



Management Information Base

Technical Information Guide

Part Number MN/MID-MIB.IOM

Revision 1

Copyright © Comtech EF Data Corporation, 2002.
All rights reserved.
Printed in the USA.

Comtech EFData, 2114 West 7th Street, Tempe, Arizona, 85281, USA, 480.333.2200, FAX: 480.333.2161



Management Information Base Technical Information Guide

Part Number **MN/MID-MIB.IOM**

Revision 1
February 28, 2002

Comtech EF Data is an ISO 9001
Registered Company.



Network Customer Support

The Network Customer Support Plan identifies the steps to be followed in resolving the Customer's concern.

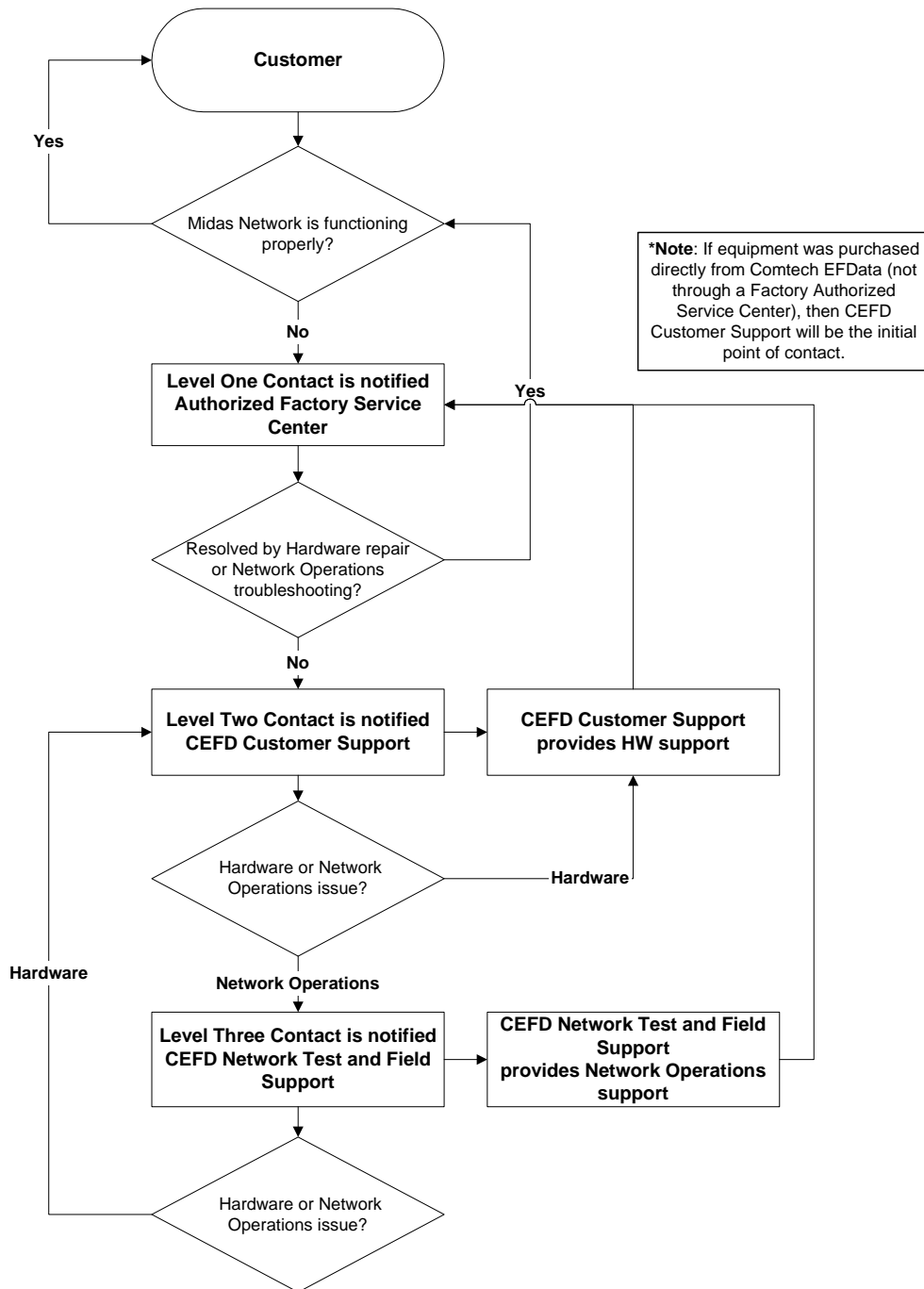
The resolution efforts will follow these levels of contact:

- **Level One Contact** – Factory Authorized Service Center.
- **Level Two Contact** – Comtech EF Data Customer Support.
- **Level Three Contact** – Network Test and Field Support

Procedural Steps

Step	Procedure
1	The Customer raises a concern with the Level One Contact .
2	The Level One Contact will perform <i>Hardware</i> repairs and <i>Network Operations</i> troubleshooting in accordance with the Comtech EF Data Service Center agreement.
3	If the Level One Contact is unable to resolve the concern, then the Level One Contact will inform the Level Two Contact of the concern in accordance with the instructions found within the attached Comtech EF Data Customer Support Department's document.
4	The Level Two Contact will enter the concern into the Comtech EF Data database and determine whether the concern is a <i>Hardware</i> concern or a <i>Network Operations</i> concern
5	The Level Two Contact will interface with the Level One Contact and provide the appropriate hardware support and enter all correspondence into the Comtech EF Data database.
6	If the Level Two Contact determines that the concern is a <i>Network Operations</i> concern, then the Level Two Contact will inform the Level Three Contact .
7	The Level Three Contact will interface with the Level One Contact and provide the appropriate support and enter all correspondence into the Comtech EF Data database.
8	If the Level Three Contact determines that there is a <i>Hardware</i> failure then the Level Three Contact will inform the Level Two Contact . Go to Step 5.

Network Support Customer Plan



See the Comtech EF Data website at <http://www.comtechefdata.com> for contact information for a Factory Authorized Service Center. Contact the Factory Authorized Service Center for:

- Product support
- Information on upgrading or returning a product

Contact the Comtech EF Data Customer Support Department for:

- Product support or training
- Information on upgrading or returning a product

A Customer Support representative may be reached at:

Comtech EF Data
Attention: Customer Support Department
2114 West 7th Street
Tempe, Arizona 85281 USA

480.333.2200 (Main Comtech EF Data Number)
480.333.4357 (Customer Support Desk)
480.333.2500 FAX

or, E-Mail can be sent to the Customer Support Department at:

service@comtechefdata.com

1. To return a Comtech EF Data product (in-warranty and out-of-warranty) for repair or replacement:
2. Request a Return Material Authorization (RMA) number from the Comtech EF Data Customer Support Department.
3. Be prepared to supply the Customer Support representative with the model number, serial number, and a description of the problem.
4. To ensure that the product is not damaged during shipping, pack the product in its original shipping carton/packaging.
5. Ship the product back to Comtech EF Data. (Shipping charges should be prepaid.)

Contact the Comtech EF Data Network Test and Field Support

- System level Network Operations support
- Information on upgrading Network Operation software
- Reporting comments or suggestions concerning manuals

A Network Test and Field Support representative may be reached at:

Comtech EF Data
Attention: Network Test and Field Support
2114 West 7th Street
Tempe, Arizona 85281 USA

480.225.2200 (Main Comtech EF Data Number)
480.225.3693 (Network Test and Field Support)
480.333.2161 FAX

or, E-Mail can be sent to the Network Customer Support Department at:

<mailto:midasfss@comtechefdata.com>

Contact us via the web at www.comtechefdata.com.

About this Manual

This manual provides a description of the Management Information Base (MIB) application. In order to manage SNMP-capable devices, Network Management platforms require access to a collection of MIB definition files, which have to be “compiled” into the platform. This means parsing and validating all required MIB files and loading them into an internal data structure, after which they can be used for actual work of the application.

Reporting Comments or Suggestions Concerning this Manual

Comments and suggestions regarding the content and design of this manual will be appreciated. To submit comments, please contact the:

Comtech EF Data Technical Publications Department: techpub@comtechefdata.com

Comtech EF Data Network Customer Support Department:

midasfss@comtechefdata.com

Overview of Changes to Previous Editions

Appendix A – Replace listing with version February 2002.

Disclaimer

Comtech EF Data has reviewed this manual thoroughly in order that it will be an easy-to-use guide to your equipment. All statements, technical information, and recommendations in this manual and in any guides or related documents are believed reliable, but the accuracy and completeness thereof are not guaranteed or warranted, and they are not intended to be, nor should they be understood to be, representations or warranties concerning the products described.

Inappropriate use of the MIB outside the provided MIDAS software can, impair, prohibit, or crash the MIDAS Controller.

Further, Comtech EF Data reserves the right to make changes in the specifications of the products described in this manual at any time without notice and without obligation to notify any person of such changes.

If you have any questions regarding your equipment or the information in this manual, please contact the Comtech EF Data Network Customer Support Department.

Table of Contents

CHAPTER 1. INTRODUCTION	1-1
MANAGEMENT INFORMATION BASE OVERVIEW.....	1-2
MIDAS SNMP-MIB.....	1-2
CHAPTER 2. MIDAS SNMP AGENT HELP.....	2-1
REFERENCE DOCUMENTS.....	2-1
TERMS AND ACRONYMS	2-1
SNMP AGENT HELP	2-2
APPENDIX A. MIB LISTING	A-1

This page is intentionally left blank.

Management Information Base Overview	1-1
MIDAS SNMP-MIB	1-2

The Multimedia Integrated Digital Access System (MIDAS) is a sophisticated power and transponder bandwidth management system for digital satellite communication networks. MIDAS software, Version 4.X.X, provides the signaling and control elements for the network, manages power, transponder bandwidth, and circuits on a demand basis. MIDAS manages and allocates satellite network resources efficiently and cost effectively.

Management Information Base (MIB) Overview

The Management Information Base (MIB), included with MIDAS 4.X (and subsequent), is the mechanism prepared for other applications to communicate with the MIDAS System. By using a standard mechanism (MIB) and a common transport (SNMP), MIDAS can receive and respond to requests from a variety of commercial and customer applications.

MIDAS SNMP-MIB

Comtech EF Data suggests that the parameters and settings be maintained per the factory specifications. Inappropriate use of the MIB files can impair, prohibit, or crash the MIDAS Controller.

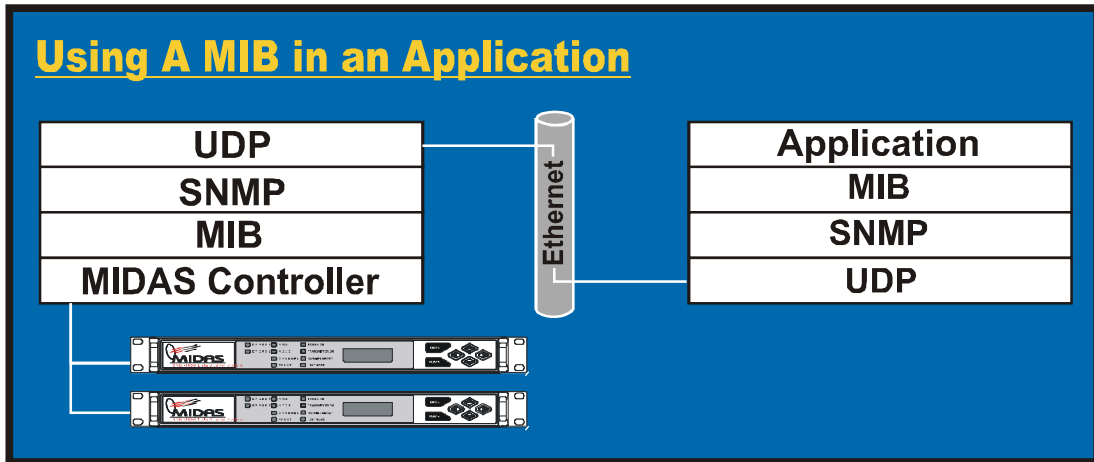


Figure 1-1. MIB Format



2. MIDAS SNMP Agent Help

Reference Documents	2-1
Terms and Acronyms	2-1
SNMP Agent Help	2-2

Referenced Documents

RFC1155	SMIv1
RFC1157	Simple Network Management Protocol
RFC2578	SMIv2
RFC1212	Concise MIB Definitions
RFC1213	MIB-II
RFC2579	Textual Conventions
MN/MIDAS40.OG	MIDAS 4.0 Operator's Guide
MN/MID.SOFTWR.IOM	MIDAS NMS Operator's Guide
MIDAS MIB	

Terms and Acronyms

MidasOID	.iso.org.dod.internet.private.enterprise.comtechefdata.midas (.1.3.6.1.4.1.6247)
-----------------	--

SNMP Agent Help

Question	Response
How to configure system to receive traps from MIDAS.	<p>The IP address and port should be added into the TrapIPAddr list under [Application] located in server.ini:</p> <p>For example: TrapIPAddr=10.6.7.109:162,10.6.7.20:3737,10.6.7.109:8888 Where: 10.6.7.109.162.10.6.7.20:3737 and 10.6.7.109:8888 will receive traps from MIDAS. Character “,” is used to separate pair IP address and port. Character “:” is used to separate IP address and port.</p>
How to handle trap missing in MIDAS.	<p>There is a trap sequence number object defined in all MIDAS traps to identify whether there is any trap missing.</p> <p>Object “MidasOID.snmpAgent.maxTrapsKept” (1.3.6.1.4.1.6247.2.6.8) Where: To setup the number of traps sent (which are saved in MIDAS). The table “MidasOID.snmpAgent.trapTable” (.1.3.6.14.1.6247.2.6.9) is used to save recently sent traps. There is a sequence number object in the table to match the sequence number of the sent trap. If any trap is missing, it can be read from the trap table.</p>
Site Type Table Operations	<p>Site TypeOID = MidasOID.siteConfigurations.siteTypes.siteTypeTable. <u>Site Type Table Entry</u></p> <p>Items in Site type table are not allowed to be deleted. The example as follows is used to illustrate how to insert Site Types into the Site Table.</p> <p>Message 1 : (SiteTypeOID.1.1 1) (SiteTypeOID.2.1 “st1”)(SiteTypeOID.3.1 “st1 desc”) (SiteTypeOID.101.1 “st1”)(SiteTypeOID.102.1 “-10”) To insert site type: st1 and the power between st1 to st1(SiteTypeOID.102.1 “-10”).</p> <p>Message 2: (SiteTypeOID.1.2 2) (SiteTypeOID.2.2 “st2”)(SiteTypeOID.3.2 “st2 desc”)(SiteTypeOID.111.2 “st2”)(SiteTypeOID.112.2 “-10”) To insert site type: st2 and the power between st2 to st2(SiteTypeOID.112.2 “-10”).</p>
Site Table Operation	<p>Site Type Table shall be inserted before Site Tables, which belong to the Site Type, and as follows:</p> <ul style="list-style-type: none"> • There shall always be a NMS site in the database. • NMS site cannot be delete. • There is only one NMS site in the system. • NMS site does not have to have node. • A site cannot be deleted if there are some nodes which is residing in that site. • All of nodes under Site shall be deleted prior to deletion of the Site Table.

Node Table Operation	<p>Site Table must be inserted before nodes under the site are inserted. All channels under the node must be deleted before the node is deleted.</p> <p>A node is created in the disabled state and does not become active until it is enabled. Once enabled, the node will be polled and placed online. On an enabled (active) node, channels can be set up and respond to calls. Enabled Node cannot be disabled if there carry an active call. A node cannot be enabled/disabled when the controller mode is not Normal.</p> <p>Under node table entry, object nodeEnabled is used to enable/disable node, object nodeOperationTag is used to delete item in node table.</p>
Channel Table Operation	<p>Node shall be inserted prior to Channel table (under the node). Before a channel is deleted all of the table rows which are associated with the channel such as circuit table, channel hunt group table, directory table, and MultiPointMember table have to be deleted.</p> <p>When deleting a channel:</p> <ol style="list-style-type: none"> 1. Make sure there is no directory number associated with the particular channel. 2. Make sure there is no Hunt Groups contained in the channels. 3. Make sure there is no circuit that uses the channel. <p>Enabled Channels cannot be disabled if there is an active call related to them. Channels cannot be enabled/disabled when the controller mode is not Normal.</p> <p>Under channel table entry, object channelEnabled is used to enable/disable channel, object channelOperationTag is used to delete item in channel table.</p>
How to Enable the Predefined Circuits in MIDAS	<p>To set object: “MidasOID.connections.circuits.circuitTable.circuitTableEntry.circuitEnabled” (.1.3.6.1.4.1.6247.2.4.4.1.1.9) to be true (2).</p>
Directory Table Operation	<p>Before any item in the Directory table is deleted or updated, all of the items related to it in other tables such as Circuit, Channel (DefaultHGrpDir), and MultiPointMember shall be deleted or updated. The item in the Directory table can point to either channel or Hunt Group. The HuntGroupID field is used to distinguish them. The field should be string '** none **' if it points to channel, otherwise it shall be HuntGroup.</p>
Hunt Group Table Operation	<p>Before any item in HuntGroup table is deleted or updated, all of the items related to it in other tables such as Directory shall be deleted or updated and all of members in the Hunt Group which are in ChannelHunt table shall be deleted.</p> <p>Under the table object hgrpSelectDirect allows the operator to select the searchorder (the order in which the channels will be called). Choices include: Forward, Reverse, or Pack.</p> <p>Forward and Reverse options - the search algorithm will continue moving the index in the forward or reverse direction for the next available channel. The index will remain at its last available slot.</p> <p>Pack option - will cause the search algorithm to always search from the top for the next available channel.</p>
How to Shut Down the Controller or Bring Controller to Running Mode.	<p>To shut down controller, set object MidasOID.system.controllerStatus.ctrlsMode to be mode_shuttingdown if the object equals to mode_Normal.</p> <p>To bring controller to running mode from shut down or exception mode, set object MidasOID.system.controllerStatus.ctrlsMode to be mode_Normal.</p>
How to Switch	<p>To switch controller between online and standby, set: Object</p>

Between Online and Standby Controller	MidasOID.system.systemSetup.sysRedundantController.redundantCtrlSwitchover to be true (2). In order to perform a redundant switchover operation, the online controller must be in 'normal' state for the mode and standby controller must be in "monitoring" state for mode.
How to Send Text String to Remote Sites from the Hub.	<p>Use the Service Message option to send a message or a command from the Operator Workstation to the remote node terminal, and directly to the node. The message can be up to 28 characters in length.</p> <p>All commands issued to a remote node shall begin with a prefix of "IDC>". Refer to the IDC-150 Network Control Modem Operation and Maintenance Manual for a complete listing of all commands supported. The purpose of this option is to allow simple, text-based communication between the remote node and the operator and to provide the operator with remote command capability.</p> <p>Any text string of up to 28 characters that can be accepted from the remote node terminal through the user port can be sent as a service message. The operator can send such commands to a remote node for configuration or call requests. All service messages from the remote node to the NMS are displayed in the Event Log and sent out by SNMP traps. Similarly, all responses for commands originating from the Operator Workstation are displayed in the Event Log.</p> <p>Under "MidasOID.maintenance.serviceMessage" (.1.3.6.1.4.1.6247.2.7.1), object "toNode" is the node, which the text string is sent. Object "OutStr" is the text string sent to remote node and is limited less than 28 characters. When object "send" is set true (2), the text string in object "outStr" will be sent to remote node which is defined in object "toNode".</p>
How to Receive Text String from Remote Sites	<p>Any text string (less than 28 characters) sent by remote sites will be received by the hub and sent out by the trap.</p> <p>The customer system can receive text string from remote sites by configuring MIDAS to send a trap. In MIDAS trap type 2 (Midas value change trap), the first object in variable list of trap defines the remote node which is sending the text string, the second object in variable list of trap defines text string sent by remote node. Object ID of the first object in trap type 2 shall be "MidasOID.maintenance.serviceMessage.toNode" (.1.3.6.1.4.1.6247.2.7.1.1).</p>



Appendix A. MIB Listing

Notes:

1. The following MIB listing is recorded as of March 4, 2002.
2. This manual is applicable to all configurations incorporating versions up to 4.3.

```

MIB-MIDAS DEFINITIONS ::= BEGIN
IMPORTS
OBJECT-TYPE FROM RFC-1212
TRAP-TYPE FROM RFC-1215
enterprises FROM RFC1155-SMI
;

BOOLEAN ::= INTEGER {
    false (1),
    true (2)
}

BYTE ::= INTEGER (0..255)

LONGINT ::= INTEGER (0..2147483647)

OCTETSTRING15 ::= OCTET STRING (SIZE(0..15))

OCTETSTRING20 ::= OCTET STRING (SIZE(0..20))

OCTETSTRING256 ::= OCTET STRING (SIZE(0..256))

OCTETSTRING28 ::= OCTET STRING (SIZE(0..28))

OCTETSTRING30 ::= OCTET STRING (SIZE(0..30))

OCTETSTRING80 ::= OCTET STRING (SIZE(0..80))

REAL ::= OCTET STRING (SIZE(0..20))

SMALLINT ::= INTEGER (-32768..32767)

TACTIVATIONSOURCE ::= INTEGER {
    activationSrc-RTS (1),
    activationSrc-UserRemote (2),
    activationSrc-Hub (3),
    activationSrc-slottedRemote (4),
    activationSrc-slottedHub (5)
}

TACTIVATIONTYPE ::= INTEGER {
    activation-Perm (1),
    activation-Sche (2),
    activation-Remote (3),
    activation-RTS (4)
}

TACTIVESTANDBYSTATUS ::= INTEGER {
    waitForConfig (1),
    offline (2),
    active (3),
    suspending (4),
    transferQ (5),
    standby (6),
    acceptQ (7),
    restore (8),
}

```

```

        preStandby (9),
        acceptDB (10),
        sendDB (11),
        waitPassSelfTest (12)
    }

TBWUSAGE ::= INTEGER {
    bw-Unused (1),
    bw-Public (2),
    bw-Private (3),
    bw-Overflow (4)
}

TCARRIERLOSSACTION ::= INTEGER {
    hold (1),
    nominal (2),
    max (3)
}

TCHANNELID ::= INTEGER (0..30)

TCHANNELNCSSTATE ::= INTEGER {
    channelncs-CIP (1),
    channelncs-Idle (2),
    channelncs-Fault (3),
    channelncs-Warning (4),
    channelncs-Disabled (5)
}

TCHANNELSTATUS ::= INTEGER {
    channel-Inactive (1),
    channel-Activating (2),
    channel-Active (3),
    channel-Deactivating (4)
}

TCHANNELTYPE ::= INTEGER {
    channel-Data (1),
    channel-ISDN (2),
    channel-Voice (3),
    channel-Video (4)
}

TCIRCUITID ::= INTEGER

TCIRCUITSTATUS ::= INTEGER {
    circuit-Inactive (1),
    circuit-Activating (2),
    circuit-Active (3),
    circuit-Deactivating (4),
    circuit-Fault (5),
    circuit-Scheduled (6),
    circuit-Retry (7)
}

TCIRCUITTYPE ::= INTEGER {
    circuit-Data (1),

