAX-ACCESS            read-write -- REALLY: write-only
    STATUS            current
    DESCRIPTION
        "Reboot base station.

        This is a write-only element; only a value of TRUE is valid.
        All existing calls will be terminated abruptly.
        All components of the base station will be reinitialized
        according to the permanent contents of the Base Station database.
        The base station may be incommunicado for a period of time.

        The reinitialization may be delayed by a few seconds to allow
        for a clean shutdown.

        (From mibCtl ElementType 44 RebootBaseStation)
        Description for mibCtl Type 16 BooleanType :
            Truth value, 0=FALSE, 1=TRUE.

            This is a subset of TriStateType; no UNDEFINED value is provided.
            [Limits: 0 1 ]
        Description for mibCtl BooleanType 0 FALSE :
            False.
        Description for mibCtl BooleanType 1 TRUE :
            True.
        "
    ::= { dot20m625kmcSysScalars 18 }



dot20m625kmcBSModelNumber                   OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base Station model number.

        Base Station model number

        (From mibCtl ElementType 204 BSModelNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 19 }



dot20m625kmcBSManufactureDate               OBJECT-TYPE
```

```
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base Station manufacture date.

        Base Station manufacture date

        (From mibCtl ElementType 205 BSManufactureDate)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 20 }



dot20m625kmcBSHardwareRevision              OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Base Station hardware revision.

        Base Station hardware revision

        (From mibCtl ElementType 206 BSHardwareRevision)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcSysScalars 21 }



dot20m625kmcMiscComponents                  OBJECT IDENTIFIER    --
DESCRIPTION       "Miscellaneous Component"
    ::= { IEEE802dot20-625k-MC-MIB 2 }



dot20m625kmcAntenna                         OBJECT IDENTIFIER    --
DESCRIPTION       "Antenna"
    ::= { dot20m625kmcMiscComponents 1 }



dot20m625kmcAntennaTable                    OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcAntennaTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Antenna Table"
    ::= { dot20m625kmcAntenna 1 }



dot20m625kmcAntennaTableEntry               OBJECT-TYPE
    SYNTAX           dot20m625kmcAntennaTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcAntennaTableIndex }
    ::= { dot20m625kmcAntennaTable 1 }

dot20m625kmcAntennaTableEntry ::= SEQUENCE {
   dot20m625kmcAntennaTableIndex              INTEGER, -- AntennaAddressType
   dot20m625kmcStateOfAntenna                 INTEGER -- ComponentStateType
    }



dot20m625kmcAntennaTableIndex               OBJECT-TYPE
    SYNTAX            INTEGER -- AntennaAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 210 AntennaAddressType :
            Component antenna address.
            [Limits: 0 11 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcAntennaTableEntry 1 }



dot20m625kmcStateOfAntenna                  OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "State of Antenna as a whole.


        (From mibCtl ElementType 211 StateOfAntenna)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.

            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.

            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up.

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading.

            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction.

            Restriction state.
        "
    ::= { dot20m625kmcAntennaTableEntry 2 }



dot20m625kmcBSTemperatures                  OBJECT IDENTIFIER    --
DESCRIPTION       "BS Temperature"
    ::= { dot20m625kmcMiscComponents 3 }



dot20m625kmcBSTemperature                   OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "The temperature of Base station (degrees Celsius).

        The latest recorded temperature of a given BS.

        (From mibCtl ElementType 536 BSTemperature)
        Description for mibCtl Type 807 DegreesCelsiusType :
            Temperature in degrees Celsius.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcBSTemperatures 1 }
```

```
dot20m625kmcCableInfo                       OBJECT IDENTIFIER    --
DESCRIPTION       "Cable Info"
    ::= { dot20m625kmcMiscComponents 4 }



dot20m625kmcCableLossValueForLoCal          OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Value of cable loss for Local Oscilator.

        Value of cable loss.

