

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2009

Y. Shi, Ed.
H3C Tech. Co., Ltd
D. Perkins, Ed.
SNMPinfo
C. Elliott, Ed.
Cisco Systems, Inc.
Y. Zhang, Ed.
Fortinet, Inc.
October 11, 2008

CAPWAP Protocol Base MIB
draft-ietf-capwap-base-mib-01

Status of This Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 14, 2009.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular, it describes managed objects for modeling the Control And Provisioning of Wireless Access Points (CAPWAP) Protocol.

Table of Contents

1.	Introduction	3
2.	The Internet-Standard Management Framework	3
3.	Terminology	3
4.	Conventions	3
5.	Overview	4
6.	Structure of the MIB Module	5
6.1.	The capwapBaseObjects Subtree	6
6.2.	The capwapBaseConformance Subtree	6
6.3.	The capwapBaseNotifications Subtree	6
6.4.	Brief Description of MIB Objects	6
7.	Relationship to Other MIB Modules	7
7.1.	Relationship to the SNMPv2-MIB	7
7.2.	Relationship to the IF-MIB	7
7.3.	Relationship to the ENTITY-MIB	8
7.4.	Relationship to MIB standards of other SDOs	9
7.5.	MIB modules required for IMPORTS	9
8.	Example of CAPWAP-BASE-MIB Usage	9
9.	Definitions	11
10.	Security Considerations	49
11.	IANA Considerations	50
11.1.	IANA Considerations for CAPWAP-BASE-MIB	50
11.2.	IANA Considerations for ifType	50
12.	Contributors	50
13.	Acknowledgements	50
14.	References	50
14.1.	Normative References	50
14.2.	Informative References	51

1. Introduction

The CAPWAP Protocol [[I-D.ietf-capwap-protocol-specification](#)] defines a standard, interoperable protocol, which enables an Access Controller (AC) to manage a collection of Wireless Termination Points(WTPs)

This document defines a MIB module that can be used to manage CAPWAP implementations. This MIB module covers both configuration and WTP status-monitoring aspects of CAPWAP, and provides a way to reuse current MIB standards and future extensions for any wireless binding technology.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to [section 7 of RFC 3410](#) [[RFC3410](#)].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].

3. Terminology

This document uses terminology from the document describing the CAPWAP Protocol specification [[I-D.ietf-capwap-protocol-specification](#)]. WTPs are viewed as remote RF interfaces controlled by the AC via CAPWAP protocol. The CAPWAP protocol supports two modes of operation: Split and Local MAC. In Split MAC mode all L2 wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between the AC and the WTPs. The Local MAC mode of operation allows for the data frames to be either locally bridged, or tunneled as 802.3 frames. From AC, operator could centrally control WTPs configuration and monitor their status. CAPWAP use DTLS protocol to implement control channel security.

4. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

5. Overview

The CAPWAP Protocol MIB module (CAPWAP-BASE-MIB) have the following design objectives:

- To implement an architecture that uses SNMP for the management and control of wireless networks, answering the operators requirements for centralized management, whatever the deployment or configuration of the network devices (centralized, distributed, or some mix)
- To be consistent with CAPWAP protocol
- To reuse current MIB standards and future extensions for a wireless binding technology
- To enable interoperability between vendors
- To meet operator requirements for centralized architectures

The basic idea of CAPWAP-BASE-MIB is:

- The CAPWAP-BASE-MIB MUST be run on the AC devices and is NOT REQUIRED on the WTP devices. It follows same idea as CAPWAP protocol: Centralized Control
- As a generic mechanism, it is independent of any wireless binding technologies and defined by an independent MIB file
- To be independent of any wireless binding technologies and have ability to reuse MIB standards of other SDOs, is the main challenge for design of the MIB
- ifIndex [[RFC2863](#)] will be used as common handler for corresponding interfaces in the CAPWAP-BASE-MIB and specific wireless technologies MIB modules
- The operator could manage and control the centralized wireless architectures using multiple MIB standards defined by multiple SDOs, while keeping them loosely coupled

It is designed to satisfy the following requirements and constraints:

- From AC to centrally manage and monitor WTPs
- The MIB module supports CAPWAP protocol parameters queries
- The MIB module supports showing WTPs current state

- The MIB module provides the information of AC, WTPs, radio and station objects' basic property and their relationship
- The MIB module supports indicating the 'WTP Virtual Radio Interface' and PHY radio's mapping relationship
- The counters are provided for WTP, radio's reboot event, hardware event failure and so on
- The MIB module provides the various notification like channel up, join failure and so on

Before coming to details of CAPWAP-BASE-MIB module, it will introduce how CAPWAP-BASE-MIB is able to be independent of any wireless binding technologies and reuse MIB standards of other SDOs. As centralized Wireless architecture, the operator has to prepare configuration on the AC before WTPs connects to AC. For any wireless binding technology, the configuration and management of radio is very important. Under centralized Wireless architecture, according to [\[I-D.ietf-capwap-protocol-specification\]](#), a specific PHY radio could be identified by identifier of a WTP and radio (WTP id + radio id). As usual, the standard of a binding technology provides MIB standard for radio management on its own. For example, according to IEEE 802.11 WG MIB standards, the MIB tables such as Dot11OperationTable are able to support WTP radio configuration. These tables use ifIndex as the index, and work well under standalone Wireless architecture.

To reuse MIB objects (defined by SDOs such as IEEE) for radio is very important, and the key point is to reuse the idea of ifIndex. So it is required a way to maintain the mapping relationship between 'WTP id + radio id' and 'ifIndex'. As a generic mechanism, ifIndex can identify an interface in abstract way, and it does NOT care for an interface's PHY location (either on the WTP or AC). AC can have interfaces of 'WTP Virtual Radio Interface' ifType, which will logically represent PHY radios on the WTP. It looks like that PHY radios are located on the AC, and PHY location of WTP (radio) is hidden to the operator. Operator can operate radios by MIB tables (such as IEEE 802.11 WG's) with ifIndex of 'WTP Virtual Radio Interface'. As an Abstract interface, 'WTP Virtual Radio Interface' could be used by any wireless binding technology such as IEEE 802.11 and 802.16. The table of capwapRadioBindTable will indicate the mapping relationship between 'WTP id + Radio id' and IfIndex.

6. Structure of the MIB Module

6.1. The capwapBaseObjects Subtree

The subtree provides information for statistic data and configuration parameters of WTP and radio.

6.2. The capwapBaseConformance Subtree

The subtree provides conformance information of MIB objects.

6.3. The capwapBaseNotifications Subtree

The subtree describes the notifications defined in the MIB module, and their purpose.

6.4. Brief Description of MIB Objects

The MIB objects were derived from the CAPWAP protocol document [I-D.ietf- capwap-protocol-specification].

1) capwapBaseWtpStateTable

The WTPs status table is used to indicate each WTP's CAPWAP FSM state.

2) capwapBaseWtpTable

The WTPs table is used for providing property and configuration information in details for WTPs in running state.

3) capwapBaseRadioBindTable

The radio bind table is used to indicate the mapping relationship between logical interface of 'WTP Virtual Radio Interface' ifType and PHY radio.

4) capwapBaseStationTable

The station table is used for providing stations' basic property information.

5) capwapBaseWtpRebootStatsTable

The WTP reboot statistic table is used for collecting WTP reboot count, link failure count, hardware failure count and so on.

6) capwapBaseRadioStatsTable

The WTP radio statistic table is used for collecting radio reset

count, channel change count, hardware failure count and so on.

7. Relationship to Other MIB Modules

7.1. Relationship to the SNMPv2-MIB

The 'system' group in the SNMPv2-MIB [[RFC3418](#)] is defined as being mandatory for all systems, and the objects apply to the entity as a whole. The 'system' group provides identification of the management entity and certain other system-wide data. The CAPWAP-BASE-MIB does not duplicate those objects.

7.2. Relationship to the IF-MIB

The Interfaces Group [[RFC2863](#)] defines generic managed objects for managing interfaces. This memo contains the media-specific extensions to the Interfaces Group for managing WTP radio object that are modeled as interfaces.