```

```

        circuit-ISDN (2),
        circuit-Voice (3),
        circuit-Video (4),
        circuit-Reject (5),
        circuit-OffTransponder (6),
        circuit-Dummy (7)
    }

TCLOCKMODE ::= INTEGER {
    tclockmode-terrestrial (1),
    tclockmode-internal (2),
    tclockmode-satellite (3),
    tclockmode-extref (4)
}

TCONNECTIONID ::= INTEGER (1..9999)

TCONNECTIVITY ::= INTEGER {
    conn-Duplex (1),
    conn-Simplex (2),
    conn-MultiDuplex (3),
    conn-MultiSimplex (4),
    conn-Broadcast (5)
}

TCONTROLCHANNELSTATUS ::= INTEGER {
    cc-Inactive (1),
    cc-Activating (2),
    cc-Active (3),
    cc-Deactivating (4),
    cc-Fault (5)
}

TCONTROLCHANNELTYPE ::= INTEGER {
    ctrl-master (1),
    ctrl-slave (2),
    ctrl-local (3)
}

TCONTROLLERMODE ::= INTEGER {
    mode-Unknown (1),
    mode-Initializing (2),
    mode-Startup (3),
    mode-WaitBeforeStartup (4),
    mode-CallEstablish (5),
    mode-Normal (6),
    mode-Reset (7),
    mode-Maintenance (8),
    mode-ChangeControl (9),
    mode-ShuttingDown (10),
    mode-ShuttDown (11),
    mode-NetFail (12),
    mode-WaitingForUpload (13),
    mode-DetectingOutBound (14),
    mode-SecondaryWaitingForUpload (15),
    mode-SecondaryMonitoring (16),
    mode-CtrlChnsSelfTest (17),

```

```

        mode-Reserved1 (18),
        mode-Reserved2 (19),
        mode-Reserved3 (20),
        mode-Exception (21),
        mode-SettingException (22),
        mode-RetryNormal (23),
        mode-Terminated (24)
    }

TDATE ::= OCTET STRING (SIZE(0..10))

TDVBFRAME ::= INTEGER {
    dvb187 (1),
    dvb188 (2),
    dvb204 (3)
}

TENABLEDDISABLEDSTATUSTYPE ::= INTEGER {
    disabled (1),
    hasenabled (2),
    enabling (3),
    disabling (4),
    enabledNoAck (5),
    disabledNoAck (6),
    reserved (7)
}

TENABLEDKILLTYPE ::= INTEGER {
    enabledkill-enabled (1),
    enabledkill-disabled (2),
    enabledkill-killed (3)
}

TENCODE ::= INTEGER {
    encode-Viterbi (1),
    encode-Sequential (2),
    encode-Turbo (3),
    encode-Viterbi-RS (4),
    encode-Sequential-RS (5),
    encode-TCM (6),
    encode-TCM-RS (7),
    encode-None (8)
}

TFECRATE ::= INTEGER {
    fec12 (1),
    fec34 (2),
    fec23 (3),
    fec78 (4),
    fec56 (5),
    fec516 (6),
    fec2144 (7),
    fec11 (8),
    fec89 (9)
}

TFREQBAND ::= INTEGER {

```

```

        sbCBand (1),
        sbKuBand (2)
    }

THALFLONGWORD ::= INTEGER (0..2147483647)

THGSELECTTYPE ::= INTEGER {
    select-reverse (1),
    select-direct (2),
    select-pack (3)
}

THUNTGROUPTYPE ::= INTEGER {
    hgrp-data (1),
    hgrp-ISDN (2),
    hgrp-Voice (3)
}

TINTERNALEXTERNAL ::= INTEGER {
    intExt-NA (1),
    intExt-Internal (2),
    intExt-External (3)
}

TISDNGRADEOFSERVICE ::= INTEGER {
    g1BD-X31-12 (1),
    g1BD-X31-34 (2),
    g1BD-X31-78 (3),
    g2BD-X31-12 (4),
    g2BD-X31-34 (5),
    g2BD-X31-78 (6)
}

TKEYSTATUS ::= INTEGER {
    key-OK (1),
    key-Expired (2),
    key-NotExist (3)
}

TMODEMCHANNEL ::= INTEGER {
    channel-efsms301 (1),
    channel-efsdm100 (2)
}

TMODEMTYPE ::= INTEGER {
    modem-none (1),
    modem-SDM100 (2),
    modem-SDM150 (3),
    modem-SDM300 (4),
    modem-SDM6000 (5),
    modem-SDM8000 (6),
    modem-SDM9000 (7),
    modem-IDC150D (8),
    modem-SNM1010 (9),
    modem-SNM1020 (10),
    modem-SNM2200 (11),
    modem-SNTK1010 (12),

```

```

        modem-SNTK1020 (13),
        modem-SNTC1010 (14),
        modem-SNTC1020 (15),
        modem-SDM2020M (16),
        modem-SDM2020D (17),
        modem-SDM300L (18),
        modem-CDM550 (19),
        modem-CDM600 (20),
        modem-SNM1010L (21),
        modem-CIM550 (22),
        modem-CIM300L (23),
        modem-CDM550T (24),
        modem-SDM300A (25)
    }

TMODULATION ::= INTEGER {
    mod-BPSK (1),
    mod-QPSK (2),
    mod-PSK8 (3),
    mod-QAM16 (4),
    mod-OQPSK (5)
}

TNCSSSTATE ::= INTEGER {
    ncs-OnlineCIP (1),
    ncs-Online (2),
    ncs-Fault (3),
    ncs-Warning (4),
    ncs-Disabled (5),
    ncs-TempFlag (6)
}

TNODEENABLESTATE ::= INTEGER {
    node-Inactive (1),
    node-Activating (2),
    node-Active (3),
    node-Deactivating (4)
}

TNODEID ::= INTEGER (1..9999)

TOPERATION ::= INTEGER {
    operation-none (1),
    operation-delete (2)
}

TPOLLTYPE ::= INTEGER {
    poll-Aloha (1),
    poll-TDMA (2)
}

TPRESET ::= INTEGER {
    tpreset-non (1),
    tpreset-a (2),
    tpreset-b (3),
    tpreset-c (4),
    tpreset-d (5),

```

```

        tpreset-v (6)
    }

TPRIMARYSTATUS ::= INTEGER {
    undef (1),
    nbk (2),
    primary (3),
    secondary (4),
    preSecondary (5)
}

TTIME ::= OCTET STRING (SIZE(0..8))

TUNITS ::= INTEGER {
    unit-Unknown (1),
    unit-kHz (2),
    unit-kbitsPerSec (3),
    unit-bitsPerSec (4),
    unit-second (5),
    unit-ms (6),
    unit-bits (7),
    unit-SymbolPerSec (8)
}

TVIDEOGRADEOFSERVICE ::= INTEGER {
    g128-12 (1),
    g128-34 (2),
    g256-12 (3),
    g256-34 (4),
    g384-12 (5),
    g384-34 (6),
    g512-12 (7),
    g512-34 (8),
    g768-12 (9),
    g768-34 (10)
}

TVPROTOCOL ::= INTEGER {
    g728 (1),
    g729 (2)
}

WORD ::= INTEGER (0..65535)

comtechefdata OBJECT IDENTIFIER ::= { enterprises 6247 }

--
--                               MidasOID =
.iso.org.dod.internet.private.enterprises.comtechefdata.midas
--                               (.1.3.6.1.4.1.6247)
--                               How to configure system to receive traps from
Midas
--                               If any IP address and port are supposed to
receive traps from Midas, the IP
--                               address and port should be added into
TrapIPAddr list under [Application]
--                               in server.ini.

```

```

--                                     For example
TrapIPAddr=10.6.7.109:162,10.6.7.20:3737,10.6.7.109:8888, It
--                                     means that 10.6.7.109:162,10.6.7.20:3737 and
10.6.7.109:8888 will receive
--                                     traps from Midas. Character ',' is used to
separate pair of IP address and
--                                     port. Character ':' is used to separate IP
address and port.
--                                     How to handle trap missing in Midas
--                                     There is trap sequence number object defined
in all of Midas traps to identify
--                                     whether there is any trap missing.
--                                     Object '.MidasOID.snmpAgent.maxTrapsKept'
(1.3.6.1.4.1.6247.2.6.8) is used
--                                     to setup the number of traps sent which are
saved in Midas. The table
--                                     'MidasOID.snmpAgent.trapTable'
(.1.3.6.1.4.1.6247.2.6.9) is used to save
--                                     traps sent recently. There is sequence
number object in the table to match
--                                     sequence number of trap sent. If any trap is
missing, It can be read from the
--                                     trap table.
--                                     How to send text string to remote sites from hub
--                                     Under 'MidasOID.maintenance.serviceMessage'
(.1.3.6.1.4.1.6247.2.7.1),
--                                     object 'toNode' is the node which the text
string is sent to. Object 'OutStr' is the
--                                     text string sent to remote node and is
limited less than 28 characters. When
--                                     object 'send' is set true (2), the text
string in object 'outStr' will be sent to
--                                     remote node which is defined in object
'toNode'.
--                                     How to receive text sting from remote sites
--                                     Any text string (less than 28 characters)
sent by remote sites will be received
--                                     by hub and sent out by trap.
--                                     The customer system can receive text string
from remote sites by configurating
--                                     Midas to send trap to it. In Midas trap type
2 (Midas value change trap), the
--                                     first object in variable list of trap
defines the remote node which is sending
--                                     the text string, the second object in
variable list of trap defines text string sent
--                                     by remote node. Object ID of the first
object in trap type 2 should be
--                                     'MidasOID.maintenance.serviceMessage.toNode'
(.1.3.6.1.4.1.6247.2.7.1.1).
--                                     How to enable the predefined circuits in Midas.
--                                     To set object
--                                     'MidasOID.connections.circuits.circuitTable.circuitTableEntry.circuitEnabled'
--                                     (.1.3.6.1.4.1.6247.2.4.4.1.1.9) to be true
(2)

```

midas OBJECT IDENTIFIER ::= { comtechefdata 2 }

system OBJECT IDENTIFIER ::= { midas 1 }

sysGroup OBJECT IDENTIFIER ::= { system 1 }

sysgrpSysDescr OBJECT-TYPE
SYNTAX OCTETSTRING80
ACCESS read-write
STATUS mandatory
::= { sysGroup 1 }

sysgrpSysObjectID OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
ACCESS read-write
STATUS mandatory
::= { sysGroup 2 }

sysgrpSysUptime OBJECT-TYPE
SYNTAX TTIME
ACCESS read-only
STATUS mandatory
::= { sysGroup 3 }

sysgrpSysContact OBJECT-TYPE
SYNTAX OCTETSTRING80
ACCESS read-write
STATUS mandatory
::= { sysGroup 4 }

sysgrpSysName OBJECT-TYPE
SYNTAX OCTETSTRING80
ACCESS read-write
STATUS mandatory
::= { sysGroup 5 }

sysgrpSysLocation OBJECT-TYPE
SYNTAX OCTETSTRING80
ACCESS read-write
STATUS mandatory
::= { sysGroup 6 }

sysgrpSysServices OBJECT-TYPE
SYNTAX THALFLONGWORD
ACCESS read-write
STATUS mandatory
::= { sysGroup 7 }

sysgrpSysVersion OBJECT-TYPE
SYNTAX OCTETSTRING20
ACCESS read-only
STATUS mandatory
DESCRIPTION
"version number of MIDAS controller"
::= { sysGroup 8 }

```

--
MIDAS system(redundant
--
--
all OID with '0' at the end
--
provide modes
--
controllers

```

```

the branch provides info about
or non-redundant)
If MIDAS is non-redundant, then
are meaningless. Otherwise, oids
and status of the MIDAS

```

```

controllerStatus OBJECT IDENTIFIER ::= { system 2 }

```

```

ctrlstMode OBJECT-TYPE

```

```

SYNTAX TCONTROLLERMODE

```

```

ACCESS read-write

```

```

STATUS mandatory

```

```

DESCRIPTION

```

```

"Midass controller mode

```

```

1. To shutdown controller

```

```

If the object = mode_Normal then

```

```

set the object = mode_shuttingdown.

```

```

2. To bring controller to be normal from shutdown

```

```

If the object in [mode_shutdown, mode_exception,

```

```

mode_shuttingdown] then

```

```

set the object = mode_Normal"

```

```

::= { controllerStatus 1 }

```

```

ctrlstPrimaryStatus OBJECT-TYPE

```

```

SYNTAX TPRIMARYSTATUS

```

```

ACCESS read-only

```

```

STATUS mandatory

```

```

DESCRIPTION

```

```

"If this value is nbk, the MIDAS controller is operating

```

```

in non-redundant setup. All oids that end with '0' will be

```

```

meaningless.

```

```

If the value is primary, the controller is online taking

```

```

call request. If the value is secondary, the controller

```

```

is standby monitoring the online controller."

```

```

::= { controllerStatus 2 }

```

```

ctrlstActiveStandbyStatus OBJECT-TYPE

```

```

SYNTAX TACTIVESTANDBYSTATUS

```

```

ACCESS read-only

```

```

STATUS mandatory

```

```

DESCRIPTION

```

```

"the internal states of online controller"

```

```

::= { controllerStatus 3 }

```

```

ctrlstMode0 OBJECT-TYPE

```

```

SYNTAX TCONTROLLERMODE

```

```

ACCESS read-only

```

```

STATUS mandatory

```

```

DESCRIPTION

```

```

"In a redundant setup, this object shows

```

```

the other controller mode"

```

```

::= { controllerStatus 4 }

```

```

ctrlstPrimaryStatusO OBJECT-TYPE
    SYNTAX      TPRIMARYSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "In a redundant setup, Primary Status
        of the other controller"
    ::= { controllerStatus 5 }

ctrlstActiveStandbyStatusO OBJECT-TYPE
    SYNTAX      TACTIVESTANDBYSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "the internal states of the other controller"
    ::= { controllerStatus 6 }

ctrlstConfiguredAsO OBJECT-TYPE
    SYNTAX      TPRIMARYSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "If this object is primary or secondary, then MIDAS
        is a redundant system. If this object is secondary,
        then the current controller is configured as primary.
        If this object is primary then the current controller
        is configured as secondary"
    ::= { controllerStatus 7 }

ctrlstStartTime OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The starting running time of this controller"
    ::= { controllerStatus 8 }

ctrlstStartDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The starting running date of this controller"
    ::= { controllerStatus 9 }

--
pertaining
--
redundant operation

systemSetup OBJECT IDENTIFIER ::= { system 3 }

sysParameters OBJECT IDENTIFIER ::= { systemSetup 1 }

sysParaSatelliteName OBJECT-TYPE
    SYNTAX      OCTETSTRING20
    ACCESS      read-write

```

```

STATUS      mandatory
DESCRIPTION
    "Satellite name Midas is operating on"
 ::= { sysParameters 1 }

sysParaAllocationFactor  OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is a factor, or multiplier, used as a carrier
    spacing that is placed between carrier assignments
    to avoid interference (roll-offs, skirts). A typical
    Allocation Factor is 1.4. Using a larger number
    will provide a larger spacing, but will also take up
    valuable transponder space. The operator will need
    to determine what works best in their particular
    situation."
 ::= { sysParameters 2 }

sysParaStepSize  OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the smallest unit of bandwidth that will be
    displayed and allocated on the Space Segment.
    The available Step Sizes are
    2.5 to 25 kHz, in 2.5 kHz increments."
 ::= { sysParameters 3 }

sysParaLogDays  OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the
    number of days that logs will be kept before
    they are deleted from the system. The operator
    will set this according their specific record
    keeping needs."
 ::= { sysParameters 4 }

sysParaEnablePowerManagement  OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "the power mamangement is turned on."
 ::= { sysParameters 5 }

sysParaEnableAFC  OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "the AFC(automatical frequency control) is turned on."

```

```

 ::= { sysParameters 6 }

sysParaLockTime OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This the amount of time, in seconds, that a node
        will wait after a circuit is successfully established
        before checking the traffic modems for loss of
        carrier. In this manner, it is only after this stabilization
        period that the node would send a terminate message
        back to the NMS in the event that the circuit failed due to
        carrier loss. Reducing this amount of time would cause
        a terminate message sooner if the circuit fails due
        to loss of carrier."
 ::= { sysParameters 7 }

sysParaMaxNode OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the max number of nodes which Midas can
        manage."
 ::= { sysParameters 8 }

sysISDNSetup OBJECT IDENTIFIER ::= { systemSetup 2 }

isdnsetupSNMPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Nominal power level for ISDN circuit"
 ::= { sysISDNSetup 1 }

isdnsetupSNTCPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Nominal ISDN power for C band SNT terminal "
 ::= { sysISDNSetup 2 }

isdnsetupSNTKuPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Nominal ISDN power for Ku band SNT terminal "
 ::= { sysISDNSetup 3 }

isdnsetupModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-write
    STATUS      mandatory

```

```

 ::= { sysISDNSetup 4 }

isdnssetupDataRate OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 5 }

isdnssetupFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 6 }

isdnssetupEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 7 }

isdnssetupReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 8 }

isdnssetupClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 9 }

isdnssetupRingTimeOut OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This call will ring waiting for call
        setup before disconnecting"
    ::= { sysISDNSetup 10 }

isdnssetupCallTimeout OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 11 }

isdnssetupGradeofService OBJECT-TYPE
    SYNTAX      TISDNGRADEOFSERVICE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysISDNSetup 12 }

isdnssetupInterfaceBuffer OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory

```

```

 ::= { sysISDNSetup 13 }

sysDataSetup OBJECT IDENTIFIER ::= { systemSetup 3 }

datasetupSNMPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Nominal SNM power of 10 KBPS BPSK ½
        data circuit "
    ::= { sysDataSetup 1 }

datasetupSNTCPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "C band SNT power of 10 KBPS BPSK ½
        data circuit"
    ::= { sysDataSetup 2 }

datasetupSNTKuPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Ku band SNT power of 10 KBPS BPSK ½
        data circuit"
    ::= { sysDataSetup 3 }

datasetupEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysDataSetup 4 }

datasetupReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysDataSetup 5 }

datasetupClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysDataSetup 6 }

datasetupRingTimeOut OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysDataSetup 7 }

datasetupCallTimeout OBJECT-TYPE
    SYNTAX      THALFLONGWORD

```

```

ACCESS      read-write
STATUS      mandatory
::= { sysDataSetup 8 }

datasetupInterfaceBuffer OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
::= { sysDataSetup 9 }

sysAFCSetsup OBJECT IDENTIFIER ::= { systemSetup 4 }

afcsetupAFCSetsupStartDelay OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "The number of minutes delayed between
    system startup and the beginning of node
    alignment "
::= { sysAFCSetsup 1 }

afcsetupNMSAlignInterval OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Time in seconds between outbound control
    channel frequency readings (and possible
    inbound frequency corrections)."
::= { sysAFCSetsup 2 }

afcsetupAlignHour OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Hour of the day when periodic node
    alignment is performed "
::= { sysAFCSetsup 3 }

afcsetupAlignMinute OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Minutes for node alignment start time "
::= { sysAFCSetsup 4 }

afcsetupNodePerDay OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Nodes per day for node alignment"
::= { sysAFCSetsup 5 }

```

```

afcsetupTimeBetweenNodes OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Minutes between alignment when multiple node
        alignments are performed in day "
    ::= { sysAFCSetup 6 }

afcsetupNodeAlignAlarmThreshold OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "A node offset of greater than this number of Hertz
        will cause an event to be logged "
    ::= { sysAFCSetup 7 }

afcsetupNodeAcquisitionTimeout OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Timeout in seconds for initial node acquisition
        (typical value is 30)"
    ::= { sysAFCSetup 8 }

sysPowerSetup OBJECT IDENTIFIER ::= { systemSetup 5 }

powersetupUplinkMargin OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "For remote circuits under power management,
        additional power (in dB) added to a circuit
        when uplink conditions are unknown "
    ::= { sysPowerSetup 1 }

powersetupDownlinkMargin OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "For remote circuits under power management,
        additional power (in dB) added to a circuit
        when downlink conditions are unknown "
    ::= { sysPowerSetup 2 }

powersetupAverageNumber OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "For remote circuit under power management,
        maximum number of Eb/No readings to average
        together to determine link conditions"

```

```

 ::= { sysPowerSetup 3 }

powersetupAverageTimeLimit OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "For remote circuit under power management,
        maximum time to keep an Eb/No readings"
 ::= { sysPowerSetup 4 }

powersetupUPCTimeInterval OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Number of seconds between EB/No readings
        of the outbound control channel, taken at the
        hub for uplink power control"
 ::= { sysPowerSetup 5 }

powersetupUPCThreshold OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Minimum change in power level (in dB) for
        adjusting output power of outbound control
        channel for uplink power control "
 ::= { sysPowerSetup 6 }

powersetupTFCThreshold OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Not used"
 ::= { sysPowerSetup 7 }

powersetupPowerBumping OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "If true, high priority circuits can bump low
        priority circuits when they would otherwise
        be blocked by site power limits"
 ::= { sysPowerSetup 8 }

sysVideoASetup OBJECT IDENTIFIER ::= { systemSetup 6 }

videoAsetupPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Broadcaster video default values"

```

```

 ::= { sysVideoASetup 1 }

videoAsetupModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        modulation standard to be used for the
        connection. Choices include: BPSK, QPSK,
        8PSK, 16QAM, etc."
    ::= { sysVideoASetup 2 }

videoAsetupDataRate OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
    ::= { sysVideoASetup 3 }

videoAsetupFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include:1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
    ::= { sysVideoASetup 4 }

videoAsetupEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
    ::= { sysVideoASetup 5 }

videoAsetupReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
    ::= { sysVideoASetup 6 }

videoAsetupClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION

```

```

        "To select the clock mode. Choices include:
        Terrestrial, Internal, Satellite, and
        External Reference."
 ::= { sysVideoASetup 7 }

videoAsetupRingTimeOut OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysVideoASetup 8 }

videoAsetupCallTimeout OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysVideoASetup 9 }

videoAsetupGradeofService OBJECT-TYPE
    SYNTAX      TVIDEOGRADEOFSERVICE
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysVideoASetup 10 }

videoAsetupInterfaceBuffer OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { sysVideoASetup 11 }

sysVideoBSetup OBJECT IDENTIFIER ::= { systemSetup 7 }

videoBsetupPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Returning channel video default values"
    ::= { sysVideoBSetup 1 }

videoBsetupModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        modulation standard to be used for the
        connection. Choices include: BPSK, QPSK,
        8PSK, 16QAM, etc."
    ::= { sysVideoBSetup 2 }

videoBsetupDataRate OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data