        (From mibCtl ElementType 801 CableLossValueForLoCal)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcCableInfo 1 }




dot20m625kmcAntCableTable                   OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcAntCableTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Antenna Cable"
    ::= { dot20m625kmcCableInfo 2 }



dot20m625kmcAntCableTableEntry              OBJECT-TYPE
    SYNTAX           dot20m625kmcAntCableTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcAntCableTableIndex }
    ::= { dot20m625kmcAntCableTable 1 }

dot20m625kmcAntCableTableEntry ::= SEQUENCE {
   dot20m625kmcAntCableTableIndex              INTEGER, -- AntennaAddressType
   dot20m625kmcCableLossValueForAntenna        OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcAntCableTableIndex              OBJECT-TYPE
    SYNTAX            INTEGER -- AntennaAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 210 AntennaAddressType :
            Component antenna address.
            [Limits: 0 11 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcAntCableTableEntry 1 }



dot20m625kmcCableLossValueForAntenna        OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Value of cable loss for Antenna.

        Value of cable loss.

        (From mibCtl ElementType 802 CableLossValueForAntenna)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcAntCableTableEntry 2 }




dot20m625kmcGPSCableTable                   OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcGPSCableTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "GPS Cable"
    ::= { dot20m625kmcCableInfo 10 }



dot20m625kmcGPSCableTableEntry              OBJECT-TYPE
    SYNTAX           dot20m625kmcGPSCableTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcGPSCableTableIndex }
    ::= { dot20m625kmcGPSCableTable 1 }

dot20m625kmcGPSCableTableEntry ::= SEQUENCE {
   dot20m625kmcGPSCableTableIndex             INTEGER, -- GpsAddressType
   dot20m625kmcCableLengthForGps              OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcGPSCableTableIndex              OBJECT-TYPE
    SYNTAX            INTEGER -- GpsAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 209 GpsAddressType :
            Base station GPS component address.


            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcGPSCableTableEntry 1 }



dot20m625kmcCableLengthForGps               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Cable length for Gps.

        This cable is used for calibration.

        (From mibCtl ElementType 803 CableLengthForGps)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcGPSCableTableEntry 2 }



dot20m625kmcGPS                             OBJECT IDENTIFIER    --
DESCRIPTION       "GPS"
    ::= { dot20m625kmcMiscComponents 6 }



dot20m625kmcGPSTable                        OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcGPSTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "GPS Table"
    ::= { dot20m625kmcGPS 1 }



dot20m625kmcGPSTableEntry                   OBJECT-TYPE
    SYNTAX           dot20m625kmcGPSTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcGPSTableIndex }
    ::= { dot20m625kmcGPSTable 1 }

dot20m625kmcGPSTableEntry ::= SEQUENCE {
   dot20m625kmcGPSTableIndex                  INTEGER, -- GpsAddressType
   dot20m625kmcStateOfGps                     INTEGER, -- ComponentStateType
   dot20m625kmcGpsNumberOfSatelliteSeen       Unsigned32, -- Unsigned32Type
   dot20m625kmcGpsIndication                  INTEGER, -- IndicationType
   dot20m625kmcGpsSerialNumber                OCTET STRING (SIZE(0..20))
-- TextType X 20
    }



dot20m625kmcGPSTableIndex                   OBJECT-TYPE
    SYNTAX            INTEGER -- GpsAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
```

```
        Description for mibCtl Type 209 GpsAddressType :
            Base station GPS component address.


            [Limits: 0 1 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcGPSTableEntry 1 }



dot20m625kmcStateOfGps                      OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "GPS state.

        The state of the GPS (Global Positioning System)
        on the active local oscillator unit

        (From mibCtl ElementType 420 StateOfGps)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.

            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.
            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up.

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading.

            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction.

            Restriction state.
        "
    ::= { dot20m625kmcGPSTableEntry 2 }



dot20m625kmcGpsNumberOfSatelliteSeen        OBJECT-TYPE
    SYNTAX            Unsigned32 -- Unsigned32Type
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Number of satellites seen by GPS.

        The number of satellites seen by the GPS (Global Positioning System)
        on the active local oscillator unit

        (From mibCtl ElementType 421 GpsNumberOfSatelliteSeen)
        Description for mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
        "
    ::= { dot20m625kmcGPSTableEntry 3 }



dot20m625kmcGpsIndication                   OBJECT-TYPE
    SYNTAX            INTEGER -- IndicationType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Status of GPS indicator.