IF-MIB is required to support on the AC. For each PHY radio on the WTP, it will have a logical interface of 'WTP Virtual Radio Interface' ifType on the AC. 'WTP Virtual Radio Interface' provides a way to configure radio's parameters and query radio's traffic statistics, and reuse the MIBs defined by a specific wireless binding technology. The interface SHOULD be modeled as an ifEntry and provide appropriate interface information.

Some specific interpretations of ifTable for CAPWAP-BASE-MIB are as follow.

Object	Use for the CAPWAP-BASE-MIB.
ifIndex	Each interface of 'WTP Virtual Radio Interface' ifType maybe be represented by an ifEntry.
ifDescr	Description of the interface of 'WTP Virtual Radio Interface' ifType.
ifType	IANAifType of 'WTP Virtual Radio Interface'.
ifName	Textual name (unique on this system) of the interface or an octet string of zero length.
ifAlias	The nonvolatile 'alias' name for the interface, as specified by a network manager.
ifPhysAddress	The physical address of the interface, such as the BSSID of a 802.11 radio.

ifAdminStatus	This variable indicates the operator's intent as to whether PHY should be enabled, disabled, or running in some diagnostic testing mode on this interface. Also see [RFC2863].
ifOperStatus	This value reflects the actual or operational status of radio. Also see [RFC2863].
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state. Also see [RFC2863].
ifInOctets	The number of received octets over the interface; such as the number of octets received as 802.11 frames.
ifOutOctets	The number of transmitted octets over the interface; such as the number of octets transmitted as 802.11 frames.
ifInErrors	The number of inbound frames dropped due to uncorrectable errors.
ifInUnknownProtos	The number of received frame discarded during frame header validation, including frames with unrecognized label values.
ifOutErrors	The number of outbound frames dropped due to uncorrectable errors.

[7.3.](#) Relationship to the ENTITY-MIB

The ENTITY-MIB [[RFC4133](#)] meets need for a standardized way of representing a single agent, which supports multiple instances of one MIB. It could express a certain relationship between multiple entities, and provide entity properties for each entity.

Under the wireless centralized architectures, the SNMP agent will run on the AC, and not required on the WTP. By the ENTITY-MIB on the AC, it could keep entity information such as firmware revision and software revision of AC and WTPs. From the ENTITY-MIB perspective, the overall physical entity (AC) is a 'compound' of multiple physical entities (WTPs which connects to AC), all entities are identified by Physical index. In the capwapBaseWtpTable of CAPWAP-BASE-MIB, it uses capwapBaseWtpPhyIndex object to keep the mapping relationship of

WTP object between CAPWAP-BASE-MIB and ENTITY-MIB.

By combining the MIBs, operators could query AC and WTP's status and properties. For example, they could get WTP's current status through CAPWAP-BASE-MIB, and WTP's software revision information through ENTITY-MIB.

7.4. Relationship to MIB standards of other SDOs

The MIB standards (such as IEEE 802.11 MIB) of a wireless binding is required to support on the AC. The CAPWAP-BASE-MIB module is able to support any wireless binding technology. Through ifIndex of 'WTP Virtual Radio Interface' ifType, it provides consistent and abstract way of reusing MIB objects of a wireless binding technology.

7.5. MIB modules required for IMPORTS

The following MIB module IMPORTS objects from SNMPv2-SMI [[RFC2578](#)], SNMPv2-TC [[RFC2579](#)], SNMPv2-CONF [[RFC2580](#)], IF-MIB [[RFC2863](#)], INET-ADDRESS-MIB [[RFC4001](#)] and ENTITY-MIB [[RFC4133](#)].

8. Example of CAPWAP-BASE-MIB Usage

With the idea of 'WTP Virtual Radio Interface' in the mind, the usage of MIB will be easily understood. Here takes IEEE 802.11 binding technology as an example.

1) Identify each PHY radio by 'WTP Virtual Radio Interface'

According to [[I-D.ietf-capwap-protocol-specification](#)], each radio on a WTP will be identified by a radio Id. Each WTP could be identified by its serial number.