```

```

        rate that is valid for the traffic
        modem."
 ::= { sysVideoBSetup 3 }

videoBsetupFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include:1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
 ::= { sysVideoBSetup 4 }

videoBsetupEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
 ::= { sysVideoBSetup 5 }

videoBsetupReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { sysVideoBSetup 6 }

videoBsetupClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the clock mode. Choices include:
        Terrestrial, Internal, Satellite, and
        External Reference."
 ::= { sysVideoBSetup 7 }

videoBsetupRingTimeOut OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
 ::= { sysVideoBSetup 8 }

videoBsetupCallTimeout OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
 ::= { sysVideoBSetup 9 }

videoBsetupGradeofService OBJECT-TYPE
    SYNTAX      TVIDEOGRADEOFSERVICE
    ACCESS      read-write

```

```

STATUS      mandatory
 ::= { sysVideoBSetup 10 }

videoBsetupInterfaceBuffer OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
 ::= { sysVideoBSetup 11 }

sysRedundantController OBJECT IDENTIFIER ::= { systemSetup 8 }

redundantCtrlConfigAs OBJECT-TYPE
SYNTAX      TPRIMARYSTATUS
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The redundant configuration of controller.
    nbk: non redundant setup.
    primary: primary controller
    secondary: secondary controller"
 ::= { sysRedundantController 1 }

redundantCtrlOnlineOutboundTimeout OBJECT-TYPE
SYNTAX      INTEGER (10..100)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "The duration time for which the online controller
    tries to detect whether outbound signal exists. During
    this period of time if the online controller detects
    outbound signal, it means that there is controllers
    running on the outbound. The controller will
    enter mode_exception mode"
 ::= { sysRedundantController 2 }

redundantCtrlStandbyOutboundTimeout OBJECT-TYPE
SYNTAX      INTEGER (10..100)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "The period of time during which the standby
    controller continues to detect whether there
    is any online outbound. If the standby controller
    detects no online outbound for the period of
    time defined by the object, it will become
    Online to take over Midas operations."
 ::= { sysRedundantController 3 }

redundantCtrlAutoRestore OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "The object can be set to be true (2) only if
    controller configured as primary. The primary
    controller with the object turned on will switch
    to be online whenever it becomes standby. "

```

```

 ::= { sysRedundantController 4 }

redundantCtrlSwitchover OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To switch over between online and standby
        In order to perform a redundant switch operation,
        the online controller must be in 'normal' state for
        the mode and 'primary' for the primary status in
        the MIB"
    ::= { sysRedundantController 5 }

--
-- This branch provides the content
of the dongle key

sysKey OBJECT IDENTIFIER ::= { systemSetup 9 }

keyMaxNumberOfNodesEnabled OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Max number of operable nodes in Midas"
    ::= { sysKey 1 }

keyLinkSyncPowerManagement OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "whether the controller includes power management."
    ::= { sysKey 2 }

keyRedundantController OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Whether the controller includes redundant function. "
    ::= { sysKey 3 }

keyExpirationDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "the expiration date when midas controller
        runs."
    ::= { sysKey 4 }

keyStatus OBJECT-TYPE
    SYNTAX      TKEYSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION

```



```
ACCESS      not-accessible
STATUS      mandatory
INDEX { siteTypeIndex }
 ::= { siteTypeTable 1 }
```

```
SiteTypeTableEntry ::= SEQUENCE {
    siteTypeIndex
        THALFLONGWORD,
    siteTypeName
        OCTETSTRING30,
    siteTypeDescription
        OCTETSTRING30,
    siteTypeAntennaGain
        REAL,
    siteTypeOperations
        TOPERATION,
    siteType1Name
        OCTETSTRING30,
    siteType1TxPower
        REAL,
    siteType2Name
        OCTETSTRING30,
    siteType2TxPower
        REAL,
    siteType3Name
        OCTETSTRING30,
    siteType3TxPower
        REAL,
    siteType4Name
        OCTETSTRING30,
    siteType4TxPower
        REAL,
    siteType5Name
        OCTETSTRING30,
    siteType5TxPower
        REAL,
    siteType6Name
        OCTETSTRING30,
    siteType6TxPower
        REAL,
    siteType7Name
        OCTETSTRING30,
    siteType7TxPower
        REAL,
    siteType8Name
        OCTETSTRING30,
    siteType8TxPower
        REAL,
    siteType9Name
        OCTETSTRING30,
    siteType9TxPower
        REAL,
    siteType10Name
        OCTETSTRING30,
    siteType10TxPower
        REAL
}
```

```

siteTypeIndex OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { siteTypeTableEntry 1 }

siteTypeName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Site type name"
    ::= { siteTypeTableEntry 2 }

siteTypeDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To describe site type"
    ::= { siteTypeTableEntry 3 }

siteTypeAntennaGain OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To describe site type"
    ::= { siteTypeTableEntry 4 }

siteTypeOperations OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "not allow to delete now."
    ::= { siteTypeTableEntry 99 }

siteType1Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 1-th column of
        site type power matrix "
    ::= { siteTypeTableEntry 101 }

siteType1TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index points
        to and siteType1Name"
    ::= { siteTypeTableEntry 102 }

```

```

siteType2Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 2-th column of
        site type power matrix "
    ::= { siteTypeTableEntry 111 }

siteType2TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index
        points to and siteType2Name"
    ::= { siteTypeTableEntry 112 }

siteType3Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 3-th column of
        site type power matrix "
    ::= { siteTypeTableEntry 121 }

siteType3TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index
        points to and siteType3Name"
    ::= { siteTypeTableEntry 122 }

siteType4Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 4-th column of
        site type power matrix "
    ::= { siteTypeTableEntry 131 }

siteType4TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index points
        to and siteType4Name"
    ::= { siteTypeTableEntry 132 }

siteType5Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write

```

```

STATUS      mandatory
DESCRIPTION
    "the site type name of  the 5-th column of
    site type power matrix "
 ::= { siteTypeTableEntry 141 }

siteType5TxPower  OBJECT-TYPE
SYNTAX      REAL
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "power between site type which index points to
    and siteType5Name"
 ::= { siteTypeTableEntry 142 }

siteType6Name  OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "the site type name of  the 6-th column of
    site type power matrix "
 ::= { siteTypeTableEntry 151 }

siteType6TxPower  OBJECT-TYPE
SYNTAX      REAL
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "power between site type which index points to
    and siteType6Name"
 ::= { siteTypeTableEntry 152 }

siteType7Name  OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "the site type name of  the 7-th column of
    site type power matrix "
 ::= { siteTypeTableEntry 161 }

siteType7TxPower  OBJECT-TYPE
SYNTAX      REAL
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "power between site type which index points
    to and siteType7Name"
 ::= { siteTypeTableEntry 162 }

siteType8Name  OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS     read-write
STATUS     mandatory
DESCRIPTION
    "the site type name of  the 8-th column of

```

```

        site type power matrix "
 ::= { siteTypeTableEntry 171 }

siteType8TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index points
         to and siteType8Name"
 ::= { siteTypeTableEntry 172 }

siteType9Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 9-th column of
         site type power matrix "
 ::= { siteTypeTableEntry 181 }

siteType9TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index points
         to and siteType9Name"
 ::= { siteTypeTableEntry 182 }

siteType10Name OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type name of the 10-th column of
         site type power matrix "
 ::= { siteTypeTableEntry 191 }

siteType10TxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "power between site type which index points
         to and siteType10Name"
 ::= { siteTypeTableEntry 192 }

sites OBJECT IDENTIFIER ::= { siteConfigurations 2 }

--
--                               SiteType table must be inserted before site
table is inserted.
--
--                               There must always be a NMS site in the
database. User can't delete it.
--
--                               There is only one NMS site in the system. NMS
site does not have to have
--
--                               node.

```

```
--          a site cannot be deleted if there are some
nodes is residing in that site
--          Node table must be deleted before Site table
related to it is deleted
```

```
siteTable OBJECT-TYPE
    SYNTAX SEQUENCE OF SiteTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { sites 1 }
```

```
siteTableEntry OBJECT-TYPE
    SYNTAX      SiteTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { siteIndex }
    ::= { siteTable 1 }
```

```
SiteTableEntry ::= SEQUENCE {
    siteIndex
        THALFLONGWORD,
    siteName
        OCTETSTRING30,
    siteDescription
        OCTETSTRING30,
    siteTypeNameBelonged
        OCTETSTRING30,
    siteIsNMS
        BOOLEAN,
    siteContactName
        OCTETSTRING30,
    siteContactTitle
        OCTETSTRING30,
    sitePhone
        OCTETSTRING30,
    siteFax
        OCTETSTRING30,
    siteAddr1
        OCTETSTRING30,
    siteAddr2
        OCTETSTRING30,
    siteAddr3
        OCTETSTRING30,
    siteModemHPATxGain
        REAL,
    siteeccRefEbNo
        REAL,
    siteTrafficRefEbNo
        REAL,
    sitetxGainOffSet
        REAL,
    siteHPAPowerRated
        REAL,
    siteHPAPowerBackOff
        REAL,
    siteeccBurstPower
        REAL,
```

```

    siteNcsState
        TNCSSSTATE,
    siteOperationTag
        TOPERATION
}

siteIndex OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { siteTableEntry 1 }

siteName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "unique site name"
    ::= { siteTableEntry 2 }

siteDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The description of site"
    ::= { siteTableEntry 3 }

siteTypeNameBelonged OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the site type which the site belongs to."
    ::= { siteTableEntry 4 }

siteIsNMS OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "2: Midas controller runs in the site.
        1: Midas controller does not run in the site."
    ::= { siteTableEntry 5 }

siteContactName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to type a contact
        name for the customer."
    ::= { siteTableEntry 6 }

siteContactTitle OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write

```

```

STATUS      mandatory
DESCRIPTION
    "This field allows the operator to
    type the contact's title (Mr., Ms., Dr., etc.)."
 ::= { siteTableEntry 7 }

sitePhone OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to type
    the contact's phone number."
 ::= { siteTableEntry 8 }

siteFax OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to type the
    contact's fax number."
 ::= { siteTableEntry 9 }

siteAddr1 OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to type the first line
    of the address for the site."
 ::= { siteTableEntry 10 }

siteAddr2 OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to type the second line
    of the address for the site, if necessary."
 ::= { siteTableEntry 11 }

siteAddr3 OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to type the
    third line of the address for the site,
    if necessary."
 ::= { siteTableEntry 12 }

siteModemHPATxGain OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-write
STATUS      mandatory
DESCRIPTION

```

```

        "This field allows the operator to type the amount in
        dB of the modem-to-HPA TX gain.
        (Range: 0 to 60)
        Net gain from modem output to base of
        antenna."
 ::= { siteTableEntry 13 }

siteccRefEbNo OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "0.0 .. 20.0 dB
        This field allows the operator to type the reference
        Eb/N0 (in dB) for the control channel.
        (Range: 0 to 20).
        Nominal received Eb/No from control
        Channel outbound."
 ::= { siteTableEntry 14 }

siteTrafficRefEbNo OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "0.0 .. 20.0 dB
        This field allows the operator to type the
        reference Eb/N0 (in dB) for the traffic channel.
        Nominal Eb/No expected for traffic
        Channel."
 ::= { siteTableEntry 15 }

siteTxGainOffset OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "-10.0 .. +10.0 dB
        This field allows the operator to type the offset
        in dB for the TX gain. (Range: -10 to +10)
        Offset value added to ModemHPATxGain.
        Normally 0."
 ::= { siteTableEntry 16 }

siteHPAPowerRated OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "0 .. 1000 W
        This field allows the operator to type the rated
        power level in Watts. (Range: 0 to 1000).
        Rated output power of HPA."
 ::= { siteTableEntry 17 }

siteHPAPowerBackOff OBJECT-TYPE
    SYNTAX      REAL

```



```
nodeTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NodeTableEntry
    ACCESS not-accessible
    STATUS mandatory
    ::= { nodes 1 }
```

```
nodeTableEntry OBJECT-TYPE
    SYNTAX NodeTableEntry
    ACCESS not-accessible
    STATUS mandatory
    INDEX { nodeNodeID }
    ::= { nodeTable 1 }
```

```
NodeTableEntry ::= SEQUENCE {
    nodeNodeID
        TNODEID,
    nodeName
        OCTETSTRING30,
    nodeEnabled
        BOOLEAN,
    nodeStatus
        TNODEENABLESTATE,
    nodeExtFlag
        BOOLEAN,
    nodeControlChannelNo
        WORD,
    nodeMeasuredCCEbNo
        REAL,
    nodeDACEditLevel
        BYTE,
    nodeDACVersionLo
        BYTE,
    nodeDACVersionHi
        BYTE,
    nodeSiteName
        OCTETSTRING30,
    nodeNcsState
        TNCSSSTATE,
    nodeTotalCounts
        THALFLONGWORD,
    nodeReset
        BOOLEAN,
    nodeUplinkConvFactor
        LONGINT,
    nodeDownlinkConvFactor
        LONGINT,
    nodeOperationTag
        TOPERATION,
    nodeFaultModulator
        BOOLEAN,
    nodeFaultDemodulator
        BOOLEAN,
    nodeFaultMC
        BOOLEAN,
    nodeFaultInterface
        BOOLEAN,
```

nodeFaultBatteryClock
 BOOLEAN,
nodeFaultChannelOffline
 BOOLEAN,
nodeFaultHealthPollFail
 BOOLEAN,
nodeFaultStartupFail
 BOOLEAN,
nodeFaultHungCall
 BOOLEAN,
nodeFaultSave1
 BOOLEAN,
nodeFaultSave2
 BOOLEAN,
nodeFaultSave3
 BOOLEAN,
nodeWarning1
 BOOLEAN,
nodeWarning2
 BOOLEAN,
nodeWarning3
 BOOLEAN,
nodeWarning4
 BOOLEAN,
nodeHungCallChannel0
 BOOLEAN,
nodeHungCallChannel1
 BOOLEAN,
nodeHungCallChannel2
 BOOLEAN,
nodeHungCallChannel3
 BOOLEAN,
nodeHungCallChannel4
 BOOLEAN,
nodeHungCallChannel5
 BOOLEAN,
nodeHungCallChannel6
 BOOLEAN,
nodeHungCallChannel7
 BOOLEAN,
nodeHungCallChannel8
 BOOLEAN,
nodeHungCallChannel9
 BOOLEAN,
nodeHungCallChannel10
 BOOLEAN,
nodeHungCallChannel11
 BOOLEAN,
nodeHungCallChannel12
 BOOLEAN,
nodeHungCallChannel13
 BOOLEAN,
nodeHungCallChannel14
 BOOLEAN,
nodeHungCallChannel15
 BOOLEAN,
nodeHungCallChannel16

BOOLEAN,
nodeHungCallChannel17
BOOLEAN,
nodeHungCallChannel18
BOOLEAN,
nodeHungCallChannel19
BOOLEAN,
nodeHungCallChannel20
BOOLEAN,
nodeHungCallChannel21
BOOLEAN,
nodeHungCallChannel22
BOOLEAN,
nodeHungCallChannel23
BOOLEAN,
nodeHungCallChannel24
BOOLEAN,
nodeHungCallChannel25
BOOLEAN,
nodeHungCallChannel26
BOOLEAN,
nodeHungCallChannel27
BOOLEAN,
nodeHungCallChannel28
BOOLEAN,
nodeHungCallChannel29
BOOLEAN,
nodeHungCallChannel30
BOOLEAN,
nodeCallInProgressingChannel0
BOOLEAN,
nodeCallInProgressingChannel1
BOOLEAN,
nodeCallInProgressingChannel2
BOOLEAN,
nodeCallInProgressingChannel3
BOOLEAN,
nodeCallInProgressingChannel4
BOOLEAN,
nodeCallInProgressingChannel5
BOOLEAN,
nodeCallInProgressingChannel6
BOOLEAN,
nodeCallInProgressingChannel7
BOOLEAN,
nodeCallInProgressingChannel8
BOOLEAN,
nodeCallInProgressingChannel9
BOOLEAN,
nodeCallInProgressingChannel10
BOOLEAN,
nodeCallInProgressingChannel11
BOOLEAN,
nodeCallInProgressingChannel12
BOOLEAN,
nodeCallInProgressingChannel13
BOOLEAN,

nodeCallInProgressingChannel14
 BOOLEAN,
nodeCallInProgressingChannel15
 BOOLEAN,
nodeCallInProgressingChannel16
 BOOLEAN,
nodeCallInProgressingChannel17
 BOOLEAN,
nodeCallInProgressingChannel18
 BOOLEAN,
nodeCallInProgressingChannel19
 BOOLEAN,
nodeCallInProgressingChannel20
 BOOLEAN,
nodeCallInProgressingChannel21
 BOOLEAN,
nodeCallInProgressingChannel22
 BOOLEAN,
nodeCallInProgressingChannel23
 BOOLEAN,
nodeCallInProgressingChannel24
 BOOLEAN,
nodeCallInProgressingChannel25
 BOOLEAN,
nodeCallInProgressingChannel26
 BOOLEAN,
nodeCallInProgressingChannel27
 BOOLEAN,
nodeCallInProgressingChannel28
 BOOLEAN,
nodeCallInProgressingChannel29
 BOOLEAN,
nodeCallInProgressingChannel30
 BOOLEAN,
nodeRetryCounts0
 THALFLONGWORD,
nodeRetryCounts1
 THALFLONGWORD,
nodeRetryCounts2
 THALFLONGWORD,
nodeRetryCounts3
 THALFLONGWORD,
nodeRetryCounts4
 THALFLONGWORD,
nodeRetryCounts5
 THALFLONGWORD,
nodeRetryCounts6
 THALFLONGWORD,
nodeRetryCounts7
 THALFLONGWORD,
nodeRetryCounts8
 THALFLONGWORD,
nodeRetryCounts9
 THALFLONGWORD,
nodeRetryCounts10
 THALFLONGWORD,
nodeRetryCounts11

THALFLONGWORD,
nodeRetryCounts12
THALFLONGWORD,
nodeRetryCounts13
THALFLONGWORD,
nodeRetryCounts14
THALFLONGWORD,
nodeRetryCounts15
THALFLONGWORD,
nodeRetryCounts16
THALFLONGWORD,
nodeRetryCounts17
THALFLONGWORD,
nodeRetryCounts18
THALFLONGWORD,
nodeRetryCounts19
THALFLONGWORD,
nodeRetryCounts20
THALFLONGWORD,
nodeRetryCounts21
THALFLONGWORD,
nodeRetryCounts22
THALFLONGWORD,
nodeRetryCounts23
THALFLONGWORD,
nodeRetryCounts24
THALFLONGWORD,
nodeRetryCounts25
THALFLONGWORD,
nodeRetryCounts26
THALFLONGWORD,
nodeRetryCounts27
THALFLONGWORD,
nodeRetryCounts28
THALFLONGWORD,
nodeRetryCounts29
THALFLONGWORD,
nodeRetryCounts30
THALFLONGWORD,
nodeRetryCounts31
THALFLONGWORD,
nodeRetryCounts32
THALFLONGWORD,
nodeRetryCounts33
THALFLONGWORD,
nodeRetryCounts34
THALFLONGWORD,
nodeRetryCounts35
THALFLONGWORD,
nodeRetryCounts36
THALFLONGWORD,
nodeRetryCounts37
THALFLONGWORD,
nodeRetryCounts38
THALFLONGWORD,
nodeRetryCounts39
THALFLONGWORD,

```

        nodeRetryCounts40
            THALFLONGWORD
    }

nodeNodeID OBJECT-TYPE
    SYNTAX      TNODEID
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to type the node
         ID number."
    ::= { nodeTableEntry 1 }

nodeName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to type the name of
         the node."
    ::= { nodeTableEntry 2 }

nodeEnabled OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To bring node active A node is created in the
         disabled state, and does not become active
         until it is enabled. Once enabled, the node
         will be polled and brought online. On an
         enabled (active) node, channels can set up
         and respond to calls.
         Enabled Node and Channel can't be disabled if
         there is an active call related to them
         Can't enable/disable a nodet when the
         controller mode is not normal"
    ::= { nodeTableEntry 3 }

nodeStatus OBJECT-TYPE
    SYNTAX      TNODEENABLESTATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The status of node"
    ::= { nodeTableEntry 4 }

--
--          Make sure the modem type is correct for
internal and external such as
--          SNM1010 can't be an external node. For
data/internal, there are IDC-150D,
--          SNM1010. For data/external, there are 100, 150,
300, 6000, 6000, 9000, 2200,
--          2020, 300L. For ISNN/internal, there are IDC-
150, SNM-1020. For
--          ISDN/external. There are IDC-150, 300, 2200.
For video/internal, there are

```

-- none and video/external is the same as
data/external

nodeExtFlag OBJECT-TYPE
SYNTAX BOOLEAN
ACCESS read-write
STATUS mandatory
DESCRIPTION
"2: external, 1: internal
INTERNAL
This allows the operator to select INTERNAL,
to set the modem as an internal modem. When a
node is set to internal, only one channel can appear
in the Data Channels list. This single channel represents
the modem that switches to traffic mode from control
channel mode when a call is established.
EXTERNAL
This allows the operator to select EXTERNAL, to
set the modem as an external modem.
Can't change a node from external to internal or
vice versa when the node is enable
Internal node can't be video channel"
 ::= { nodeTableEntry 5 }

nodeControlChannelNo OBJECT-TYPE
SYNTAX WORD
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Select the control channel to which the node is
assigned. Each node is allocated to one control
channel. Many nodes can belong to the same control
channel. However, the more nodes on the same control
channel, the greater the degradation of the control
channel performance."
 ::= { nodeTableEntry 6 }

nodeMeasuredCCEbNo OBJECT-TYPE
SYNTAX REAL
ACCESS read-only
STATUS mandatory
 ::= { nodeTableEntry 7 }

nodeDACEditLevel OBJECT-TYPE
SYNTAX BYTE
ACCESS read-only
STATUS mandatory
 ::= { nodeTableEntry 8 }

nodeDACVersionLo OBJECT-TYPE
SYNTAX BYTE
ACCESS read-only
STATUS mandatory
 ::= { nodeTableEntry 9 }

nodeDACVersionHi OBJECT-TYPE
SYNTAX BYTE

```

ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 10 }

nodeSiteName OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This displays the site that the current node is
    associated with."
 ::= { nodeTableEntry 11 }

nodeNcsState OBJECT-TYPE
SYNTAX      TNCSSTATE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "the status of node"
 ::= { nodeTableEntry 12 }

nodeTotalCounts OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 13 }

nodeReset OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
 ::= { nodeTableEntry 14 }

nodeUplinkConvFactor OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
 ::= { nodeTableEntry 15 }

nodeDownlinkConvFactor OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
 ::= { nodeTableEntry 16 }

nodeOperationTag OBJECT-TYPE
SYNTAX      TOPERATION
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "2: delete node
    Before a node is deleted, all Channels under the
    node should be deleted."
 ::= { nodeTableEntry 99 }

nodeFaultModulator OBJECT-TYPE
SYNTAX      BOOLEAN