        (From mibCtl ElementType 530 GpsIndication)
        Description for mibCtl Type 230 IndicationType :
            Hardware indication status (LEDs).
        Description for mibCtl IndicationType 0 Off :
            Off.
        Description for mibCtl IndicationType 1 Amber :
            Amber.
        Description for mibCtl IndicationType 2 Red :
            Red.
        Description for mibCtl IndicationType 3 Green :
            Green.
        Description for mibCtl IndicationType 4 NotPresent :
            Not present.
        "
    ::= { dot20m625kmcGPSTableEntry 4 }



dot20m625kmcGpsSerialNumber                 OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "GPS serial number text.

        Factory set uniquely for each component.

        (From mibCtl ElementType 570 GpsSerialNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcGPSTableEntry 5 }



dot20m625kmcPowerAmplifier                  OBJECT IDENTIFIER    --
DESCRIPTION       "Power Amplifier"
    ::= { dot20m625kmcMiscComponents 8 }




dot20m625kmcPAUnitTable                     OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcPAUnitTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "PA Table"
    ::= { dot20m625kmcPowerAmplifier 1 }



dot20m625kmcPAUnitTableEntry                OBJECT-TYPE
    SYNTAX           dot20m625kmcPAUnitTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcPAUnitTableIndex }
    ::= { dot20m625kmcPAUnitTable 1 }

dot20m625kmcPAUnitTableEntry ::= SEQUENCE {
   dot20m625kmcPAUnitTableIndex               INTEGER, -- PAUnitAddressType
   dot20m625kmcRebootPAUnit                   INTEGER, -- BooleanType
   dot20m625kmcStateOfPAUnit                  INTEGER, -- ComponentStateType
   dot20m625kmcPAUnitIndication               INTEGER, -- IndicationType
   dot20m625kmcPAUnitSerialNumber             OCTET STRING (SIZE(0..20)),
-- TextType X 20
```

```
   dot20m625kmcPAUnitModelNumber              OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcPAUnitManufactureDate          OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcPAUnitHardwareRevision         OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcPAUnitManufactureID            OCTET STRING (SIZE(0..20)),
-- TextType X 20
   dot20m625kmcPAUnitTemperature              OCTET STRING (SIZE(0..4))
-- DegreesCelsiusType
    }



dot20m625kmcPAUnitTableIndex                OBJECT-TYPE
    SYNTAX            INTEGER -- PAUnitAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 207 PAUnitAddressType :
            Base station power amplifier component unit address.

            A power amplifier unit address is a subset of Base Station
            component addresses,
            restricted to power amplifier components only.
            Power amplifiers boost radio frequency
            signal levels.
            [Limits: 0 3 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcPAUnitTableEntry 1 }



dot20m625kmcRebootPAUnit                    OBJECT-TYPE
    SYNTAX            INTEGER -- BooleanType
    MAX-ACCESS            read-write -- REALLY: write-only
    STATUS            current
    DESCRIPTION
        "Action to reboot a PA unit.

        This is a write-only element; only a value of TRUE is valid.

        (From mibCtl ElementType 506 RebootPAUnit)
        Description for mibCtl Type 16 BooleanType :
            Truth value, 0=FALSE, 1=TRUE.

            This is a subset of TriStateType; no UNDEFINED value is provided.
            [Limits: 0 1 ]
        Description for mibCtl BooleanType 0 FALSE :
            False.
        Description for mibCtl BooleanType 1 TRUE :
            True.
        "
    ::= { dot20m625kmcPAUnitTableEntry 2 }



dot20m625kmcStateOfPAUnit                   OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "State of PA as a whole.


        (From mibCtl ElementType 212 StateOfPAUnit)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.

            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.

            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up.

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading.
            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction.

            Restriction state.
        "
    ::= { dot20m625kmcPAUnitTableEntry 3 }



dot20m625kmcPAUnitIndication                OBJECT-TYPE
    SYNTAX            INTEGER -- IndicationType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Status of PA Unit indicator.



        (From mibCtl ElementType 526 PAUnitIndication)
        Description for mibCtl Type 230 IndicationType :
            Hardware indication status (LEDs).


        Description for mibCtl IndicationType 0 Off :
            Off.
        Description for mibCtl IndicationType 1 Amber :
            Amber.
        Description for mibCtl IndicationType 2 Red :
            Red.
        Description for mibCtl IndicationType 3 Green :
            Green.
        Description for mibCtl IndicationType 4 NotPresent :
            Not present.
        "
    ::= { dot20m625kmcPAUnitTableEntry 4 }



dot20m625kmcPAUnitSerialNumber              OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "PA unit serial number text.