When configuration for a WTP is prepared before a WTP connects to AC, the following information is available in the CapwapBaseRadioBindTable.

In CapwapBaseRadioBindTable

```
{
    capwapBaseWtpId           = 12345678,
    capwapBaseRadioId         = 1,
    capwapBaseWtpVirtualRadioifIndex = 10,
    capwapBaseRadioWirelessBinding = dot11(2)
}
```

Suppose WTP's serial number is 12345678, and first radio id is 1. On the AC, the ifIndex of interface in 'WTP Virtual Radio Interface' ifType is 10 which represents the PHY radio 1.

By the mechanism of 'WTP Virtual Radio Interface', it seems WTP PHY radios are located on the AC. The interface of 'WTP Virtual Radio Interface' ifType is modeled by ifTable.

```
In ifTable
{
    ifIndex          = 10,
    ifDescr          = 'WTP Virtual Radio Interface',
    ifType            = IANAifType of 'WTP Virtual
                        Radio Interface',
    ifMtu             = 0,
    ifSpeed           = 0,
    ifPhysAddress     = 0.0.0.0.0.0,
    ifAdminStatus     = true,
    ifOperStatus      = false,
    ifLastChange      = 0,
    ifInOctets        = 0,
    ifInUcastPkts     = 0,
    ifInDiscards      = 0,
    ifInErrors        = 0,
    ifInUnknownProtos = 0,
    ifOutOctets       = 0,
    ifOutUcastPkts    = 0,
    ifOutDiscards     = 0,
    ifOutErrors       = 0
}
```

2) Configure specific wireless binding parameters for 'WTP Virtual Radio Interface'

It will be done on the AC through specific wireless binding MIB such as IEEE 802.11 MIB.

For example, to configure parameter for 'WTP Virtual Radio Interface' by 802.11 Dot11OperationTable.

```
In Dot11OperationTable
{
    ifIndex          = 10,
    dot11MACAddress  = 0.0.0.0.0.0,
    dot11RTSThreshold = 2347,
    dot11ShortRetryLimit = 7,
    dot11LongRetryLimit = 4,
    dot11FragmentationThreshold = 256,
    dot11MaxTransmitMSDULifetime = 512,
    dot11MaxReceiveLifetime = 512,
    dot11ManufacturerID = 'capwap',
    dot11ProductID = 'capwap'
}
```


In the example, it suppose ifIndex of an interface in 'WTP Virtual Radio Interface' ifType is 10.

3) Other configurations for a specific wireless binding

For example, WLAN service configuration will be done through CAPWAP binding MIB and IEEE 802.11 MIB. In the CAPWAP 802.11 binding MIB draft, it will give more explain.

4) WTP reports its current configuration status

After join phase and before WTP get configuration from AC, it will report its current configuration status to AC through configuration status request message. The data of MIB objects will be updated on the AC. For example, for 802.11 binding, WTP will update data in the ifTable and IEEE 802.11 MIB so on according to message content. As an example for ifIndex 10, its ifOperStatus in ifTable will be updated according to current radio operational status in the CAPWAP message.

5) Query WTP and radio statistics data

After WTPs come to run status, operator could query WTP and radio statistics data through CAPWAP-BASE-MIB and specific binding MIB. For example, through dot11CountersTable in the IEEE 802.11 MIB, operator could query counter data for radio which is identified by ifIndex of a virtual radio interface. With capwapBaseAcState table in the MIB, operator could query configuration and properties of WTPs which are in run status.

6) Query other statistics data of a specific wireless binding

For example, operator could query the statistics data of WLAN service through 802.11 binding MIB and IEEE 802.11 MIB. In the CAPWAP 802.11 binding MIB draft, it will give more explain.

7) Query other properties of WTP

Operator could query MIB objects in the ENTITY-MIB by capwapBaseWtpPhyIndex in the capwapBaseWtpTable of CAPWAP-BASE-MIB. The properties of WTP such as software version, hardware version and so on are available in the ENTITY-MIB.