```

```

ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 101 }

nodeFaultDemodulator OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 102 }

nodeFaultMC OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 103 }

nodeFaultInterface OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 104 }

nodeFaultBatteryClock OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 105 }

nodeFaultChannelOffline OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 106 }

nodeFaultHealthPollFail OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 107 }

nodeFaultStartupFail OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 108 }

nodeFaultHungCall OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { nodeTableEntry 109 }

nodeFaultSave1 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
DESCRIPTION

```

```

        "Reserved for future"
 ::= { nodeTableEntry 110 }

nodeFaultSave2 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 111 }

nodeFaultSave3 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 112 }

nodeWarning1 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 201 }

nodeWarning2 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 202 }

nodeWarning3 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 203 }

nodeWarning4 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Reserved for future"
 ::= { nodeTableEntry 204 }

nodeHungCallChannel0 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { nodeTableEntry 301 }

```

```

nodeHungCallChannel1 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 302 }

nodeHungCallChannel2 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 303 }

nodeHungCallChannel3 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 304 }

nodeHungCallChannel4 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 305 }

nodeHungCallChannel5 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 306 }

nodeHungCallChannel6 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 307 }

nodeHungCallChannel7 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 308 }

nodeHungCallChannel8 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 309 }

nodeHungCallChannel9 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 310 }

nodeHungCallChannel10 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only

```

```

STATUS      mandatory
 ::= { nodeTableEntry 311 }

nodeHungCallChannel11 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 312 }

nodeHungCallChannel12 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 313 }

nodeHungCallChannel13 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 314 }

nodeHungCallChannel14 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 315 }

nodeHungCallChannel15 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 316 }

nodeHungCallChannel16 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 317 }

nodeHungCallChannel17 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 318 }

nodeHungCallChannel18 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 319 }

nodeHungCallChannel19 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 320 }

```

```

nodeHungCallChannel20 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 321 }

nodeHungCallChannel21 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 322 }

nodeHungCallChannel22 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 323 }

nodeHungCallChannel23 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 324 }

nodeHungCallChannel24 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 325 }

nodeHungCallChannel25 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 326 }

nodeHungCallChannel26 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 327 }

nodeHungCallChannel27 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 328 }

nodeHungCallChannel28 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 329 }

nodeHungCallChannel29 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only

```

```

STATUS      mandatory
 ::= { nodeTableEntry 330 }

nodeHungCallChannel30 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 331 }

nodeCallInProgressingChannel0 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 401 }

nodeCallInProgressingChannel1 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 402 }

nodeCallInProgressingChannel2 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 403 }

nodeCallInProgressingChannel3 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 404 }

nodeCallInProgressingChannel4 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 405 }

nodeCallInProgressingChannel5 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 406 }

nodeCallInProgressingChannel6 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 407 }

nodeCallInProgressingChannel7 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 408 }

```

```

nodeCallInProgressingChannel8 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 409 }

nodeCallInProgressingChannel9 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 410 }

nodeCallInProgressingChannel10 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 411 }

nodeCallInProgressingChannel11 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 412 }

nodeCallInProgressingChannel12 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 413 }

nodeCallInProgressingChannel13 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 414 }

nodeCallInProgressingChannel14 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 415 }

nodeCallInProgressingChannel15 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 416 }

nodeCallInProgressingChannel16 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 417 }

nodeCallInProgressingChannel17 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only

```

```

STATUS      mandatory
 ::= { nodeTableEntry 418 }

nodeCallInProgressingChannel18 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 419 }

nodeCallInProgressingChannel19 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 420 }

nodeCallInProgressingChannel20 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 421 }

nodeCallInProgressingChannel21 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 422 }

nodeCallInProgressingChannel22 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 423 }

nodeCallInProgressingChannel23 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 424 }

nodeCallInProgressingChannel24 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 425 }

nodeCallInProgressingChannel25 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 426 }

nodeCallInProgressingChannel26 OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { nodeTableEntry 427 }

```

```

nodeCallInProgressingChannel27 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 428 }

nodeCallInProgressingChannel28 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 429 }

nodeCallInProgressingChannel29 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 430 }

nodeCallInProgressingChannel30 OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 431 }

nodeRetryCounts0 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 500 }

nodeRetryCounts1 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 501 }

nodeRetryCounts2 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 502 }

nodeRetryCounts3 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 503 }

nodeRetryCounts4 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 504 }

nodeRetryCounts5 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only

```

```

        STATUS      mandatory
        ::= { nodeTableEntry 505 }

nodeRetryCounts6 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 506 }

nodeRetryCounts7 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 507 }

nodeRetryCounts8 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 508 }

nodeRetryCounts9 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 509 }

nodeRetryCounts10 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 510 }

nodeRetryCounts11 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 511 }

nodeRetryCounts12 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 512 }

nodeRetryCounts13 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 513 }

nodeRetryCounts14 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 514 }

```

```

nodeRetryCounts15 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 515 }

nodeRetryCounts16 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 516 }

nodeRetryCounts17 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 517 }

nodeRetryCounts18 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 518 }

nodeRetryCounts19 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 519 }

nodeRetryCounts20 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 520 }

nodeRetryCounts21 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 521 }

nodeRetryCounts22 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 522 }

nodeRetryCounts23 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 523 }

nodeRetryCounts24 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only

```

```

        STATUS      mandatory
        ::= { nodeTableEntry 524 }

nodeRetryCounts25 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 525 }

nodeRetryCounts26 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 526 }

nodeRetryCounts27 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 527 }

nodeRetryCounts28 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 528 }

nodeRetryCounts29 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 529 }

nodeRetryCounts30 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 530 }

nodeRetryCounts31 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 531 }

nodeRetryCounts32 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 532 }

nodeRetryCounts33 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 533 }

```

```

nodeRetryCounts34 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 534 }

nodeRetryCounts35 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 535 }

nodeRetryCounts36 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 536 }

nodeRetryCounts37 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 537 }

nodeRetryCounts38 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 538 }

nodeRetryCounts39 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 539 }

nodeRetryCounts40 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { nodeTableEntry 540 }

channels OBJECT IDENTIFIER ::= { siteConfigurations 4 }

--
-- Node table must be inserted before Channel
table

channelTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ChannelTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { channels 1 }

channelTableEntry OBJECT-TYPE
    SYNTAX      ChannelTableEntry
    ACCESS      not-accessible
    STATUS      mandatory

```

```
INDEX { channelNodeID, channelChannelID }
 ::= { channelTable 1 }
```

```
ChannelTableEntry ::= SEQUENCE {
    channelNodeID
        TNODEID,
    channelChannelID
        TCHANNELID,
    channelEnabled
        BOOLEAN,
    channelStatus
        TCHANNELSTATUS,
    channelDescription
        OCTETSTRING30,
    channelType
        TCHANNELTYPE,
    channelDefaultHGrpDir
        OCTETSTRING30,
    channelMaxBandwidth
        LONGINT,
    channelAccountID
        OCTETSTRING15,
    channelModemType
        TMODEMTYPE,
    channelHighPowerOption
        BOOLEAN,
    channelReedSolomon
        BOOLEAN,
    channelBlocking
        BOOLEAN,
    channelBumping
        BOOLEAN,
    channelNcsState
        TCHANNELNCSSTATE,
    channelCallInProgressing
        BOOLEAN,
    channelOperationTag
        TOPERATION,
    channelAUPCParamsMaxPowerDelta
        REAL,
    channelAUPCParamsMinPowerDelta
        REAL,
    channelAUPCParamsTargetEbNo
        REAL,
    channelAUPCParamsTrackingRate
        REAL,
    channelAUPCParamsLocalAction
        TCARRIERLOSSACTION,
    channelAUPCParamsRemoteAction
        TCARRIERLOSSACTION,
    channelAUPCParamsCarrierLossTime
        BYTE,
    channelAUPCParamsEnable
        BOOLEAN,
    channelFaultModulator
        BOOLEAN,
    channelFaultDemodulator
```

```

        BOOLEAN,
channelFaultMC
        BOOLEAN,
channelFaultInterface
        BOOLEAN,
channelFaultBatteryClock
        BOOLEAN,
channelFaultChannelOffline
        BOOLEAN,
channelFaultSave1
        BOOLEAN,
channelFaultSave2
        BOOLEAN,
channelWarning1
        BOOLEAN,
channelWarning2
        BOOLEAN,
channelWarning3
        BOOLEAN,
channelWarning4
        BOOLEAN
}

channelNodeID OBJECT-TYPE
    SYNTAX      TNODEID
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the ID assigned to the node."
    ::= { channelTableEntry 1 }

channelChannelID OBJECT-TYPE
    SYNTAX      TCHANNELID
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the channel number assigned to the node."
    ::= { channelTableEntry 2 }

channelEnabled OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "2: to bring channel online.
        Enabled Channel can't be disabled if there is an
        active call related to them
        Can't enable/disable a channel when the
        controller mode is not normal"
    ::= { channelTableEntry 3 }

channelStatus OBJECT-TYPE
    SYNTAX      TCHANNELSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "the status of channel"

```

```

 ::= { channelTableEntry 4 }

channelDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to type a description
        of the channel."
 ::= { channelTableEntry 5 }

channelType OBJECT-TYPE
    SYNTAX      TCHANNELTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the channel type.
        Choose from: Data, ISDN, or Video."
 ::= { channelTableEntry 6 }

channelDefaultHGpDir OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the default gateway
        (a directory number associated with a hunt group)."
 ::= { channelTableEntry 7 }

channelMaxBandwidth OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
 ::= { channelTableEntry 8 }

channelAccountID OBJECT-TYPE
    SYNTAX      OCTETSTRING15
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to type the pool/account ID
        number (customer account) to use for the selected channel.
        All links originating from this channel will allocate
        bandwidth in the pool specified here."
 ::= { channelTableEntry 9 }

channelModemType OBJECT-TYPE
    SYNTAX      TMODEMTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the modem type."
 ::= { channelTableEntry 10 }

channelHighPowerOption OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write

```

```

STATUS      mandatory
DESCRIPTION
    "To enables/disables the High Power option
    for the selected modem type. If a modem
    does not have the high power option, there
    will be no change to the power output that
    was previously selected."
 ::= { channelTableEntry 11 }

channelReedSolomon OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "To enable and disable the Reed-Solomon
    function for the selected individual channel."
 ::= { channelTableEntry 12 }

channelBlocking OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
 ::= { channelTableEntry 13 }

channelBumping OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
 ::= { channelTableEntry 14 }

channelNcsState OBJECT-TYPE
SYNTAX      TCHANNELNCSSTATE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "the status of channel"
 ::= { channelTableEntry 15 }

channelCallInProgressing OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "There is call in progressing in the channel."
 ::= { channelTableEntry 16 }

channelOperationTag OBJECT-TYPE
SYNTAX      TOPERATION
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "2: delete channel
    Before a channel is deleted, all of table rows which
    are associated with the channel such as circuit table,
    channel hunt group table, directory table, and
    multiPointMember table have to be deleted."
 ::= { channelTableEntry 99 }

```

```

channelAUPCParamsMaxPowerDelta OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "+3.0 .. +20.0 dB default: +10 dB
        If AUPC is used in the circuit, this
        is the maximum increase in the power
        from nominal during the call"
    ::= { channelTableEntry 101 }

channelAUPCParamsMinPowerDelta OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "-3.0 .. -20 dB default: -10 dB
        If AUPC is used, the maximum decrease
        from nominal power"
    ::= { channelTableEntry 102 }

channelAUPCParamsTargetEbNo OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "+3.2 .. +16.0 dB default: 12 dB"
    ::= { channelTableEntry 103 }

channelAUPCParamsTrackingRate OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "0.5 .. 6.0 dB/min default: 2.0 db/min
        Maximum rate of change of power
        during an AUPC call."
    ::= { channelTableEntry 104 }

channelAUPCParamsLocalAction OBJECT-TYPE
    SYNTAX      TCARRIERLOSSACTION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "default:max
        Action to take at channel A if channel A
        loses carrier lock."
    ::= { channelTableEntry 105 }

channelAUPCParamsRemoteAction OBJECT-TYPE
    SYNTAX      TCARRIERLOSSACTION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "default: max
        Action to take at channel A if channel B

```

```

        reported loss of carrier lock."
 ::= { channelTableEntry 106 }

channelAUPCParamsCarrierLossTime OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "0 .. 256 sec default: 0
        Number of seconds of no lock before
        carrier loss is declared."
 ::= { channelTableEntry 107 }

channelAUPCParamsEnable OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "True if the channel has AUPC capability."
 ::= { channelTableEntry 108 }

channelFaultModulator OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 201 }

channelFaultDemodulator OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 202 }

channelFaultMC OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 203 }

channelFaultInterface OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 204 }

channelFaultBatteryClock OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 205 }

channelFaultChannelOffline OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
 ::= { channelTableEntry 206 }

```

```

channelFaultSave1 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 207 }

channelFaultSave2 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 208 }

channelWarning1 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 301 }

channelWarning2 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 302 }

channelWarning3 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 303 }

channelWarning4 OBJECT-TYPE
    SYNTAX          BOOLEAN
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION
        "Reserved for future"
    ::= { channelTableEntry 304 }

resources OBJECT IDENTIFIER ::= { midas 3 }

controlchannels OBJECT IDENTIFIER ::= { resources 1 }

controlChannelTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ControlChannelTableEntry
    ACCESS          not-accessible
    STATUS          mandatory
    ::= { controlchannels 1 }

```

```

controlChannelTableEntry OBJECT-TYPE
    SYNTAX      ControlChannelTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { controlChannelNo }
    ::= { controlChannelTable 1 }

```

```

ControlChannelTableEntry ::= SEQUENCE {
    controlChannelNo
        WORD,
    ctrlChanDescription
        OCTETSTRING30,
    ctrlChanType
        TCONTROLCHANNELTYPE,
    ctrlChanPollType
        TPOLLTYPE,
    ctrlChanAliveCheckTime
        WORD,
    ctrlChanAliveCheckCount
        WORD,
    ctrlChanOfflineCheckTime
        WORD,
    ctrlChanOfflineCheckCount
        WORD,
    ctrlChanNodeTimeOut
        WORD,
    ctrlChanTimeoutsBeforeWarning
        WORD,
    ctrlChanTimeoutsBeforeOffline
        WORD,
    ctrlChanSlotTime
        WORD,
    ctrlChanRandomMin
        WORD,
    ctrlChanRandomMax
        WORD,
    ctrlChanNoAckCount
        WORD,
    ctrlChanMaxRetries
        WORD,
    ctrlChanCCTimeOut
        WORD,
    ctrlChanPowerLevel
        REAL,
    ctrlChanModulation
        TMODULATION,
    ctrlChanFECRate
        TFECRATE,
    ctrlChanDataRate
        LONGINT,
    ctrlChanReferenceEbNo
        REAL,
    ctrlChanUplinkPowerControl
        BOOLEAN,
    ctrlChanAcquisitionRange
        LONGINT,

```

```

ctrlChanDeltaPowerLimit
    REAL,
ctrlChanTransponderName
    OCTETSTRING30,
ctrlChanEnabled
    BOOLEAN,
ctrlChanCCStatus
    TCONTROLCHANNELSTATUS,
ctrlChanOperationTag
    TOPERATION,
ctrlChanModemChannelA
    TMODEMCHANNEL,
ctrlChanModemChannelB
    TMODEMCHANNEL,
ctrlChanOutboundBandManID
    WORD,
ctrlChanOutboundFirstAllocUnit
    THALFLONGWORD,
ctrlChanOutboundNumAllocUnits
    THALFLONGWORD,
ctrlChanOutboundTxCenterIF
    THALFLONGWORD,
ctrlChanInboundBandManID
    WORD,
ctrlChanInboundFirstAllocUnit
    THALFLONGWORD,
ctrlChanInboundNumAllocUnits
    THALFLONGWORD,
ctrlChanInboundTxCenterIF
    THALFLONGWORD,
ctrlChanOutboundLBandFreq
    THALFLONGWORD,
ctrlChanInboundLBandFreq
    THALFLONGWORD
}

controlChannelNo OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the number of the selected Control Channel."
    ::= { controlChannelTableEntry 1 }

ctrlChanDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the description text of the selected Control Channel."
    ::= { controlChannelTableEntry 2 }

ctrlChanType OBJECT-TYPE
    SYNTAX      TCONTROLCHANNELTYPE
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 3 }

```

```

ctrlChanPollType OBJECT-TYPE
    SYNTAX      TPOLLTYPE
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 4 }

ctrlChanAliveCheckTime OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { controlChannelTableEntry 5 }

ctrlChanAliveCheckCount OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 6 }

ctrlChanOfflineCheckTime OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 7 }

ctrlChanOfflineCheckCount OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 8 }

ctrlChanNodeTimeOut OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 9 }

ctrlChanTimeoutsBeforeWarning OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 10 }

ctrlChanTimeoutsBeforeOffline OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 11 }

ctrlChanSlotTime OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelTableEntry 12 }

ctrlChanRandomMin OBJECT-TYPE
    SYNTAX      WORD

```

```

ACCESS      read-write
STATUS      mandatory
 ::= { controlChannelTableEntry 13 }

ctrlChanRandomMax OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
 ::= { controlChannelTableEntry 14 }

ctrlChanNoAckCount OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelTableEntry 15 }

ctrlChanMaxRetries OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelTableEntry 16 }

ctrlChanCCTimeOut OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelTableEntry 17 }

ctrlChanPowerLevel OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-write
STATUS      mandatory
 ::= { controlChannelTableEntry 18 }

ctrlChanModulation OBJECT-TYPE
SYNTAX      TMODULATION
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the
    modulation standard to be used for the
    connection. Choices include: BPSK, QPSK,
    8PSK, 16QAM, etc."
 ::= { controlChannelTableEntry 19 }

ctrlChanFECRate OBJECT-TYPE
SYNTAX      TFECRATE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "To select the forward error correction
    rate (FEC) to be used for the channel.
    Choices include: 1/2, 2/3, 3/4, 5/6, 7/8,
    etc."
 ::= { controlChannelTableEntry 20 }

ctrlChanDataRate OBJECT-TYPE

```

SYNTAX LONGINT
ACCESS read-only
STATUS mandatory
DESCRIPTION
 >Select the data rate from the list of
 available choices, or enter any data
 rate that is valid for the traffic
 modem."
 ::= { controlChannelTableEntry 21 }

ctrlChanReferenceEbNo OBJECT-TYPE
SYNTAX REAL
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "This is the stored value at which the
 received Outbound Control Channel Eb/No
 will be maintained under clear sky conditions,
 as measured at the LinkSync modem."
 ::= { controlChannelTableEntry 22 }

ctrlChanUplinkPowerControl OBJECT-TYPE
SYNTAX BOOLEAN
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "This allows the operator to enable/disable
 the Uplink Power Control option.
 Uplink Power Control (UPC), a standard
 feature of the MIDAS system, is the method
 used to establish a network-wide reference
 for power. Specifically, UPC is used to
 ensure that the Outbound Control Channel
 carrier is downlinked from the satellite at
 a constant power level. This reference
 allows each of the nodes in the network
 to determine and report (via regular
 polling from the NMS) if, and to what
 degree, downlink degradation of a node
 (and its associated traffic channels)
 may be occurring due to fade, whether
 from rain or other non-atmospheric
 conditions."
 ::= { controlChannelTableEntry 23 }

ctrlChanAcquisitionRange OBJECT-TYPE
SYNTAX LONGINT
ACCESS read-write
STATUS mandatory
DESCRIPTION
 "Maximum frequency adjustment allowed
 of the nodes during node acquisition.
 This is the Inbound Acquisition Range in kHz
 (Range: 0 to 35).
 The Acquisition Range is a value (in kilohertz)
 that is used to provide additional guard band
 for the bursting Inbound Control Channel. This

value is used to complement the operation of Automatic Frequency Control (AFC), and allows nodes a wider sweep range when performing transmit acquisition by the nodes in systems where satellite frequency translation error may exceed the limited demodulator sweep range of the SNM-1001 Control Channel modem. This value is stored in the database, and used by the bandwidth manager task to assign frequencies for traffic channels, with respect to the additional guard band. This value is also sent to the nodes when they are commanded to become enabled, where it is stored and then used by the Node Control modem to limit the transmit sweep range when performing transmit acquisition. This allows the node to have a controlled sweep range to compensate for excessive frequency offsets in the network, while preventing interference with adjacent carriers."

```
::= { controlChannelTableEntry 24 }
```

```
ctrlChanDeltaPowerLimit OBJECT-TYPE
```

```
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
```

```
DESCRIPTION
```

```
"Not used"
```

```
::= { controlChannelTableEntry 25 }
```

```
ctrlChanTransponderName OBJECT-TYPE
```

```
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
```

```
DESCRIPTION
```

```
"the transponder which the control channel run."
```

```
::= { controlChannelTableEntry 26 }
```

```
ctrlChanEnabled OBJECT-TYPE
```

```
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
```

```
DESCRIPTION
```

```
"2: To bring control channel online"
```

```
::= { controlChannelTableEntry 27 }
```

```
ctrlChanCCStatus OBJECT-TYPE
```

```
SYNTAX      TCONTROLCHANNELSTATUS
ACCESS      read-only
STATUS      mandatory
```

```
DESCRIPTION
```

```
"Not used"
```

```
::= { controlChannelTableEntry 28 }
```

```
ctrlChanOperationTag OBJECT-TYPE
```

```
SYNTAX      TOPERATION
ACCESS      read-write
STATUS      mandatory
```

```
DESCRIPTION
```



```

SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the L-band Frequency of
    the Inbound Control Channel, for the burst
    transmissions sent to the NMS from the various
    network elements. This field allows the operator
    to define L-band frequencies on which
    the various network elements will transmit to the
    NMS. Accordingly, the nodes' Inbound Center
    frequencies (via the User Port) are set to receive
    the Outbound Control Channel transmissions. "
 ::= { controlChannelTableEntry 402 }

```

```

controlChannelStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ControlChannelStatusTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { controlchannels 2 }

```

```

controlChannelStatusTableEntry OBJECT-TYPE
    SYNTAX      ControlChannelStatusTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { ctrlChanStatsChannel }
    ::= { controlChannelStatusTable 1 }

```

```

ControlChannelStatusTableEntry ::= SEQUENCE {
    ctrlChanStatsChannel
        BYTE,
    ctrlChanStatsMonitorEBNo
        REAL,
    ctrlChanStatsPollCount
        THALFLONGWORD,
    ctrlChanStatsRxMessages
        THALFLONGWORD,
    ctrlChanStatsTxMessages
        THALFLONGWORD,
    ctrlChanStatsRxFailures
        THALFLONGWORD,
    ctrlChanStatsTxFailures
        THALFLONGWORD,
    ctrlChanStatsRetryRate
        REAL,
    ctrlChanStatsPollTime
        WORD,
    ctrlChanStatsKeepAliveRate
        REAL,
    ctrlChanStatsPollType
        TPOLLTYPE,
    ctrlChanStatsAdaptive
        BOOLEAN,
    ctrlChanStatsTransponderName
        OCTETSTRING30,
    ctrlChanStatsMaxAlohaRetries
        BYTE,