        Factory set uniquely for each component.

        (From mibCtl ElementType 560 PAUnitSerialNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 5 }



dot20m625kmcPAUnitModelNumber               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "PA unit model number.
        Factory set with description of component type, including
        the major revision level.

        (From mibCtl ElementType 561 PAUnitModelNumber)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 6 }



dot20m625kmcPAUnitManufactureDate           OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "PA unit manufacture date.

        Factory set to month and date of manufacture date of the module.

        (From mibCtl ElementType 562 PAUnitManufactureDate)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 7 }



dot20m625kmcPAUnitHardwareRevision          OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "PA unit hardware revision name.

        Set at the factory to indicate the minor hardware revision
        level of the module.

        (From mibCtl ElementType 563 PAUnitHardwareRevision)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 8 }



dot20m625kmcPAUnitManufactureID             OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..20)) -- TextType X 20
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "PA unit manufacture ID.



        (From mibCtl ElementType 564 PAUnitManufactureID)
        Description for mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 9 }



dot20m625kmcPAUnitTemperature               OBJECT-TYPE
    SYNTAX            OCTET STRING (SIZE(0..4)) -- DegreesCelsiusType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "The temperature of PAUnit (degrees Celsius).

        The latest recorded temperature of a given PAUnit.

        (From mibCtl ElementType 539 PAUnitTemperature)
        Description for mibCtl Type 807 DegreesCelsiusType :
            Temperature in degrees Celsius.
        Type derived from mibCtl Type 15 TextType :
            ASCII or compatible text.
        Type derived from mibCtl Type 12 OctetType :
            8 bits of raw opaque data.
        Derived from basic 8 bit word type.
        "
    ::= { dot20m625kmcPAUnitTableEntry 10 }




dot20m625kmcPAModuleTable                   OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcPAModuleTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "PA Module Table"
    ::= { dot20m625kmcPowerAmplifier 2 }



dot20m625kmcPAModuleTableEntry              OBJECT-TYPE
    SYNTAX           dot20m625kmcPAModuleTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcPAModuleTableIndex1,dot20m625kmcPAModuleTableIndex2 }
    ::= { dot20m625kmcPAModuleTable 1 }

dot20m625kmcPAModuleTableEntry ::= SEQUENCE {
   dot20m625kmcPAModuleTableIndex1            INTEGER, -- PAUnitAddressType
   dot20m625kmcPAModuleTableIndex2            INTEGER -- PAModuleAddressType
    }



dot20m625kmcPAModuleTableIndex1             OBJECT-TYPE
    SYNTAX            INTEGER -- PAUnitAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 207 PAUnitAddressType :
            Base station power amplifier component unit address.

            A power amplifier unit address is a subset of Base Station
            component addresses,
            restricted to power amplifier components only.
            Power amplifiers boost radio frequency
            signal levels.
            [Limits: 0 3 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcPAModuleTableEntry 1 }



dot20m625kmcPAModuleTableIndex2             OBJECT-TYPE
    SYNTAX            INTEGER -- PAModuleAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 208 PAModuleAddressType :
            Base station power amplifier component module address.

            A power amplifier module address is a subset of Base Station
            component addresses,
            [Limits: 0 2 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcPAModuleTableEntry 2 }



dot20m625kmcPowerSupply                     OBJECT IDENTIFIER    --
DESCRIPTION       "Power supply"
    ::= { dot20m625kmcMiscComponents 9 }



dot20m625kmcPowerSupplyTable                OBJECT-TYPE
    SYNTAX SEQUENCE OF dot20m625kmcPowerSupplyTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       "Power supply Table"
    ::= { dot20m625kmcPowerSupply 1 }



dot20m625kmcPowerSupplyTableEntry           OBJECT-TYPE
    SYNTAX           dot20m625kmcPowerSupplyTableEntry
    MAX-ACCESS            not-accessible
    STATUS            current
    DESCRIPTION       ""
    INDEX  { dot20m625kmcPowerSupplyTableIndex }
    ::= { dot20m625kmcPowerSupplyTable 1 }

dot20m625kmcPowerSupplyTableEntry ::= SEQUENCE {
   dot20m625kmcPowerSupplyTableIndex          INTEGER, -- PowerAddressType
   dot20m625kmcStateOfPowerSupply             INTEGER, -- ComponentStateType
   dot20m625kmcPowerSupplyIndication          INTEGER -- IndicationType
    }
```