```

```

ctrlChanStatsNode
    THALFLONGWORD,
ctrlChanStatsReset
    BOOLEAN,
ctrlChanStatsRetry0
    THALFLONGWORD,
ctrlChanStatsRetry1
    THALFLONGWORD,
ctrlChanStatsRetry2
    THALFLONGWORD,
ctrlChanStatsRetry3
    THALFLONGWORD,
ctrlChanStatsRetry4
    THALFLONGWORD,
ctrlChanStatsRetry5
    THALFLONGWORD,
ctrlChanStatsRetry6
    THALFLONGWORD,
ctrlChanStatsRetry7
    THALFLONGWORD,
ctrlChanStatsRetry8
    THALFLONGWORD,
ctrlChanStatsRetry9
    THALFLONGWORD,
ctrlChanStatsRetry10
    THALFLONGWORD,
ctrlChanStatsRetry11
    THALFLONGWORD,
ctrlChanStatsRetry12
    THALFLONGWORD,
ctrlChanStatsRetry13
    THALFLONGWORD,
ctrlChanStatsRetry14
    THALFLONGWORD,
ctrlChanStatsRetry15
    THALFLONGWORD,
ctrlChanStatsRetry16
    THALFLONGWORD,
ctrlChanStatsRetry17
    THALFLONGWORD,
ctrlChanStatsRetry18
    THALFLONGWORD,
ctrlChanStatsRetry19
    THALFLONGWORD,
ctrlChanStatsRetry20
    THALFLONGWORD
}

ctrlChanStatsChannel OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 1 }

ctrlChanStatsMonitorEBNo OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only

```

```

STATUS      mandatory
 ::= { controlChannelStatusTableEntry 2 }

ctrlChanStatsPollCount OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 3 }

ctrlChanStatsRxMessages OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 4 }

ctrlChanStatsTxMessages OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 5 }

ctrlChanStatsRxFailures OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 6 }

ctrlChanStatsTxFailures OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 7 }

ctrlChanStatsRetryRate OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 8 }

ctrlChanStatsPollTime OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 9 }

ctrlChanStatsKeepAliveRate OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 10 }

ctrlChanStatsPollType OBJECT-TYPE
SYNTAX      TPOLLTYPE
ACCESS      read-only
STATUS      mandatory
 ::= { controlChannelStatusTableEntry 11 }

```

```

ctrlChanStatsAdaptive OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 12 }

ctrlChanStatsTransponderName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 13 }

ctrlChanStatsMaxAlohaRetries OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 14 }

ctrlChanStatsNode OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 15 }

ctrlChanStatsReset OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 16 }

ctrlChanStatsRetry0 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 100 }

ctrlChanStatsRetry1 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 101 }

ctrlChanStatsRetry2 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 102 }

ctrlChanStatsRetry3 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 103 }

ctrlChanStatsRetry4 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only

```

```

        STATUS      mandatory
        ::= { controlChannelStatusTableEntry 104 }

ctrlChanStatsRetry5 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 105 }

ctrlChanStatsRetry6 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 106 }

ctrlChanStatsRetry7 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 107 }

ctrlChanStatsRetry8 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 108 }

ctrlChanStatsRetry9 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 109 }

ctrlChanStatsRetry10 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 110 }

ctrlChanStatsRetry11 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 111 }

ctrlChanStatsRetry12 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 112 }

ctrlChanStatsRetry13 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 113 }

```

```

ctrlChanStatsRetry14 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 114 }

ctrlChanStatsRetry15 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 115 }

ctrlChanStatsRetry16 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 116 }

ctrlChanStatsRetry17 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 117 }

ctrlChanStatsRetry18 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 118 }

ctrlChanStatsRetry19 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 119 }

ctrlChanStatsRetry20 OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { controlChannelStatusTableEntry 120 }

pools OBJECT IDENTIFIER ::= { resources 2 }

--
-- Band table must be deleted before Pool table
-- related to it is deleted

poolTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PoolTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { pools 1 }

poolTableEntry OBJECT-TYPE
    SYNTAX      PoolTableEntry
    ACCESS      not-accessible
    STATUS      mandatory

```

```
INDEX { poolPoolID }
 ::= { poolTable 1 }
```

```
PoolTableEntry ::= SEQUENCE {
    poolPoolID
        WORD,
    poolname
        OCTETSTRING30,
    poolaccountID
        OCTETSTRING15,
    pooltoPrivatePool
        BOOLEAN,
    poolpublicLimit
        WORD,
    pooloverflowLimit
        WORD,
    pooloverflowAllowed
        BOOLEAN,
    poolslotted
        BOOLEAN,
    pooloperationTag
        TOPERATION
}
```

```
poolPoolID OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "ID = 1 for public pool"
    ::= { poolTableEntry 1 }
```

```
poolname OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    ::= { poolTableEntry 2 }
```

```
poolaccountID OBJECT-TYPE
    SYNTAX      OCTETSTRING15
    ACCESS      read-write
    STATUS      mandatory
    ::= { poolTableEntry 3 }
```

```
pooltoPrivatePool OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Not used"
    ::= { poolTableEntry 4 }
```

```
poolpublicLimit OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
```

```

        "Not used"
 ::= { poolTableEntry 5 }

pooloverflowLimit OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Percentage of the public pool which can
         be used for overflow from this private pool"
 ::= { poolTableEntry 6 }

pooloverflowAllowed OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "If true, the pool can overflow to public,
         up to poolOverflowLimit % of the
         public pool."
 ::= { poolTableEntry 7 }

poolslotted OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "If true, the pool contains slotted bandwidth
         (customer special feature) "
 ::= { poolTableEntry 8 }

pooloperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write
    STATUS      mandatory
 ::= { poolTableEntry 99 }

poolStatus OBJECT IDENTIFIER ::= { pools 2 }

poolStatusTotalAllocatedUnits OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
 ::= { poolStatus 1 }

poolStatusTotalManagedUnits OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
 ::= { poolStatus 2 }

bands OBJECT IDENTIFIER ::= { resources 3 }

--                               Pool table must be inserted before Band table

bandTable OBJECT-TYPE
    SYNTAX SEQUENCE OF BandTableEntry

```

```

ACCESS      not-accessible
STATUS      mandatory
 ::= { bands 1 }

bandTableEntry OBJECT-TYPE
SYNTAX      BandTableEntry
ACCESS      not-accessible
STATUS      mandatory
INDEX { bandManID, bandFirstAllocUnit }
 ::= { bandTable 1 }

BandTableEntry ::= SEQUENCE {
    bandManID
        WORD,
    bandFirstAllocUnit
        LONGINT,
    bandNumAllocUnits
        LONGINT,
    bandLocked
        BOOLEAN,
    bandFirstAllocUnitLocked
        BOOLEAN,
    bandNumAllocUnitsLocked
        BOOLEAN,
    bandOperationTag
        TOPERATION
}

bandManID OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Identify the pool this band is in."
 ::= { bandTableEntry 1 }

bandFirstAllocUnit OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
 ::= { bandTableEntry 2 }

bandNumAllocUnits OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
 ::= { bandTableEntry 3 }

bandLocked OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "If true, bandwidth cannot be allocated
    in this band (the portion of the band
    identified by FirstAllocUnitLocked
    and NumAllocUnitsLocked)"

```

```

 ::= { bandTableEntry 4 }

bandFirstAllocUnitLocked OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { bandTableEntry 5 }

bandNumAllocUnitsLocked OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { bandTableEntry 6 }

bandOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write
    STATUS      mandatory
    ::= { bandTableEntry 99 }

transponders OBJECT IDENTIFIER ::= { resources 4 }

transponderTable OBJECT-TYPE
    SYNTAX SEQUENCE OF TransponderTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { transponders 1 }

transponderTableEntry OBJECT-TYPE
    SYNTAX      TransponderTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { transponderIndex }
    ::= { transponderTable 1 }

TransponderTableEntry ::= SEQUENCE {
    transponderIndex
        THALFLONGWORD,
    transponderName
        OCTETSTRING30,
    transponderXponderBW
        LONGINT,
    transponderUplinkCenterRF
        REAL,
    transponderDownlinkCenterRF
        REAL,
    transponderIFCenterBand
        LONGINT,
    transponderLBandUplinkFactor
        REAL,
    transponderLBandDownlinkFactor
        REAL,
    transponderLBandCenter
        LONGINT,
    transponderHighSideInjection
        BOOLEAN,
    transponderFreqBand

```

```

        TFREQBAND,
transponderMaxPower
        REAL,
transponderCurrentPower
        REAL,
transponderRFTranslationFactor
        REAL,
transponderOperationTag
        TOPERATION
}

transponderIndex OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { transponderTableEntry 1 }

transponderName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "the transponder name"
    ::= { transponderTableEntry 2 }

--
in Hz)                                ranges from 36, 54, 72 MHz( unit

transponderXponderBW OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    ::= { transponderTableEntry 3 }

--
2.0 GHz to 15.0 GHz (unit in Hz)

transponderUplinkCenterRF OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    ::= { transponderTableEntry 4 }

--
unit in Hz

transponderDownlinkCenterRF OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { transponderTableEntry 5 }

--
36 MHz). 70 or 140 MHz                70 Mhz (if the transponder size is
--
72 KHz) unit in Hz                    (if the transponder sie is 54 or

transponderIFCenterBand OBJECT-TYPE
    SYNTAX      LONGINT

```

```

ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the IF Center frequency of the transponder,
    depending on the specific
    hardware involved, 70 or 140 MHz."
 ::= { transponderTableEntry 6 }

--
--                                     ranges from 2.0 GHz ro 15.0 GHz
(unit in Hz)

transponderLBandUplinkFactor  OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Enter the value (in GigaHertz) for the Uplink Factor
    to be used for low- or high-side injection
    calculations.
    Difference between RF up link center
    and the L band center frequency."
 ::= { transponderTableEntry 7 }

--
--                                     unit in Hz

transponderLBandDownlinkFactor  OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "Enter the value (in GigaHertz) for the Downlink
    Factor to be used for low- or high-side injection
    calculations.
    Downlink Factor = Uplink Factor Translation Factor
    This is difference between RF down link
    center and L band center frequency."
 ::= { transponderTableEntry 8 }

--
--                                     unit in Hz

transponderLBandCenter  OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { transponderTableEntry 9 }

transponderHighSideInjection  OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "High-side injection is calculated as follows:
    LBand = Uplink Factor RF Uplink
           - or -
    L-band = Downlink Factor RF Downlink
    Indicate whether the LO frequency is higher
    than (high side injection) or lower than

```

```

        (low side injection) the RF frequencies.
        Normally C Band is high side (True) and
        Ku band is low side (FALSE).
 ::= { transponderTableEntry 10 }

transponderFreqBand OBJECT-TYPE
    SYNTAX      TFREQBAND
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Choose the satellite's frequency band.
        Choices include: C-Band or Ku-Band.
        For C-band, high-side injection is the default.
        For Ku-band, low-side injection is the default."
 ::= { transponderTableEntry 11 }

transponderMaxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
 ::= { transponderTableEntry 12 }

transponderCurrentPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
 ::= { transponderTableEntry 13 }

--                                     unit in Hz

transponderRFTranslationFactor OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Enter the value (in GigaHertz) for the Translation
        Factor to be used for low- or high-side injection
        calculations.
        Translation Factor = RF Uplink RF Downlink
        This is the difference between the up link
        Frequency and down link frequency."
 ::= { transponderTableEntry 14 }

transponderOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-only
    STATUS      mandatory
 ::= { transponderTableEntry 99 }

modemTypes OBJECT IDENTIFIER ::= { resources 5 }

modemTypesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ModemTypesTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
 ::= { modemTypes 1 }

```

```

modemTypesTableEntry OBJECT-TYPE
    SYNTAX      ModemTypesTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { modemTypesIndex }
    ::= { modemTypesTable 1 }

ModemTypesTableEntry ::= SEQUENCE {
    modemTypesIndex
        TMODEMTYPE,
    modemTypesName
        OCTETSTRING30,
    modemTypesTxTuningGranularity
        LONGINT,
    modemTypesLowRangeMinPower
        REAL,
    modemTypesHighRangeMinPower
        REAL,
    modemTypesLowRangeMaxPower
        REAL,
    modemTypesHighRangeMaxPower
        REAL,
    modemTypesPowerStep
        REAL,
    modemTypesIsExternal
        TINTERNALEXTERNAL,
    modemTypesLBand
        BOOLEAN,
    modemTypesClockMode
        OCTETSTRING256,
    modemTypesAUPC
        BYTE,
    modemTypesAUPCStep
        REAL,
    modemTypesAUPCMinPower
        REAL,
    modemTypesAUPCMaxPower
        REAL,
    modemTypesAUPCTargetEbNoMin
        REAL,
    modemTypesAUPCTargetEbNoMax
        REAL,
    modemTypesAUPCTrackRateMin
        REAL,
    modemTypesAUPCTrackRateMax
        REAL,
    modemTypesAUPCCarrierLossRemote
        OCTETSTRING256,
    modemTypesAUPCCarrierLossLocal
        OCTETSTRING256,
    modemTypesAUPCCarrierLossTimeUnit
        TUNITS,
    modemTypesAUPCCarrierLossTimeoutMin
        LONGINT,
    modemTypesAUPCCarrierLossTimeoutMax
        LONGINT,
    modemTypesSymbolRateUnit

```

TUNITS,
 modemTypesSymbolRateMin
 REAL,
 modemTypesSymbolRateMax
 REAL,
 modemTypesDataRateUnit
 TUNITS,
 modemTypesDataRateMin
 REAL,
 modemTypesDataRateMax
 REAL,
 modemTypesDataRateStep
 LONGINT,
 modemTypesDataRateStepUnit
 TUNITS,
 modemTypesModulation
 OCTETSTRING256,
 modemTypesEncoding
 OCTETSTRING256,
 modemTypesViterbi
 OCTETSTRING256,
 modemTypesSequential
 OCTETSTRING256,
 modemTypesTurbo
 OCTETSTRING256,
 modemTypesTCM
 OCTETSTRING256,
 modemTypesReedSolomon
 OCTETSTRING256,
 modemTypesReedSolomonViterbi
 OCTETSTRING256,
 modemTypesReedSolomonSequential
 OCTETSTRING256,
 modemTypesReedSolomonTCM
 OCTETSTRING256,
 modemTypesDopplerBuffer
 OCTETSTRING30,
 modemTypesDopplerBufferMSStep
 LONGINT,
 modemTypesDopplerBufferBitsStep
 LONGINT,
 modemTypesDopplerBufferBitsMin
 LONGINT,
 modemTypesDopplerBufferBitsMax
 LONGINT,
 modemTypesDopplerBufferMSMin
 LONGINT,
 modemTypesDopplerBufferMSMax
 LONGINT,
 modemTypesDataRateBPSK12
 OCTETSTRING256,
 modemTypesDataRateQPSK12
 OCTETSTRING256,
 modemTypesDataRateQPSK34
 OCTETSTRING256,
 modemTypesDataRateQPSK78
 OCTETSTRING256,

```

modemTypesDataRateOQPSK12
    OCTETSTRING256,
modemTypesDataRateOQPSK34
    OCTETSTRING256,
modemTypesDataRateOQPSK78
    OCTETSTRING256,
modemTypesDataRate8PSK23
    OCTETSTRING256
}

modemTypesIndex OBJECT-TYPE
    SYNTAX      TMODEMTYPE
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 1 }

modemTypesName OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 2 }

modemTypesTxTuningGranularity OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 3 }

modemTypesLowRangeMinPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 4 }

modemTypesHighRangeMinPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 5 }

modemTypesLowRangeMaxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 6 }

modemTypesHighRangeMaxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 7 }

modemTypesPowerStep OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 8 }

```

```

modemTypesIsExternal OBJECT-TYPE
    SYNTAX      TINTERNALEXTERNAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 9 }

modemTypesLBand OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 10 }

modemTypesClockMode OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 11 }

modemTypesAUPC OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 12 }

modemTypesAUPCStep OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 13 }

modemTypesAUPCMinPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 14 }

modemTypesAUPCMaxPower OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 15 }

modemTypesAUPCTargetEbNoMin OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 16 }

modemTypesAUPCTargetEbNoMax OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 17 }

modemTypesAUPCTrackRateMin OBJECT-TYPE
    SYNTAX      REAL

```

```

ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 18 }

modemTypesAUPCTrackRateMax OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 19 }

modemTypesAUPCCarrierLossRemote OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 20 }

modemTypesAUPCCarrierLossLocal OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 21 }

modemTypesAUPCCarrierLossTimeUnit OBJECT-TYPE
SYNTAX      TUNITS
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 22 }

modemTypesAUPCCarrierLossTimeoutMin OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 23 }

modemTypesAUPCCarrierLossTimeoutMax OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 24 }

modemTypesSymbolRateUnit OBJECT-TYPE
SYNTAX      TUNITS
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 25 }

modemTypesSymbolRateMin OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 26 }

modemTypesSymbolRateMax OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 27 }

```

```

modemTypesDataRateUnit OBJECT-TYPE
    SYNTAX      TUNITS
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 28 }

modemTypesDataRateMin OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 29 }

modemTypesDataRateMax OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 30 }

modemTypesDataRateStep OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 31 }

modemTypesDataRateStepUnit OBJECT-TYPE
    SYNTAX      TUNITS
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 32 }

modemTypesModulation OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 33 }

modemTypesEncoding OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 34 }

modemTypesViterbi OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 35 }

modemTypesSequential OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 36 }

modemTypesTurbo OBJECT-TYPE
    SYNTAX      OCTETSTRING256

```

```

ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 37 }

modemTypesTCM OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 38 }

modemTypesReedSolomon OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 39 }

modemTypesReedSolomonViterbi OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 40 }

modemTypesReedSolomonSequential OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 41 }

modemTypesReedSolomonTCM OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 42 }

modemTypesDopplerBuffer OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 201 }

modemTypesDopplerBufferMSStep OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 202 }

modemTypesDopplerBufferBitsStep OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 203 }

modemTypesDopplerBufferBitsMin OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 204 }

```

```

modemTypesDopplerBufferBitsMax OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 205 }

modemTypesDopplerBufferMSMin OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 206 }

modemTypesDopplerBufferMSMax OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 207 }

modemTypesDataRateBPSK12 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 208 }

modemTypesDataRateQPSK12 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 209 }

modemTypesDataRateQPSK34 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 210 }

modemTypesDataRateQPSK78 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 211 }

modemTypesDataRateOQPSK12 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 212 }

modemTypesDataRateOQPSK34 OBJECT-TYPE
    SYNTAX      OCTETSTRING256
    ACCESS      read-only
    STATUS      mandatory
    ::= { modemTypesTableEntry 213 }

modemTypesDataRateOQPSK78 OBJECT-TYPE
    SYNTAX      OCTETSTRING256

```

```

ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 214 }

modemTypesDataRate8PSK23 OBJECT-TYPE
SYNTAX      OCTETSTRING256
ACCESS      read-only
STATUS      mandatory
 ::= { modemTypesTableEntry 215 }

connections OBJECT IDENTIFIER ::= { midas 4 }

huntGroups OBJECT IDENTIFIER ::= { connections 1 }

--
--           Before any item in HuntGroup table is deleted
or updated, all of the items
--           related to it in other tables such as Directory
--           should be deleted or updated
--           and all of members in the hunt group which are
in ChannelHunt table should
--           be deleted, too.

huntGroupTable OBJECT-TYPE
SYNTAX SEQUENCE OF HuntGroupTableEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
    "A hunt group is a logical group of one or more traffic
    channels of a specific type."
 ::= { huntGroups 1 }

huntGroupTableEntry OBJECT-TYPE
SYNTAX      HuntGroupTableEntry
ACCESS      not-accessible
STATUS      mandatory
INDEX { hgrpHuntIndex }
 ::= { huntGroupTable 1 }

HuntGroupTableEntry ::= SEQUENCE {
    hgrpHuntIndex
        THALFLONGWORD,
    hgrpHuntGroupID
        OCTETSTRING30,
    hgrpDescription
        OCTETSTRING30,
    hgrpHuntGroupType
        THUNTGROUPTYPE,
    hgrpSelectDirect
        THGSELECTTYPE,
    hgrpOperationTag
        TOPERATION
}

hgrpHuntIndex OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory

```

```

 ::= { huntGroupTableEntry 1 }

hgrpHuntGroupID OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the ID of the Hunt Group."
 ::= { huntGroupTableEntry 2 }

hgrpDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows an additional text description of
        the Hunt Group, as needed."
 ::= { huntGroupTableEntry 3 }

hgrpHuntGroupType OBJECT-TYPE
    SYNTAX      THUNTGROUPTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the Hunt Group type"
 ::= { huntGroupTableEntry 4 }

hgrpSelectDirect OBJECT-TYPE
    SYNTAX      THGSELECTTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the search
        order (the order in which the channels will be
        called).
        Choices include: Forward, Reverse, or Pack.
        With the Forward and Reverse options, the
        search algorithm will continue moving the index
        in the forward or reverse direction for as long
        as the current index is busy. The index will
        remain at its last available slot. The Pack
        option will cause the search algorithm to
        always search from the top for the next
        available channel."
 ::= { huntGroupTableEntry 5 }

hgrpOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write
    STATUS      mandatory
 ::= { huntGroupTableEntry 99 }

channelHuntGroups OBJECT IDENTIFIER ::= { connections 2 }

channelHuntGroupTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ChannelHuntGroupTableEntry
    ACCESS      not-accessible

```

```

        STATUS      mandatory
        ::= { channelHuntGroups 1 }

channelHuntGroupTableEntry OBJECT-TYPE
    SYNTAX      ChannelHuntGroupTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { chnhgrpTableIndex }
    ::= { channelHuntGroupTable 1 }

ChannelHuntGroupTableEntry ::= SEQUENCE {
    chnhgrpTableIndex
        THALFLONGWORD,
    chnhgrpHuntGroupID
        OCTETSTRING30,
    chnhgrpNodeID
        TNODEID,
    chnhgrpChannelID
        TCHANNELID,
    chnhgrpIndex
        WORD,
    chnhgrpOperationTag
        TOPERATION
}

chnhgrpTableIndex OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { channelHuntGroupTableEntry 1 }

chnhgrpHuntGroupID OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    ::= { channelHuntGroupTableEntry 2 }

chnhgrpNodeID OBJECT-TYPE
    SYNTAX      TNODEID
    ACCESS      read-write
    STATUS      mandatory
    ::= { channelHuntGroupTableEntry 3 }

chnhgrpChannelID OBJECT-TYPE
    SYNTAX      TCHANNELID
    ACCESS      read-write
    STATUS      mandatory
    ::= { channelHuntGroupTableEntry 4 }

chnhgrpIndex OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { channelHuntGroupTableEntry 5 }

chnhgrpOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION

```

```

ACCESS      read-write
STATUS      mandatory
::= { channelHuntGroupTableEntry 99 }

directories OBJECT IDENTIFIER ::= { connections 3 }

--
--          Before any item in Directory table is deleted
or updated, all of the items related
--          to it in other tables such as Circuit, Channel
(DefaultHGGrpDir) and
--          MultiPointMember should be deleted or updated.
The item in the Directory
--          table can point to either channel or hunt
group. The HuntGroupID field is used
--          to distinguish them. The field should be string
'** none **' if it points to channel,
--          otherwise it should be HuntGroup which it
should point to

directoryTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DirectoryTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { directories 1 }

directoryTableEntry OBJECT-TYPE
    SYNTAX      DirectoryTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { dirIndex }
    ::= { directoryTable 1 }

DirectoryTableEntry ::= SEQUENCE {
    dirIndex
        THALFLONGWORD,
    dirDirectoryNumber
        OCTETSTRING30,
    dirDirectoryType
        TCHANNELTYPE,
    dirDescription
        OCTETSTRING30,
    dirMaxBandwidth
        LONGINT,
    dirHuntGroupID
        OCTETSTRING30,
    dirNodeID
        TNODEID,
    dirChannelID
        TCHANNELID,
    dirOperationTag
        TOPERATION
}

dirIndex OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-write
    STATUS      mandatory

```

```

 ::= { directoryTableEntry 1 }

dirDirectoryNumber OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the Directory Number"
 ::= { directoryTableEntry 2 }

dirDirectoryType OBJECT-TYPE
    SYNTAX      TCHANNELTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Bearer Type (channel type) for the Directory
        Number, either Data, ISDN, or Video."
 ::= { directoryTableEntry 3 }

dirDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is a text description for the Directory
        Number."
 ::= { directoryTableEntry 4 }

dirMaxBandwidth OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the maximum bandwidth value (ISDN only).
        The destination must be a Node-Channel.
        For Hunt Groups, the maximum bandwidth is
        determined automatically, according to the
        selected Hunt Group number. The maximum
        bandwidth is limited to the maximum
        bandwidth setting on the Node-Channel. If the
        Node-Channel maximum bandwidth is set
        to Default, then it is determined from the System
        Setup, ISDN Settings, Grade of Service."
 ::= { directoryTableEntry 5 }

dirHuntGroupID OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
 ::= { directoryTableEntry 6 }

dirNodeID OBJECT-TYPE
    SYNTAX      TNODEID
    ACCESS      read-write
    STATUS      mandatory
 ::= { directoryTableEntry 7 }

```

```

dirChannelID OBJECT-TYPE
    SYNTAX      TCHANNELID
    ACCESS      read-write
    STATUS      mandatory
    ::= { directoryTableEntry 8 }

dirOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write
    STATUS      mandatory
    ::= { directoryTableEntry 99 }

circuits OBJECT IDENTIFIER ::= { connections 4 }

circuitTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CircuitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { circuits 1 }

circuitTableEntry OBJECT-TYPE
    SYNTAX      CircuitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { circuitCircuitID }
    ::= { circuitTable 1 }

CircuitTableEntry ::= SEQUENCE {
    circuitCircuitID
        TCIRCUITID,
    circuitCircuitType
        TCIRCUITTYPE,
    circuitCircuitStatus
        TCIRCUITSTATUS,
    circuitConnectivity
        TCONNECTIVITY,
    circuitNodeA
        TNODEID,
    circuitNodeB
        TNODEID,
    circuitChannelA
        TCHANNELID,
    circuitChannelB
        TCHANNELID,
    circuitEnabled
        TENABLEDDISABLEDSTATUSTYPE,
    circuitActivation
        TACTIVATIONTYPE,
    circuitPriority
        WORD,
    circuitBlocking
        BOOLEAN,
    circuitBumping
        BOOLEAN,
    circuitAccessByDirNum
        BOOLEAN,
    circuitCalledPhoneNumber

```

```

        OCTETSTRING30,
circuitDescription
        OCTETSTRING30,
circuitScheduleEndDate
        TDATE,
circuitScheduleEndTime
        TTIME,
circuitRTSDelay
        WORD,
circuitOperationTag
        TOPERATION,
circuitUpLinkParamsAModulation
        TMODULATION,
circuitUpLinkParamsAFECRate
        TFECRATE,
circuitUpLinkParamsADataRate
        LONGINT,
circuitUpLinkParamsAPowerLevel
        REAL,
circuitUpLinkParamsAClockMode
        TCLOCKMODE,
circuitUpLinkParamsAPreset
        TPRESET,
circuitUpLinkParamsAReedSolomon
        BOOLEAN,
circuitUpLinkParamsAEncode
        TENCODE,
circuitUpLinkParamsAVoiceProtocol
        TVPROTOCOL,
circuitUpLinkParamsAInterfaceBuffer
        LONGINT,
circuitUpLinkParamsADvbFrame
        TDVBFRAME,
circuitUpLinkParamsBModulation
        TMODULATION,
circuitUpLinkParamsBFECRate
        TFECRATE,
circuitUpLinkParamsBDataRate
        LONGINT,
circuitUpLinkParamsBPowerLevel
        REAL,
circuitUpLinkParamsBClockMode
        TCLOCKMODE,
circuitUpLinkParamsBPreset
        TPRESET,
circuitUpLinkParamsBReedSolomon
        BOOLEAN,
circuitUpLinkParamsBEncode
        TENCODE,
circuitUpLinkParamsBVoiceProtocol
        TVPROTOCOL,
circuitUpLinkParamsBInterfaceBuffer
        LONGINT,
circuitUpLinkParamsBDvbFrame
        TDVBFRAME
}

```

```

circuitCircuitID OBJECT-TYPE
    SYNTAX          TCIRCUITID
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        " the currently selected Circuit ID."
    ::= { circuitTableEntry 1 }

circuitCircuitType OBJECT-TYPE
    SYNTAX          TCIRCUITTYPE
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "the currently selected Circuit type."
    ::= { circuitTableEntry 2 }

circuitCircuitStatus OBJECT-TYPE
    SYNTAX          TCIRCUITSTATUS
    ACCESS          read-only
    STATUS          mandatory
    ::= { circuitTableEntry 3 }

circuitConnectivity OBJECT-TYPE
    SYNTAX          TCONNECTIVITY
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "This allows the operator to select the connectivity
        type, either Simplex or Duplex.
        Simplex circuits provide one-way communication,
        and Duplex circuits provide full two-way
        communication."
    ::= { circuitTableEntry 4 }

circuitNodeA OBJECT-TYPE
    SYNTAX          TNODEID
    ACCESS          read-write
    STATUS          mandatory
    ::= { circuitTableEntry 5 }

circuitNodeB OBJECT-TYPE
    SYNTAX          TNODEID
    ACCESS          read-write
    STATUS          mandatory
    ::= { circuitTableEntry 6 }

circuitChannelA OBJECT-TYPE
    SYNTAX          TCHANNELID
    ACCESS          read-write
    STATUS          mandatory
    ::= { circuitTableEntry 7 }

circuitChannelB OBJECT-TYPE
    SYNTAX          TCHANNELID
    ACCESS          read-write
    STATUS          mandatory
    ::= { circuitTableEntry 8 }

```

```

circuitEnabled OBJECT-TYPE
    SYNTAX      TENABLEDDISABLEDSTATUSTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To enable/disable circuit.
        Can't enable/disable a circuit when the controller
        mode is not normal"
    ::= { circuitTableEntry 9 }

circuitActivation OBJECT-TYPE
    SYNTAX      TACTIVATIONTYPE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This option group allows the operator to select
        the Activation type. There are three types to
        choose from:
        PERMANENT this type of Predefined Connection
        is typically used as a dedicated, exclusive
        connection for guaranteed availability.
        SCHEDULED CIRCUIT this type of Predefined
        Connection is available only on an as scheduled
        basis.
        RTS/DTR Request to Send; this type of
        Predefined Connection can only be established
        from a remote site, and is typically used as an
        emergency backup, for circuit restoration
        purposes when terrestrial circuits fail."
    ::= { circuitTableEntry 10 }

circuitPriority OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        " A High Priority call will preempt a Normal call if
        no bandwidth is available.
        Reserved for future use."
    ::= { circuitTableEntry 11 }

circuitBlocking OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { circuitTableEntry 12 }

circuitBumping OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    ::= { circuitTableEntry 13 }

circuitAccessByDirNum OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write

```

```

STATUS      mandatory
 ::= { circuitTableEntry 14 }

circuitCalledPhoneNumber OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
 ::= { circuitTableEntry 15 }

circuitDescription OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the Circuit Description"
 ::= { circuitTableEntry 16 }

circuitScheduleEndDate OBJECT-TYPE
SYNTAX      TDATE
ACCESS      read-write
STATUS      mandatory
 ::= { circuitTableEntry 17 }

circuitScheduleEndTime OBJECT-TYPE
SYNTAX      TTIME
ACCESS      read-write
STATUS      mandatory
 ::= { circuitTableEntry 18 }

circuitRTSDelay OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write
STATUS      mandatory
 ::= { circuitTableEntry 19 }

circuitOperationTag OBJECT-TYPE
SYNTAX      TOPERATION
ACCESS      read-write
STATUS      mandatory
 ::= { circuitTableEntry 99 }

circuitUpLinkParamsAModulation OBJECT-TYPE
SYNTAX      TMODULATION
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the
    modulation standard to be used for the
    connection. Choices include: BPSK, QPSK,
    8PSK, 16QAM, etc."
 ::= { circuitTableEntry 101 }

circuitUpLinkParamsAFECRate OBJECT-TYPE
SYNTAX      TFECRATE
ACCESS      read-write
STATUS      mandatory
DESCRIPTION

```

```

        "This allows the operator to select the
        forward error correction rate (FEC) to
        be used for the channel. Choices include:
        1/2, 2/3, 3/4, 5/6, 7/8, etc."
 ::= { circuitTableEntry 102 }

circuitUpLinkParamsADataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
 ::= { circuitTableEntry 103 }

circuitUpLinkParamsAPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm)"
 ::= { circuitTableEntry 104 }

circuitUpLinkParamsAClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { circuitTableEntry 105 }

circuitUpLinkParamsAPreset OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem A.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { circuitTableEntry 106 }

circuitUpLinkParamsAReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      obsolete
    DESCRIPTION

```

```

        "To select Enable/Disable Reed-Solomon."
 ::= { circuitTableEntry 107 }

circuitUpLinkParamsAEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
         include: Viterbi, Sequential, etc."
 ::= { circuitTableEntry 108 }

circuitUpLinkParamsAVoiceProtocol OBJECT-TYPE
    SYNTAX      TVPROTOCOL
    ACCESS      read-write
    STATUS      mandatory
 ::= { circuitTableEntry 109 }

circuitUpLinkParamsAInterfaceBuffer OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
 ::= { circuitTableEntry 110 }

circuitUpLinkParamsADvbFrame OBJECT-TYPE
    SYNTAX      TDVBFRAME
    ACCESS      read-write
    STATUS      mandatory
 ::= { circuitTableEntry 111 }

circuitUpLinkParamsBModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
         modulation standard to be used for the
         connection. Choices include: BPSK, QPSK,
         8PSK, 16QAM, etc."
 ::= { circuitTableEntry 201 }

circuitUpLinkParamsBFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
         forward error correction rate (FEC) to
         be used for the channel. Choices include:
         1/2, 2/3, 3/4, 5/6, 7/8, etc."
 ::= { circuitTableEntry 202 }

circuitUpLinkParamsBDataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION

```

```

        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
 ::= { circuitTableEntry 203 }

circuitUpLinkParamsBPowerLevel  OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm)"
 ::= { circuitTableEntry 204 }

circuitUpLinkParamsBClockMode  OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { circuitTableEntry 205 }

circuitUpLinkParamsBPreSet  OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem B.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { circuitTableEntry 206 }

circuitUpLinkParamsBReedSolomon  OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      obsolete
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { circuitTableEntry 207 }

circuitUpLinkParamsBEncode  OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
 ::= { circuitTableEntry 208 }

```

```

circuitUpLinkParamsBVoiceProtocol OBJECT-TYPE
    SYNTAX      TVPROTOCOL
    ACCESS      read-write
    STATUS      mandatory
    ::= { circuitTableEntry 209 }

circuitUpLinkParamsBInterfaceBuffer OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    ::= { circuitTableEntry 210 }

circuitUpLinkParamsBDvbFrame OBJECT-TYPE
    SYNTAX      TDVBFRAME
    ACCESS      read-write
    STATUS      mandatory
    ::= { circuitTableEntry 211 }

multiPointCircuits OBJECT IDENTIFIER ::= { connections 5 }

multiPointCircuitTable OBJECT-TYPE
    SYNTAX SEQUENCE OF MultiPointCircuitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { multiPointCircuits 1 }

multiPointCircuitTableEntry OBJECT-TYPE
    SYNTAX      MultiPointCircuitTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { multipointcircuitOneMultID }
    ::= { multiPointCircuitTable 1 }

MultiPointCircuitTableEntry ::= SEQUENCE {
    multipointcircuitOneMultID
        WORD,
    multipointcircuitDescription
        OCTETSTRING30,
    multipointcircuitEnabled
        BOOLEAN,
    multipointcircuitOneMultStatus
        TCIRCUITSTATUS,
    multipointcircuitConnectivity
        TCONNECTIVITY,
    multipointcircuitBroadCast
        OCTETSTRING30,
    multipointcircuitReturn
        OCTETSTRING30,
    multipointcircuitOperationTag
        TOPERATION,
    multipointcircuitAModulation
        TMODULATION,
    multipointcircuitAFECRate
        TFECRATE,
    multipointcircuitADataRate
        LONGINT,

```

```

multipointcircuitAPowerLevel
    REAL,
multipointcircuitAClockMode
    TCLOCKMODE,
multipointcircuitAPreset
    TPRESET,
multipointcircuitAReedSolomon
    BOOLEAN,
multipointcircuitAEncode
    TENCODE,
multipointcircuitAVoiceProtocol
    TVPROTOCOL,
multipointcircuitAInterfaceBuffer
    LONGINT,
multipointcircuitADvbFrame
    TDVBFRAME,
multipointcircuitBModulation
    TMODULATION,
multipointcircuitBFECRate
    TFECRATE,
multipointcircuitBDataRate
    LONGINT,
multipointcircuitBPowerLevel
    REAL,
multipointcircuitBClockMode
    TCLOCKMODE,
multipointcircuitBPreset
    TPRESET,
multipointcircuitBReedSolomon
    BOOLEAN,
multipointcircuitBEncode
    TENCODE,
multipointcircuitBVoiceProtocol
    TVPROTOCOL,
multipointcircuitBInterfaceBuffer
    LONGINT,
multipointcircuitBDvbFrame
    TDVBFRAME
}

multipointcircuitOneMultID OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    ::= { multiPointCircuitTableEntry 1 }

multipointcircuitDescription OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    ::= { multiPointCircuitTableEntry 2 }

multipointcircuitEnabled OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION

```

```

        "Enabled/Disabled circuit"
 ::= { multiPointCircuitTableEntry 3 }

multipointcircuitOneMultStatus OBJECT-TYPE
    SYNTAX      TCIRCUITSTATUS
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This is the activation status of the circuit."
 ::= { multiPointCircuitTableEntry 4 }

multipointcircuitConnectivity OBJECT-TYPE
    SYNTAX      TCONNECTIVITY
    ACCESS      read-write
    STATUS      mandatory
 ::= { multiPointCircuitTableEntry 5 }

multipointcircuitBroadCast OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This is the directory number of the current
        broadcaster."
 ::= { multiPointCircuitTableEntry 6 }

multipointcircuitReturn OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-write
    STATUS      mandatory
 ::= { multiPointCircuitTableEntry 7 }

multipointcircuitOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write
    STATUS      mandatory
 ::= { multiPointCircuitTableEntry 99 }

multipointcircuitAModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        modulation standard to be used for the
        connection. Choices include: BPSK, QPSK,
        8PSK, 16QAM, etc."
 ::= { multiPointCircuitTableEntry 101 }

multipointcircuitAFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include:1/2, 2/3, 3/4, 5/6, 7/8,

```

```

        etc."
 ::= { multiPointCircuitTableEntry 102 }

multipointcircuitADataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
 ::= { multiPointCircuitTableEntry 103 }

multipointcircuitAPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm) "
 ::= { multiPointCircuitTableEntry 104 }

multipointcircuitAClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { multiPointCircuitTableEntry 105 }

multipointcircuitAPreset OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { multiPointCircuitTableEntry 106 }

multipointcircuitAReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      obsolete
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { multiPointCircuitTableEntry 107 }

```

```

multipointcircuitAEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
    ::= { multiPointCircuitTableEntry 108 }

multipointcircuitAVoiceProtocol OBJECT-TYPE
    SYNTAX      TVPROTOCOL
    ACCESS      read-write
    STATUS      mandatory
    ::= { multiPointCircuitTableEntry 109 }

multipointcircuitAInterfaceBuffer OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    ::= { multiPointCircuitTableEntry 110 }

multipointcircuitADvbFrame OBJECT-TYPE
    SYNTAX      TDVBFrames
    ACCESS      read-write
    STATUS      mandatory
    ::= { multiPointCircuitTableEntry 111 }

multipointcircuitBModulation OBJECT-TYPE
    SYNTAX      TModulation
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        modulation standard to be used for the
        connection. Choices include: BPSK, QPSK,
        8PSK, 16QAM, etc."
    ::= { multiPointCircuitTableEntry 201 }

multipointcircuitBFECRate OBJECT-TYPE
    SYNTAX      TFECRate
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include: 1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
    ::= { multiPointCircuitTableEntry 202 }

multipointcircuitBDataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic

```

```

        modem."
 ::= { multiPointCircuitTableEntry 203 }

multipointcircuitBPowerLevel  OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm) "
 ::= { multiPointCircuitTableEntry 204 }

multipointcircuitBClockMode  OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { multiPointCircuitTableEntry 205 }

multipointcircuitBPreSet  OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { multiPointCircuitTableEntry 206 }

multipointcircuitBReedSolomon  OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      obsolete
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { multiPointCircuitTableEntry 207 }

multipointcircuitBEncode  OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
 ::= { multiPointCircuitTableEntry 208 }

multipointcircuitBVoiceProtocol  OBJECT-TYPE
    SYNTAX      TVPROTOCOL

```

```

ACCESS      read-write
STATUS      mandatory
::= { multiPointCircuitTableEntry 209 }

multipointcircuitBInterfaceBuffer OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
::= { multiPointCircuitTableEntry 210 }

multipointcircuitBDvbFrame OBJECT-TYPE
SYNTAX      TDVBFRAME
ACCESS      read-write
STATUS      mandatory
::= { multiPointCircuitTableEntry 211 }

multiPointMembers OBJECT IDENTIFIER ::= { connections 6 }

multiPointMemberTable OBJECT-TYPE
SYNTAX SEQUENCE OF MultiPointMemberTableEntry
ACCESS      not-accessible
STATUS      mandatory
::= { multiPointMembers 1 }

multiPointMemberTableEntry OBJECT-TYPE
SYNTAX      MultiPointMemberTableEntry
ACCESS      not-accessible
STATUS      mandatory
INDEX { multipointmemberIndex }
::= { multiPointMemberTable 1 }

MultiPointMemberTableEntry ::= SEQUENCE {
    multipointmemberIndex
        THALFLONGWORD,
    multipointmemberOneMultID
        WORD,
    multipointmemberDirNumber
        OCTETSTRING30,
    multipointmemberNodeID
        TNODEID,
    multipointmemberChannelID
        TCHANNELID,
    multipointmemberMemberStatus
        TCIRCUITSTATUS,
    multipointmemberOperationTag
        TOPERATION
}

multipointmemberIndex OBJECT-TYPE
SYNTAX      THALFLONGWORD
ACCESS      read-write
STATUS      mandatory
::= { multiPointMemberTableEntry 1 }

multipointmemberOneMultID OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-write

```

```

STATUS      mandatory
 ::= { multiPointMemberTableEntry 2 }

multipointmemberDirNumber OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the directory number associated with the
    channel."
 ::= { multiPointMemberTableEntry 3 }

multipointmemberNodeID OBJECT-TYPE
SYNTAX      TNODEID
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the node and channel of the item."
 ::= { multiPointMemberTableEntry 4 }

multipointmemberChannelID OBJECT-TYPE
SYNTAX      TCHANNELID
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "This is the node and channel of the item."
 ::= { multiPointMemberTableEntry 5 }

multipointmemberMemberStatus OBJECT-TYPE
SYNTAX      TCIRCUITSTATUS
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the activation status of the circuit."
 ::= { multiPointMemberTableEntry 6 }

multipointmemberOperationTag OBJECT-TYPE
SYNTAX      TOPERATION
ACCESS      read-write
STATUS      mandatory
 ::= { multiPointMemberTableEntry 99 }

activeCalls OBJECT IDENTIFIER ::= { connections 7 }

activeCallTable OBJECT-TYPE
SYNTAX SEQUENCE OF ActiveCallTableEntry
ACCESS      not-accessible
STATUS      mandatory
 ::= { activeCalls 1 }

activeCallTableEntry OBJECT-TYPE
SYNTAX      ActiveCallTableEntry
ACCESS      not-accessible
STATUS      mandatory
INDEX { activecallCallSeqNo }
 ::= { activeCallTable 1 }

```

```

ActiveCallTableEntry ::= SEQUENCE {
    activecallCallSeqNo
        LONGINT,
    activecallOneMultID
        TCIRCUITID,
    activecallActivationSource
        TACTIVATIONSOURCE,
    activecallNotTerminatedA
        BOOLEAN,
    activecallNotTerminatedB
        BOOLEAN,
    activecallMaxCallTime
        LONGINT,
    activecallStartTime
        TTIME,
    activecallStartDate
        TDATE,
    activecallTimeBWStart
        TTIME,
    activecallDateBWStart
        TDATE,
    activecallStampTime
        TTIME,
    activecallStampDate
        TDATE,
    activecallTxAllocStartA
        LONGINT,
    activecallTxAllocStartB
        LONGINT,
    activecallTxAllocSizeA
        LONGINT,
    activecallTxAllocSizeB
        LONGINT,
    activecallTxTuningIFA
        LONGINT,
    activecallTxTuningIFB
        LONGINT,
    activecallRxTuningIFA
        LONGINT,
    activecallRxTuningIFB
        LONGINT,
    activecallStartSlotA
        LONGINT,
    activecallNumSlotsA
        LONGINT,
    activecallStartSlotB
        LONGINT,
    activecallNumSlotsB
        LONGINT,
    activecallTxGranularityA
        LONGINT,
    activecallTxGranularityB
        LONGINT,
    activecallRxGranularityA
        LONGINT,
    activecallRxGranularityB
        LONGINT,

```

activecallModemTypeA
 TMODEMTYPE,
 activecallModemTypeB
 TMODEMTYPE,
 activecallHighPowerOptionA
 BOOLEAN,
 activecallHighPowerOptionB
 BOOLEAN,
 activecallNodeActiveA
 BOOLEAN,
 activecallNodeActiveB
 BOOLEAN,
 activecallChanXmitActiveA
 BOOLEAN,
 activecallChanXmitActiveB
 BOOLEAN,
 activecallChanPowerdbmA
 REAL,
 activecallChanPowerdbmB
 REAL,
 activecallCallBlocking
 BOOLEAN,
 activecallCallBumping
 BOOLEAN,
 activecallActive
 BOOLEAN,
 activecallBwUsageA
 TBWUSAGE,
 activecallBwUsageB
 TBWUSAGE,
 activecallDisabledKilled
 TENABLEDKILLTYPE,
 activecallAUPCParamsMaxPowerDeltaA
 REAL,
 activecallAUPCParamsMinPowerDeltaA
 REAL,
 activecallAUPCParamsTargetEbNoA
 REAL,
 activecallAUPCParamsTrackingRateA
 REAL,
 activecallAUPCParamsLocalActionA
 BYTE,
 activecallAUPCParamsRemoteActionA
 BYTE,
 activecallAUPCParamsCarrierLossTimA
 BYTE,
 activecallAUPCParamsEnableA
 BOOLEAN,
 activecallAUPCParamsMaxPowerDeltaB
 REAL,
 activecallAUPCParamsMinPowerDeltaB
 REAL,
 activecallAUPCParamsTargetEbNoB
 REAL,
 activecallAUPCParamsTrackingRateB
 REAL,
 activecallAUPCParamsLocalActionB

BYTE,
 activecallAUPCParamsRemoteActionB
 BYTE,
 activecallAUPCParamsCarrierLossTimB
 BYTE,
 activecallAUPCParamsEnableB
 BOOLEAN,
 activecallcircuitCircuitID
 TCIRCUITID,
 activecallcircuitCircuitType
 TCIRCUITTYPE,
 activecallcircuitCircuitStatus
 TCIRCUITSTATUS,
 activecallcircuitConnectivity
 TCONNECTIVITY,
 activecallcircuitNodeA
 TNODEID,
 activecallcircuitNodeB
 TNODEID,
 activecallcircuitChannelA
 TCHANNELID,
 activecallcircuitChannelB
 TCHANNELID,
 activecallcircuitEnabled
 TNODEENABLESTATE,
 activecallcircuitActivation
 TACTIVATIONTYPE,
 activecallcircuitPriority
 WORD,
 activecallcircuitBlocking
 BOOLEAN,
 activecallcircuitBumping
 BOOLEAN,
 activecallcircuitAccessByDirNum
 BOOLEAN,
 activecallcircuitCalledPhoneNumber
 OCTETSTRING30,
 activecallcircuitDescription
 OCTETSTRING30,
 activecallLinkParamsAModulation
 TMODULATION,
 activecallLinkParamsAFECRate
 TFECRATE,
 activecallLinkParamsADataRate
 LONGINT,
 activecallLinkParamsAPowerLevel
 REAL,
 activecallLinkParamsAClockMode
 TCLOCKMODE,
 activecallLinkParamsAPreset
 TPRESET,
 activecallLinkParamsAReedSolomon
 BOOLEAN,
 activecallLinkParamsAEncode
 TENCODE,
 activecallLinkParamsAVoiceProtocol
 TVPROTOCOL,

```

activecallLinkParamsAInterfaceBuffe
    LONGINT,
activecallLinkParamsADvbFrame
    TDVBFRAME,
activecallLinkParamsBModulation
    TMODULATION,
activecallLinkParamsBFECRate
    TFECRATE,
activecallLinkParamsBDataRate
    LONGINT,
activecallLinkParamsBPowerLevel
    REAL,
activecallLinkParamsBClockMode
    TCLOCKMODE,
activecallLinkParamsBPreset
    TPRESET,
activecallLinkParamsBReedSolomon
    BOOLEAN,
activecallLinkParamsBEncode
    TENCODE,
activecallLinkParamsBVoiceProtocol
    TVPROTOCOL,
activecallLinkParamsBInterfaceBuffe
    LONGINT,
activecallLinkParamsBDvbFrame
    TDVBFRAME
}

```

```

activecallCallSeqNo OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 1 }

```

```

activecallOneMultID OBJECT-TYPE
    SYNTAX      TCIRCUITID
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 2 }

```

```

activecallActivationSource OBJECT-TYPE
    SYNTAX      TACTIVATIONSOURCE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 3 }

```

```

activecallNotTerminatedA OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 4 }

```

```

activecallNotTerminatedB OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 5 }

```

```

activecallMaxCallTime OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 6 }

activecallStartTime OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This is the Start time of the call, i.e., 15:23:00."
    ::= { activeCallTableEntry 7 }

activecallStartDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Start date of call in DD:MM:YYYY format."
    ::= { activeCallTableEntry 8 }

activecallTimeBWStart OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 9 }

activecallDateBWStart OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 10 }

activecallStampTime OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 11 }

activecallStampDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 12 }

activecallTxAllocStartA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 13 }

activecallTxAllocStartB OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory

```

```

        ::= { activeCallTableEntry 14 }

activecallTxAllocSizeA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 15 }

activecallTxAllocSizeB OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 16 }

activecallTxTuningIFA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 17 }

activecallTxTuningIFB OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 18 }

activecallRxTuningIFA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 19 }

activecallRxTuningIFB OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 20 }

activecallStartSlotA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 21 }

activecallNumSlotsA OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 22 }

activecallStartSlotB OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 23 }

activecallNumSlotsB OBJECT-TYPE

```

```

SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 24 }

activecallTxGranularityA OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 25 }

activecallTxGranularityB OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 26 }

activecallRxGranularityA OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 27 }

activecallRxGranularityB OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 28 }

activecallModemTypeA OBJECT-TYPE
SYNTAX      TMODEMTYPE
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 29 }

activecallModemTypeB OBJECT-TYPE
SYNTAX      TMODEMTYPE
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 30 }

activecallHighPowerOptionA OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 31 }

activecallHighPowerOptionB OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
::= { activeCallTableEntry 32 }

activecallNodeActiveA OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory

```

```

        ::= { activeCallTableEntry 33 }

activecallNodeActiveB OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 34 }

activecallChanXmitActiveA OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 35 }

activecallChanXmitActiveB OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 36 }

activecallChanPowerdbmA OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 37 }

activecallChanPowerdbmB OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 38 }

activecallCallBlocking OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 39 }

activecallCallBumping OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 40 }

activecallActive OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 41 }

activecallBwUsageA OBJECT-TYPE
    SYNTAX      TBWUSAGE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 42 }

activecallBwUsageB OBJECT-TYPE

```

```

SYNTAX      TBWUSAGE
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 43 }

activecallDisabledKilled OBJECT-TYPE
SYNTAX      TENABLEDKILLTYPE
ACCESS      read-write
STATUS      mandatory
 ::= { activeCallTableEntry 44 }

activecallAUPCParamsMaxPowerDeltaA OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 101 }

activecallAUPCParamsMinPowerDeltaA OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 102 }

activecallAUPCParamsTargetEbNoA OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 103 }

activecallAUPCParamsTrackingRateA OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 104 }

activecallAUPCParamsLocalActionA OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 105 }

activecallAUPCParamsRemoteActionA OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 106 }

activecallAUPCParamsCarrierLossTimA OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 107 }

activecallAUPCParamsEnableA OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory

```

```

 ::= { activeCallTableEntry 108 }

activecallAUPCParamsMaxPowerDeltaB OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 201 }

activecallAUPCParamsMinPowerDeltaB OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 202 }

activecallAUPCParamsTargetEbNoB OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 203 }

activecallAUPCParamsTrackingRateB OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 204 }

activecallAUPCParamsLocalActionB OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 205 }

activecallAUPCParamsRemoteActionB OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 206 }

activecallAUPCParamsCarrierLossTimB OBJECT-TYPE
    SYNTAX      BYTE
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 207 }

activecallAUPCParamsEnableB OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 208 }

activecallcircuitCircuitID OBJECT-TYPE
    SYNTAX      TCIRCUITID
    ACCESS      read-only
    STATUS      mandatory
    ::= { activeCallTableEntry 1001 }

activecallcircuitCircuitType OBJECT-TYPE

```

```

SYNTAX      TCIRCUITTYPE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The different circuit types include: DATA, ISDN,
    VIDEO, EXTERNAL."
 ::= { activeCallTableEntry 1002 }

activecallcircuitCircuitStatus OBJECT-TYPE
SYNTAX      TCIRCUITSTATUS
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The different circuit status indicators include:
    ACTIVE, FAULT, ACTIVATING, DEACTIVATING,
    RETRYING."
 ::= { activeCallTableEntry 1003 }

activecallcircuitConnectivity OBJECT-TYPE
SYNTAX      TCONNECTIVITY
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the connectivity
    type, either Simplex or Duplex.
    Simplex circuits provide one-way communication,
    and Duplex circuits provide full two-way
    communication."
 ::= { activeCallTableEntry 1004 }

activecallcircuitNodeA OBJECT-TYPE
SYNTAX      TNODEID
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the calling node and channel, i.e., 1234.5."
 ::= { activeCallTableEntry 1005 }

activecallcircuitNodeB OBJECT-TYPE
SYNTAX      TNODEID
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the called node and channel, i.e., 5555.7."
 ::= { activeCallTableEntry 1006 }

activecallcircuitChannelA OBJECT-TYPE
SYNTAX      TCHANNELID
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the calling node and channel, i.e., 1234.5."
 ::= { activeCallTableEntry 1007 }

activecallcircuitChannelB OBJECT-TYPE
SYNTAX      TCHANNELID
ACCESS      read-only

```

```

STATUS      mandatory
DESCRIPTION
    "This is the called node and channel, i.e., 5555.7."
 ::= { activeCallTableEntry 1008 }

activeCallCircuitEnabled OBJECT-TYPE
SYNTAX      TNODEENABLESTATE
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1009 }

activeCallCircuitActivation OBJECT-TYPE
SYNTAX      TACTIVATIONTYPE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The different active call types include: REMOTE, RTS,
    PERMANENT, and SCHEDULE."
 ::= { activeCallTableEntry 1010 }

activeCallCircuitPriority OBJECT-TYPE
SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1011 }

activeCallCircuitBlocking OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1012 }

activeCallCircuitBumping OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1013 }

activeCallCircuitAccessByDirNum OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1014 }

activeCallCircuitCalledPhoneNumber OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1015 }

activeCallCircuitDescription OBJECT-TYPE
SYNTAX      OCTETSTRING30
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1016 }

activeCallLinkParamsAModulation OBJECT-TYPE

```

```

SYNTAX      TMODULATION
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "MODULATION
    This allows the operator to select the
    modulation standard to be used for the
    connection. Choices include: BPSK, QPSK,
    8PSK, 16QAM, etc."
 ::= { activeCallTableEntry 1101 }

activecallLinkParamsAFECRate  OBJECT-TYPE
SYNTAX      TFECRATE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "FEC RATE
    To select the forward error correction
    rate (FEC) to be used for the channel.
    Choices include:1/2, 2/3, 3/4, 5/6, 7/8,
    etc."
 ::= { activeCallTableEntry 1102 }

activecallLinkParamsADataRate  OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Select the data rate from the list of
    available choices, or enter any data
    rate that is valid for the traffic
    modem."
 ::= { activeCallTableEntry 1103 }

activecallLinkParamsAPowerLevel  OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This field allows the operator to enter
    the TX power level setting of the traffic
    modem. Power range limits (0 to -30 dBm)"
 ::= { activeCallTableEntry 1104 }

activecallLinkParamsAClockMode  OBJECT-TYPE
SYNTAX      TCLOCKMODE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "To select the clock mode. Choices include:
    Terrestrial, Internal, Satellite, and
    External Reference."
 ::= { activeCallTableEntry 1105 }

activecallLinkParamsAPreset  OBJECT-TYPE
SYNTAX      TPRESET
ACCESS      read-only

```

```

STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the
    preset option for the traffic modem.
    Choices depend on the modem type, but
    generally include: No, A, B, C, D. For
    modems with no presets, this is always
    set to No. Choosing a preset overrides
    the Modulation, FEC Rate, and Data Rate
    settings."
 ::= { activeCallTableEntry 1106 }

activecallLinkParamsAReedSolomon OBJECT-TYPE
SYNTAX      BOOLEAN
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "To select Enable/Disable Reed-Solomon."
 ::= { activeCallTableEntry 1107 }

activecallLinkParamsAEncode OBJECT-TYPE
SYNTAX      TENCODE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "To select the encoding type. Choices
    include: Viterbi, Sequential, etc."
 ::= { activeCallTableEntry 1108 }

activecallLinkParamsAVoiceProtocol OBJECT-TYPE
SYNTAX      TVPROTOCOL
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1109 }

activecallLinkParamsAInterfaceBuffe OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1110 }

activecallLinkParamsADvbFrame OBJECT-TYPE
SYNTAX      TDVBFRAME
ACCESS      read-only
STATUS      mandatory
 ::= { activeCallTableEntry 1111 }

activecallLinkParamsBModulation OBJECT-TYPE
SYNTAX      TMODULATION
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "MODULATION
    This allows the operator to select the
    modulation standard to be used for the
    connection. Choices include: BPSK, QPSK,
    8PSK, 16QAM, etc."

```

```

 ::= { activeCallTableEntry 1201 }

activecallLinkParamsBFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "FEC RATE
        To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include:1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
 ::= { activeCallTableEntry 1202 }

activecallLinkParamsBDataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
 ::= { activeCallTableEntry 1203 }

activecallLinkParamsBPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm)"
 ::= { activeCallTableEntry 1204 }

activecallLinkParamsBClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select the clock mode. Choices include:
        Terrestrial, Internal, Satellite, and
        External Reference."
 ::= { activeCallTableEntry 1205 }

activecallLinkParamsBPreset OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate

```

```

        settings."
 ::= { activeCallTableEntry 1206 }

activecallLinkParamsBReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { activeCallTableEntry 1207 }

activecallLinkParamsBEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
         include: Viterbi, Sequential, etc."
 ::= { activeCallTableEntry 1208 }

activecallLinkParamsBVoiceProtocol OBJECT-TYPE
    SYNTAX      TVPROTOCOL
    ACCESS      read-only
    STATUS      mandatory
 ::= { activeCallTableEntry 1209 }

activecallLinkParamsBInterfaceBuffe OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
 ::= { activeCallTableEntry 1210 }

activecallLinkParamsBDvbFrame OBJECT-TYPE
    SYNTAX      TDVBFRAME
    ACCESS      read-only
    STATUS      mandatory
 ::= { activeCallTableEntry 1211 }

completedCallRecs OBJECT IDENTIFIER ::= { connections 8 }

completedCallRecTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CompletedCallRecTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
 ::= { completedCallRecs 1 }

completedCallRecTableEntry OBJECT-TYPE
    SYNTAX      CompletedCallRecTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { ccallIndex }
 ::= { completedCallRecTable 1 }

CompletedCallRecTableEntry ::= SEQUENCE {
    ccallIndex
        WORD,
    ccallCircuitType

```

```

        TCIRCUITTYPE,
ccallCircuitID
        TCIRCUITID,
ccallConnectivity
        TCONNECTIVITY,
ccallNodeA
        TNODEID,
ccallNodeB
        TNODEID,
ccallChannelA
        TCHANNELID,
ccallChannelB
        TCHANNELID,
ccallActivation
        TACTIVATIONTYPE,
ccallStartTime
        TTIME,
ccallStartDate
        TDATE,
ccallTimeBWStart
        TTIME,
ccallDateBWStart
        TDATE,
ccallcircuitCalledPhoneNumber
        OCTETSTRING30,
ccallTimeStop
        TTIME,
ccallDateStop
        TDATE,
ccallTimeBWStop
        TTIME,
ccallDateBWStop
        TDATE,
ccallTerminateNode
        WORD,
ccallTerminateChannel
        WORD,
ccallTrafficEbNoA
        REAL,
ccallTrafficEbNoB
        REAL,
ccallBwUsageA
        TBWUSAGE,
ccallBwUsageB
        TBWUSAGE,
ccallOperationtag
        TOPERATION,
ccallUpLinkParamsAModulation
        TMODULATION,
ccallUpLinkParamsAFECRate
        TFECRATE,
ccallUpLinkParamsADataRate
        LONGINT,
ccallUpLinkParamsAPowerLevel
        REAL,
ccallUpLinkParamsAClockMode
        TLOCKMODE,

```

```

ccallUpLinkParamsAPreset
    TPRESET,
ccallUpLinkParamsAReedSolomon
    BOOLEAN,
ccallUpLinkParamsAEncode
    TENCODE,
ccallUpLinkParamsAVoiceProtocol
    TVPROTOCOL,
ccallUpLinkParamsAInterfaceBuffer
    LONGINT,
ccallUpLinkParamsADvbFrame
    TDVBFRAME,
ccallUpLinkParamsBModulation
    TMODULATION,
ccallUpLinkParamsBFECRate
    TFECRATE,
ccallUpLinkParamsBDataRate
    LONGINT,
ccallUpLinkParamsBPowerLevel
    REAL,
ccallUpLinkParamsBClockMode
    TCLOCKMODE,
ccallUpLinkParamsBPreSet
    TPRESET,
ccallUpLinkParamsBReedSolomon
    BOOLEAN,
ccallUpLinkParamsBEncode
    TENCODE,
ccallUpLinkParamsBVoiceProtocol
    TVPROTOCOL,
ccallUpLinkParamsBInterfaceBuffer
    LONGINT,
ccallUpLinkParamsBDvbFrame
    TDVBFRAME,
ccallTerminationReasonStatus
    BYTE,
ccallTerminationReasonCause
    BYTE,
ccallTerminationReasonSubcause
    BYTE
}

ccallIndex OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 1 }

ccallCircuitType OBJECT-TYPE
    SYNTAX      TCIRCUITTYPE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Data, ISDN, or Video."
    ::= { completedCallRecTableEntry 2 }

ccallCircuitID OBJECT-TYPE

```

```

SYNTAX      TCIRCUITID
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the circuit identification number."
 ::= { completedCallRecTableEntry 3 }

ccallConnectivity OBJECT-TYPE
SYNTAX      TCONNECTIVITY
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the connectivity
    type, either Simplex or Duplex.
    Simplex circuits provide one-way communication,
    and Duplex circuits provide full two-way
    communication."
 ::= { completedCallRecTableEntry 4 }

ccallNodeA OBJECT-TYPE
SYNTAX      TNODEID
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 5 }

ccallNodeB OBJECT-TYPE
SYNTAX      TNODEID
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 6 }

ccallChannelA OBJECT-TYPE
SYNTAX      TCHANNELID
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 7 }

ccallChannelB OBJECT-TYPE
SYNTAX      TCHANNELID
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 8 }

ccallActivation OBJECT-TYPE
SYNTAX      TACTIVATIONTYPE
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 9 }

ccallStartTime OBJECT-TYPE
SYNTAX      TTIME
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This is the Start time of the call"
 ::= { completedCallRecTableEntry 10 }

```

```

ccallStartDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This is the Start date of the call."
    ::= { completedCallRecTableEntry 11 }

ccallTimeBWStart OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 12 }

ccallDateBWStart OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 13 }

ccallcircuitCalledPhoneNumber OBJECT-TYPE
    SYNTAX      OCTETSTRING30
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 14 }

ccallTimeStop OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 15 }

ccallDateStop OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 16 }

ccallTimeBWStop OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 17 }

ccallDateBWStop OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 18 }

ccallTerminateNode OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 19 }

ccallTerminateChannel OBJECT-TYPE

```

```

SYNTAX      WORD
ACCESS      read-only
STATUS      mandatory
::= { completedCallRecTableEntry 20 }

ccallTrafficEbNoA OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "Eb/No (bit-error-to-noise-ratio)."
```

```

::= { completedCallRecTableEntry 21 }

ccallTrafficEbNoB OBJECT-TYPE
SYNTAX      REAL
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "Eb/No (bit-error-to-noise-ratio)."
```

```

::= { completedCallRecTableEntry 22 }

ccallBwUsageA OBJECT-TYPE
SYNTAX      TBWUSAGE
ACCESS      read-only
STATUS      mandatory
::= { completedCallRecTableEntry 23 }

ccallBwUsageB OBJECT-TYPE
SYNTAX      TBWUSAGE
ACCESS      read-only
STATUS      mandatory
::= { completedCallRecTableEntry 24 }

ccallOperationtag OBJECT-TYPE
SYNTAX      TOPERATION
ACCESS      read-only
STATUS      mandatory
::= { completedCallRecTableEntry 99 }

ccallUpLinkParamsAModulation OBJECT-TYPE
SYNTAX      TMODULATION
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "This allows the operator to select the
    modulation standard to be used for the
    connection. Choices include: BPSK, QPSK,
    8PSK, 16QAM, etc."
```

```

::= { completedCallRecTableEntry 101 }

ccallUpLinkParamsAFECRate OBJECT-TYPE
SYNTAX      TFECRATE
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "To select the forward error correction
    rate (FEC) to be used for the channel.
```

```

        Choices include:1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
 ::= { completedCallRecTableEntry 102 }

ccallUpLinkParamsADataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data
        rate that is valid for the traffic
        modem."
 ::= { completedCallRecTableEntry 103 }

ccallUpLinkParamsAPowerLevel OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm) "
 ::= { completedCallRecTableEntry 104 }

ccallUpLinkParamsAClockMode OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { completedCallRecTableEntry 105 }

ccallUpLinkParamsAPreset OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { completedCallRecTableEntry 106 }

ccallUpLinkParamsAReedSolomon OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { completedCallRecTableEntry 107 }

```

```

ccallUpLinkParamsAEncode OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
    ::= { completedCallRecTableEntry 108 }

ccallUpLinkParamsAVoiceProtocol OBJECT-TYPE
    SYNTAX      TVPROTOCOL
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 109 }

ccallUpLinkParamsAInterfaceBuffer OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 110 }

ccallUpLinkParamsADvbFrame OBJECT-TYPE
    SYNTAX      TDVBFRAME
    ACCESS      read-only
    STATUS      mandatory
    ::= { completedCallRecTableEntry 111 }

ccallUpLinkParamsBModulation OBJECT-TYPE
    SYNTAX      TMODULATION
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        modulation standard to be used for the
        connection. Choices include: BPSK, QPSK,
        8PSK, 16QAM, etc."
    ::= { completedCallRecTableEntry 201 }

ccallUpLinkParamsBFECRate OBJECT-TYPE
    SYNTAX      TFECRATE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select the forward error correction
        rate (FEC) to be used for the channel.
        Choices include: 1/2, 2/3, 3/4, 5/6, 7/8,
        etc."
    ::= { completedCallRecTableEntry 202 }

ccallUpLinkParamsBDataRate OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Select the data rate from the list of
        available choices, or enter any data