```
dot20m625kmcPowerSupplyTableIndex           OBJECT-TYPE
    SYNTAX            INTEGER -- PowerAddressType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION       "
        Description for mibCtl Type 211 PowerAddressType :
            Component power supply address.
            [Limits: 0 2 ]
        Type derived from mibCtl Type 14 Unsigned32Type :
            32 bit unsigned integer.
        Type derived from mibCtl Type 11 Word32Type :
            32 bits of raw opaque data.
        Derived from basic 32 bit word type.
    "
    ::= { dot20m625kmcPowerSupplyTableEntry 1 }



dot20m625kmcStateOfPowerSupply              OBJECT-TYPE
    SYNTAX            INTEGER -- ComponentStateType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "State of Power supply as a whole.


        (From mibCtl ElementType 213 StateOfPowerSupply)
        Description for mibCtl Type 71 ComponentStateType :
            Component operational state.

            A component begins in the Unknown state.
            If not detected, it enters and remains in the NotPresent state.
            If detected, it enters the Uninitialized state, from where it
            may go to the Testing and Initializing states and then to the
            Standby or Operating state depending upon permissions.
            Due to loss of permissions or resources, it may revert from
            the Operating state to the Standby state.
            Due to failure or loss of permission, it may revert to the
            Uninitialized state, perhaps by way of the ShuttingDown state
            depending on the device.
            From the Uninitialized state it may return to more advanced
            states depending upon permissions.
            In case of a waiting period before (again) initializing,
            the component is considered to be Initializing.

            Permissions include administrative permissions (from the
            operator); excessive failure restrictions; etc.
        Description for mibCtl ComponentStateType 0 Unknown :
            Component state not known.
        Description for mibCtl ComponentStateType 1 NotPresent :
            Component is not present.
        Description for mibCtl ComponentStateType 2 PowerOff :
            Component is present but powered off.
        Description for mibCtl ComponentStateType 3 Uninitialized :
            Component is present but not in use.

            The power on/off state of the component is not specified in
            this case.
        Description for mibCtl ComponentStateType 4 Testing :
            Component is being tested.
        Description for mibCtl ComponentStateType 5 Initializing :
            Component is being initialized.
        Description for mibCtl ComponentStateType 6 Ready :
            Component is ready but not operating.
        Description for mibCtl ComponentStateType 7 Operating :
            Component is operating for normal use without restriction.
            The component is either in actual use or may be used at any time,
            without restriction.
        Description for mibCtl ComponentStateType 8 Abandoned :
            Component state is not the desired state due to excessive errors.

            The component state is not that desired, and the Base Station
            software has abandoned attempts to place the component in
            the desired state.
            The actual state of the component is undefined.
            The Base Station software will resume attempting to place the
            component in the desired state if the appropriate Reinitialize
            action element is written with the correct value.
            Also, the software may resume attempts under other conditions,
            not all of which may be documented.
        Description for mibCtl ComponentStateType 9 InitialSetUp :
            Component is initial set up.

            Initial set up state.
        Description for mibCtl ComponentStateType 10 Degrading :
            Component is degrading.

            Degrading state.
        Description for mibCtl ComponentStateType 11 Restriction :
            Component is restriction.

            Restriction state.
        "
    ::= { dot20m625kmcPowerSupplyTableEntry 2 }



dot20m625kmcPowerSupplyIndication           OBJECT-TYPE
    SYNTAX            INTEGER -- IndicationType
    MAX-ACCESS            read-only
    STATUS            current
    DESCRIPTION
        "Status of Power Supply indicator.



        (From mibCtl ElementType 527 PowerSupplyIndication)
        Description for mibCtl Type 230 IndicationType :
            Hardware indication status (LEDs).


        Description for mibCtl IndicationType 0 Off :
            Off.
        Description for mibCtl IndicationType 1 Amber :
            Amber.
        Description for mibCtl IndicationType 2 Red :
            Red.
        Description for mibCtl IndicationType 3 Green :
            Green.
        Description for mibCtl IndicationType 4 NotPresent :
            Not present.
        "
    ::= { dot20m625kmcPowerSupplyTableEntry 3 }


END
```