```

```

        rate that is valid for the traffic
        modem."
 ::= { completedCallRecTableEntry 203 }

ccallUpLinkParamsBPowerLevel  OBJECT-TYPE
    SYNTAX      REAL
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This field allows the operator to enter
        the TX power level setting of the traffic
        modem. Power range limits (0 to -30 dBm) "
 ::= { completedCallRecTableEntry 204 }

ccallUpLinkParamsBClockMode  OBJECT-TYPE
    SYNTAX      TCLOCKMODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the clock
        mode. Choices include: Terrestrial, Internal,
        Satellite, and External Reference."
 ::= { completedCallRecTableEntry 205 }

ccallUpLinkParamsBPreset  OBJECT-TYPE
    SYNTAX      TPRESET
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This allows the operator to select the
        preset option for the traffic modem.
        Choices depend on the modem type, but
        generally include: No, A, B, C, D. For
        modems with no presets, this is always
        set to No. Choosing a preset overrides
        the Modulation, FEC Rate, and Data Rate
        settings."
 ::= { completedCallRecTableEntry 206 }

ccallUpLinkParamsBReedSolomon  OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select Enable/Disable Reed-Solomon."
 ::= { completedCallRecTableEntry 207 }

ccallUpLinkParamsBEncode  OBJECT-TYPE
    SYNTAX      TENCODE
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "To select the encoding type. Choices
        include: Viterbi, Sequential, etc."
 ::= { completedCallRecTableEntry 208 }

ccallUpLinkParamsBVoiceProtocol  OBJECT-TYPE

```

```

SYNTAX      TVPROTOCOL
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 209 }

ccallUpLinkParamsBInterfaceBuffer OBJECT-TYPE
SYNTAX      LONGINT
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 210 }

ccallUpLinkParamsBDvbFrame OBJECT-TYPE
SYNTAX      TDVBFRAME
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 211 }

ccallTerminationReasonStatus OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 301 }

ccallTerminationReasonCause OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 302 }

ccallTerminationReasonSubcause OBJECT-TYPE
SYNTAX      BYTE
ACCESS      read-only
STATUS      mandatory
 ::= { completedCallRecTableEntry 303 }

events OBJECT IDENTIFIER ::= { midas 5 }

eventTable OBJECT-TYPE
SYNTAX SEQUENCE OF EventTableEntry
ACCESS      not-accessible
STATUS      mandatory
 ::= { events 1 }

eventTableEntry OBJECT-TYPE
SYNTAX      EventTableEntry
ACCESS      not-accessible
STATUS      mandatory
INDEX { eventIndex }
 ::= { eventTable 1 }

EventTableEntry ::= SEQUENCE {
    eventIndex
        WORD,
    eventEventNo
        WORD,
    eventOID1
        OBJECT IDENTIFIER,

```

```

    eventOID2
        OBJECT IDENTIFIER,
    eventTime
        TTIME,
    eventDate
        TDATE,
    eventDesc
        OCTETSTRING80,
    eventOperationTag
        TOPERATION
}

eventIndex OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 1 }

eventEventNo OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 2 }

eventOID1 OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 3 }

eventOID2 OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 4 }

eventTime OBJECT-TYPE
    SYNTAX      TTIME
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 5 }

eventDate OBJECT-TYPE
    SYNTAX      TDATE
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 6 }

eventDesc OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { eventTableEntry 7 }

eventOperationTag OBJECT-TYPE
    SYNTAX      TOPERATION
    ACCESS      read-write

```

```

        STATUS      mandatory
        ::= { eventTableEntry 99 }

snmpAgent OBJECT IDENTIFIER ::= { midas 6 }

localHostIPAddress OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Local host IP address"
    ::= { snmpAgent 1 }

localPort OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The port that agent is using"
    ::= { snmpAgent 2 }

--                                     In redundant controller system, the
IP address of the other controller

otherHostIPAddress OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "In redundant system, the host IP address which
        other controller is running on."
    ::= { snmpAgent 3 }

otherPort OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "In redundant system, the port the agent
        in other controller is using"
    ::= { snmpAgent 4 }

trapIPAddress OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { snmpAgent 5 }

trapPort OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { snmpAgent 6 }

--                                     Enable or disable to send trap

trapEnabled OBJECT-TYPE

```

```

SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Enable/Disable to display traps"
 ::= { snmpAgent 7 }

--
agent
--
maxTrapsKept OBJECT-TYPE
    SYNTAX      INTEGER (500..5000)
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Maximum number of traps saved in agent"
    ::= { snmpAgent 8 }

trapTable OBJECT-TYPE
    SYNTAX SEQUENCE OF TrapTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    ::= { snmpAgent 9 }

trapTableEntry OBJECT-TYPE
    SYNTAX      TrapTableEntry
    ACCESS      not-accessible
    STATUS      mandatory
    INDEX { trapSequenceNumber }
    ::= { trapTable 1 }

TrapTableEntry ::= SEQUENCE {
    trapSequenceNumber
        THALFLONGWORD,
    trapWhichTrap
        WORD,
    trapObjID
        OBJECT IDENTIFIER
}

trapSequenceNumber OBJECT-TYPE
    SYNTAX      THALFLONGWORD
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Sequence number of traps sent."
    ::= { trapTableEntry 1 }

trapWhichTrap OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "trap number to define type of traps"
    ::= { trapTableEntry 2 }

```

```

--                                     The whole record referred to by it
is affected

trapObjID OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "object ID in trap."
    ::= { trapTableEntry 3 }

debug OBJECT IDENTIFIER ::= { snmpAgent 20 }

debugipwSNMP OBJECT IDENTIFIER ::= { debug 1 }

ipwSNMPLocalPort OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 1 }

ipwSNMPLocalHost OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 2 }

ipwSNMPObjCount OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 3 }

ipwSNMPLastObjID OBJECT-TYPE
    SYNTAX      OBJECT IDENTIFIER
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 4 }

ipwSNMPLastObjType OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 5 }

ipwSNMPLastObjValue OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 6 }

ipwSNMPInBufferSize OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 7 }

```

```

ipwSNMPOutBufferSize OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 8 }

ipwSNMPWinsockInfo OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 9 }

ipwSNMPWinsockMaxDatagramSize OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 10 }

ipwSNMPWinsockStatus OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugipwSNMP 11 }

debugAgent OBJECT IDENTIFIER ::= { debug 2 }

debugAgentSafeListCount OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugAgent 1 }

debugAgentToManagerCollCount OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugAgent 2 }

debugAgentFromManagerCollCount OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugAgent 3 }

debugAgentTimerInterval OBJECT-TYPE
    SYNTAX      LONGINT
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugAgent 4 }

debugAgentTimerEnabled OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-only
    STATUS      mandatory
    ::= { debugAgent 5 }

debugAgentIsDebugingTag OBJECT-TYPE

```

```

SYNTAX      BOOLEAN
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "Enable/Disable to display debugging information
    such as in/out SNMP messages"
 ::= { debugAgent 6 }

maintenance OBJECT IDENTIFIER ::= { midas 7 }

--
--          Use the Service Message option to send a
message or a command from the
--          Operator Workstation to the remote node
--          terminal, and directly to the node.
--          The message can be up to 28 characters in
length. All commands issued to a
--          remote node must begin with a prefix of
--          "IDC>". Refer to the IDC-150 Network
--          Control Modem Operation and Maintenance
Manual for a complete listing of
--          all commands supported.
--          The purpose of this option is to allow
--          simple, text-based communication
--          between the remote node and the operator,
and to provide the operator
--          with remote command capability. Any commands
--          of up to 28 characters
--          that can be accepted from the remote node
terminal through the user
--          port can be sent as a service message. The
--          operator can send such
--          commands to a remote node for configuration
--          or call requests.
--          All service messages from the remote node to
the NMS are displayed in
--          the Event Log. Similarly, all responses for
--          commands originating from the
--          Operator Workstation are displayed in the
Event Log.

serviceMessage OBJECT IDENTIFIER ::= { maintenance 1 }

toNode OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Node ID which service message is sent to"
    ::= { serviceMessage 1 }

toChannel OBJECT-TYPE
    SYNTAX      WORD
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Not used"
    ::= { serviceMessage 2 }

```

-- Specify the message to send

```
outStr OBJECT-TYPE
    SYNTAX      OCTETSTRING28
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The content of service message"
    ::= { serviceMessage 3 }
```

```
responseStr OBJECT-TYPE
    SYNTAX      OCTETSTRING80
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Not used"
    ::= { serviceMessage 4 }
```

```
send OBJECT-TYPE
    SYNTAX      BOOLEAN
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "to send the message"
    ::= { serviceMessage 5 }
```

```
--
--
the trap
--
from the
--
When
--
given,
--
1.3.6.1.4.1.6247.2
--
value for
--
(iso.org.dod.internet.private.enterprise.
--
is abbreviated
--
table index is
--
"n".
--
following events:
--
Description String(s)
--
m.3.4.1.1.1.n "No Bandwidth available"
--
"No Bandwidth available"
```

MIDAS Trap Definitions
The following list gives all of
messages defined for transmission
MIDAS controller to the HPOV NMS.
object identifiers (OIDs) are
they all assume a prefix of:
for MIDAS, which is the numeric
comtechefdata.midas). This prefix
"m" in the list below. Where a
part of the OID, it is shown as
MIDAS Event traps are sent for the
MIDAS Events OID1 OID2
(5) m.2.3.1.1.1.n

```

--
"Satellite power limit reached"
--
"Bandwidth & satellite power limit"
--
"Site power limit reached"
--
"Bandwidth & site power limit"
--
"Taking node offline, not responding"          (8)  m.2.3.1.1.1.n  none
--
"Node has not responded recently"              (9)  m.2.3.1.1.1.n  none
--
"Remote node is logging on to network"        (11) m.2.3.1.1.1.n  none
--
"No control channels exist"                    (14)      none      none
--
"Invalid packet received"                     (17)  m.2.3.1.1.1.n  none
--
"Changed to maintenance mode"                 (57)      none      none
--
"Changed to shutdown mode"                   (58)      none      none
--
"Changed to normal mode"                     (59)      none      none
--
"Failed to allocate CommsChannel object"      (60)  m.3.1.1.1.1.n  none
--
"CommsChannel allocated but not initialized"
--
"Failed to initialize Primary Modem"
--
"Failed to init LinkSync, AFC & UPC are not enabled"
--
"Error initializing Control Channel, no memory"
--
"Unable to create new Control Channel instance for"
--
"Error adding Control Channel"
--
"Error initializing control channel modem "    (62)  m.3.1.1.1.1.n  none
--
m.2.3.1.1.1.n "Power was too low, was set to minimum " (68)  m.4.7.1.1.1.n
--
"Reset control channel port"                  (69)  m.3.1.1.1.1.n  none
--
"NMS Operator terminate"                      (70)  m.4.7.1.1.1.n  none
--
"Pre-empted by high-priority"                 (78)  m.4.7.1.1.1.n  none
--
"Database ActiveCall record deleted"
--
"Database Node record deleted"                m.2.3.1.1.1.n  none
--
"Database Channel record deleted"             m.2.4.1.1.1.n  none
--
"Database ChnHGrp record deleted"            m.4.2.1.1.1.n  none

```

--		m.4.3.1.1.1.n	none
"Database Dir record deleted"			
--		m.4.1.1.1.1.n	none
"Database HGrp record deleted"			
--		m.3.1.1.1.1.n	none
"Database ControlChannel record deleted"			
--		m.4.4.1.1.1.n	none
"Database Circuit record deleted"			
--		m.3.3.1.1.1.n	none
"Database Band record deleted"			
--		m.2.1.1.1.1.n	none
"Database SiteType record deleted"			
--		m.2.2.1.1.1.n	none
"Database Site record deleted"			
--		m.4.5.1.1.1.n	none
"Database lMult record deleted"			
--		m.4.6.1.1.1.n	none
"Database lMultDir record deleted"			
--		m.3.2.1.1.1.n	none
"Database Pool record deleted"			
--		m.3.4.1.1.1.n	none
"Database Transponder record deleted"			
--	(79)	m.4.7.1.1.1.n	none
"Database ActiveCall record updated"			
--		m.2.3.1.1.1.n	none
"Database Node record updated"			
--		m.2.4.1.1.1.n	none
"Database Channel record updated"			
--		m.4.2.1.1.1.n	none
"Database ChnHGrp record updated"			
--		m.4.3.1.1.1.n	none
"Database Dir record updated"			
--		m.4.1.1.1.1.n	none
"Database HGrp record updated"			
--		m.3.1.1.1.1.n	none
"Database ControlChannel record updated"			
--		m.4.4.1.1.1.n	none
"Database Circuit record updated"			
--		m.3.3.1.1.1.n	none
"Database Band record updated"			
--		m.2.1.1.1.1.n	none
"Database SiteType record updated"			
--		m.2.2.1.1.1.n	none
"Database Site record updated"			
--		m.4.5.1.1.1.n	none
"Database lMult record updated"			
--		m.4.6.1.1.1.n	none
"Database lMultDir record updated"			
--		m.3.2.1.1.1.n	none
"Database Pool record updated"			
--		m.3.4.1.1.1.n	none
"Database Transponder record updated"			
--	(80)	m.4.7.1.1.1.n	none
"Database ActiveCall record added"			
--		m.2.3.1.1.1.n	none
"Database Node record added"			

--		m.2.4.1.1.1.n	none
"Database Channel record added"			
--		m.4.2.1.1.1.n	none
"Database ChnHGrp record added"			
--		m.4.3.1.1.1.n	none
"Database Dir record added"			
--		m.4.1.1.1.1.n	none
"Database HGrp record added"			
--		m.3.1.1.1.1.n	none
"Database ControlChannel record added"			
--		m.4.4.1.1.1.n	none
"Database Circuit record added"			
--		m.3.3.1.1.1.n	none
"Database Band record added"			
--		m.2.1.1.1.1.n	none
"Database SiteType record added"			
--		m.2.2.1.1.1.n	none
"Database Site record added"			
--		m.4.5.1.1.1.n	none
"Database lMult record added"			
--		m.4.6.1.1.1.n	none
"Database lMultDir record added"			
--		m.3.2.1.1.1.n	none
"Database Pool record added"			
--		m.3.4.1.1.1.n	none
"Database Transponder record added"			
--	(71)	m.4.7.1.1.1.n	
m.2.3.1.1.1.n	"Power was too high, was set to maximum"		
--	(107)	m.4.4.1.1.1.n	none
"RTS Circuit xxxx or Calling node does not exist"			
--			
"RTS Circuit xxxx is changed, to synchronize to node"			
--	(111)	m.2.3.1.1.1.n	none
"Node Startup: Invalid active call sequence (nnnn,m) terminated"			
--	(112)	m.2.3.1.1.1.n	none
"Node Startup: Invalid offline active call (nnnn,m) terminated"			
--	(114)	m.4.7.1.1.1.n	none
"Normal remote termination"			
--			
"No answer timer expired"			
--			
"User went on-hook"			
--			
"User port termination"			
--			
"RTS went inactive"			
--			
"Call time expired"			
--	(115)	m.2.3.1.1.1.n	none
"Node startup timeout"			
--	(116)	m.2.3.1.1.1.n	none
"Node Startup: DAC startup bad response for node nnnn"			
--			
"Node Startup nnnn invalid response"			
--	(117)	m.2.3.1.1.1.n	none
"Config Report (nnnn,0). Node not in internal mode"			

```

--
"Int Node Config Report (nnnn,m). Mismatch in channel type"
--
"Config Report (nnnn,0). Node not in external mode"
--
"Config Report (nnnn,m). Mismatch in channel type"
--
"Config Report (nnnn,m). Mismatch in channel"
--
"Startup Report (nnnn,0), Node may not be in internal mode"
--
"Startup Report (nnnn,m), node may not be in external mode"
--
"Startup Report (nnnn,m), Channel status mismatch(E/D)"
--
(118) m.4.7.1.1.1.n none
"type call startup failed: circuit_desc event_description "
--
(125) none none
"Move of database to DBSAVE has failed"
--
"The connection between primary and secondary NMS is broken"
--
"PrimarySecondaryConnection sending fails"
--
"PrimarySecondaryConnection is not connected"
--
"PrimarySecondaryConnection SendQ is full"
--
"SendtoSendQ is full"
--
"The secondary NMS is shutdown because primary NMS is too
--
busy to transfer db"
--
"SendQ in PrimarySecondaryConnection is full"
--
"Server SendQ is full"
--
"PrimarySecondaryConnection is broke"
--
"PrimarySecondaryConnection SendQ is full"
--
"NMS fails to communicate to the control channel modem A"
--
"NMS fails to communicate to the link sync modem A"
--
"NMS fails to communicate to the control channel modem B"
--
"NMS fails to communicate to the link sync modem B"
--
"The control channel modem A is set online, auto"
--
"The link sync modem A is set online, auto"
--
"The control channel modem B is set online, manual"
--
"The link sync modem B is set online, manual"

```

```

--
"NMS enters Exception for the ctrlchn modem A and B failing"
--
"NMS enters Exception for the ctrlchn modem failing"
--
"The secondary NMS has become the primary NMS"
--
"The primary NMS has become the secondary NMS"
--
"The secondary NMS is testing itself"
--
"The NMS is detecting if there is any other outbound"
--
"The primary NMS controller is in exception state for self
test failing"
--
"The secondary NMS controller is in exception state for self
test fail"
--
"The controller is in exception state for self test failing"
--
"The secondary NMS is transforming to primary NMS"
--
"The secondary NMS is monitoring primary outbound"
--
"The secondary NMS is testing itself"
--
"The secondary NMS is resetting"
--
"The secondary NMS is detecting no primary outbound"
--
"The secondary NMS is in exception for waiting for ctrl
or bandrec"
--
"The secondary NMS is reset by new control channel
from primary"
--
"The controller is in exception state for detecting other outbound"
--
"The NMS is testing itself"
--
"The secondary NMS begins to transform to primary NMS for
auto switchover"
--
(122) m.4.7.1.1.1.n           none
"Data call retry"
--
(123) m.4.7.1.1.1.n           none
"Maximum call retries reached"
--
(129) m.2.3.1.1.1.n           none
"Node requesting to come on-line"
--
(130) none                     none
"Max number of enabled nodes reached"

```

```

--                               (136) m.3.1.1.1.1.n      none
"Excessive offset on MIDAS control channel"
--                               (137) m.2.3.1.1.1.n      none
"Excessive offset on node ctrl channel"
--                               (146) m.2.3.1.1.1.n      none
"Frequency alignment performed for node"
--                               (147) none              none
"Frequency alignment performed for NMS"
--                               (148) m.2.3.1.1.1.n      none
"Invalid parameter in inbound CC message"
--                               (151) m.2.4.1.1.1.n      none
"Attempt to disable channel failed"
--                               (152) m.2.4.1.1.1.n      none
"Attempt to enable channel failed"
--                               (153) m.2.4.1.1.1.n      none
"ISDN Dica is off line or on line"
--                               (155) none              none
"Invalid Dongle Key"

```

```
midasTrapEvent TRAP-TYPE
```

```
  ENTERPRISE midas
```

```
  VARIABLES { trapSequenceNumber, eventIndex, eventEventNo, eventOID1,
eventOID2, eventTime, eventDate, eventDesc }
```

```
  DESCRIPTION
```

```

  "MIDAS Event Traps are sent when an event
  occurs in the MIDAS system which is
  considered time critical. All events are
  stored in the event table, and are available
  to the NMS. The NMS should periodically
  get the latest events by sending Get Next
  requests starting from the last event in
  its database. For time-critical events,
  the MIDAS controller will send a trap in
  addition to storing the event in the event
  table. Initially the NMS will simply display
  the event text, but in the future some action
  may be required on the objects indicated in
  OID1 and/or OID2. Many events are not listed
  here. Some are not listed because they are
  not important or time-sensitive enough to
  warrant a trap. Others are not here because
  special traps have been defined for them.
  The purpose of those traps is primarily to
  send specific status information to the NMS
  so that it can update its database to stay
  in sync with the controller."

```

```
 ::= 1
```

```

--                               The Value Change trap gives a
general format for informing the Client that
--                               a value in the database has
changed. The trap consists of one or more
--                               OID, value pairs for the object(s)
which changed, typically one or more
--                               fields from a record in a table.

```

```
midasTrapObjsUpdated TRAP-TYPE
```

```

ENTERPRISE midas
VARIABLES { trapSequenceNumber }
DESCRIPTION
    "Objects in variable list are updated."
 ::= 2

--
-- indicates the some or all of the fields
--
-- have changed. The NMS should
--
-- the record to ensure that the database
--
--
midasTrapDBRecUpdated TRAP-TYPE
    ENTERPRISE midas
    VARIABLES { trapSequenceNumber }
    ::= 3

--
-- indicates the specified database
--
-- should request the value of
--
-- that the database matches.

midasTrapDBRecInserted TRAP-TYPE
    ENTERPRISE midas
    VARIABLES { trapSequenceNumber }
    ::= 4

--
-- indicates the specified database
--
-- database. The NMS should delete
--
--
midasTrapDBRecDeleted TRAP-TYPE
    ENTERPRISE midas
    VARIABLES { trapSequenceNumber }
    ::= 5

END

```

The Database Record Update trap of the specified database record request the value of all fields in matches.

The Database Record Update trap record has been added. The NMS all fields in the record to ensure that the database matches.

The Database Record Update trap record has been deleted from the the record from its database.

METRIC CONVERSIONS

Units of Length

Unit	Centimeter	Inch	Foot	Yard	Mile	Meter	Kilometer	Millimeter
1 centimeter	—	0.3937	0.03281	0.01094	6.214×10^{-6}	0.01	—	—
1 inch	2.540	—	0.08333	0.2778	1.578×10^{-5}	0.254	—	25.4
1 foot	30.480	12.0	—	0.3333	1.893×10^{-4}	0.3048	—	—
1 yard	91.44	36.0	3.0	—	5.679×10^{-4}	0.9144	—	—
1 meter	100.0	39.37	3.281	1.094	6.214×10^{-4}	—	—	—
1 mile	1.609×10^5	6.336×10^4	5.280×10^3	1.760×10^3	—	1.609×10^3	1.609	—
1 mm	—	0.03937	—	—	—	—	—	—
1 kilometer	—	—	—	—	0.621	—	—	—

Temperature Conversions

Unit	° Fahrenheit	° Centigrade
32° Fahrenheit	—	0 (water freezes)
212° Fahrenheit	—	100 (water boils)
-459.6° Fahrenheit	—	273.1 (absolute 0)

Formulas
$C = (F - 32) * 0.555$
$F = (C * 1.8) + 32$

Units of Weight

Unit	Gram	Ounce Avoirdupois	Ounce Troy	Pound Avoir.	Pound Troy	Kilogram
1 gram	—	0.03527	0.03215	0.002205	0.002679	0.001
1 oz. avoir.	28.35	—	0.9115	0.0625	0.07595	0.02835
1 oz. troy	31.10	1.097	—	0.06857	0.08333	0.03110
1 lb. avoir.	453.6	16.0	14.58	—	1.215	0.4536
1 lb. Troy	373.2	13.17	12.0	0.8229	—	0.3732
1 kilogram	1.0×10^3	35.27	32.15	2.205	2.679	—



2114 WEST 7TH STREET TEMPE ARIZONA 85281 USA
480 • 333 • 2200 PHONE
480 • 333 • 2161 FAX