9. Definitions

CAPWAP-BASE-MIB DEFINITIONS ::= BEGIN

IMPORTS

PhysAddress, TEXTUAL-CONVENTION, TruthValue,
DateAndTime
FROM SNMPv2-TC

InterfaceIndex
FROM IF-MIB
PhysicalIndex
FROM ENTITY-MIB
SnmpAdminString
FROM SNMP-FRAMEWORK-MIB
NOTIFICATION-GROUP, OBJECT-GROUP, MODULE-COMPLIANCE
FROM SNMPv2-CONF
MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, mib-2,
Integer32, Unsigned32, Counter32
FROM SNMPv2-SMI
InetAddressType, InetAddress
FROM INET-ADDRESS-MIB;

capwapBaseMIB MODULE-IDENTITY

LAST-UPDATED "200810110000Z" -- Oct 11th, 2008
ORGANIZATION "IETF Control And Provisioning of Wireless Access
Points (CAPWAP) Working Group
<http://www.ietf.org/html.charters/capwap-charter.html>"

CONTACT-INFO

"General Discussion: capwap@frascone.com
To Subscribe: <http://lists.frascone.com/mailman/listinfo/capwap>

Yang Shi
H3C, Digital Technology Plaza, NO.9 Shangdi 9th Street, Haidian
District, Beijing, China(100085)
Email: young@h3c.com

David T. Perkins
228 Bayview Dr
San Carlos, CA 94070
USA
Phone: +1 408 394-8702
Email: dperkins@snmpinfo.com

Chris Elliott
Cisco Systems, Inc.
7025 Kit Creek Rd., P.O. Box 14987
Research Triangle Park 27709
USA
Phone: +1 919-392-2146
Email: chelliot@cisco.com

Yong Zhang
Fortinet, Inc.
1090 Kifer Road
Sunnyvale, CA 94086
USA

Email: yzhang@fortinet.com"

DESCRIPTION

"Copyright (C) 2008 The Internet Society. This version of the MIB module is part of RFC xxx; see the RFC itself for full legal notices.

This MIB module contains managed object definitions for the CAPWAP Protocol."

REVISION "200810110000Z"

DESCRIPTION

"Initial version published as RFC xxx"
::= { mib-2 xxx }

-- Textual Conventions

CapwapBaseWtpIdTC ::= TEXTUAL-CONVENTION

DISPLAY-HINT "128a"

STATUS current

DESCRIPTION

"Represents an unique identifier of a WTP instance.
As usual, a serial number of WTP will be used."

SYNTAX OCTET STRING(SIZE(128))

CapwapBaseStationIdTC ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1x:"

STATUS current

DESCRIPTION

"Represents an unique identifier of a station instance.
As usual, the MAC address of station will be used."

SYNTAX OCTET STRING (SIZE (6))

CapwapBaseRadioIdTC ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"Represents an unique identifier of a radio on a WTP."

SYNTAX Unsigned32 (1..31)

CapwapBaseTunnelModeTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents the tunneling modes of operation that are supported by the WTP.

The WTP MAY support more than one option, represented by the bit field below.

unused(0) - This bit is unused
localBridging(1) - Local Bridging Mode,

dot3Tunnel(2) - 802.3 Frame Tunnel Mode,
nativeTunnel(3) - Native Frame Tunnel Mode."

REFERENCE

"[Section 4.6.42](#). of CAPWAP Protocol Specification, RFC xxx."

SYNTAX BITS { unused(0),
localBridging(1),
dot3Tunnel(2),
nativeTunnel(3) }

CapwapBaseMacTypeTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents the MAC mode of operation supported by the WTP.

The following enumerated values are supported:

localMAC(0) - Local-MAC Mode,
splitMAC(1) - Split-MAC Mode,
both(2) - Both Local-MAC and Split-MAC."

REFERENCE

"[Section 4.6.43](#). of CAPWAP Protocol Specification, RFC xxx."

SYNTAX INTEGER { localMAC(0), splitMAC(1), both(2) }

CapwapBaseChannelTypeTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents the channel type for CAPWAP protocol.

The following enumerated values are supported:

data(1) - data Channel
control(2) - control Channel."

SYNTAX INTEGER { data(1), control(2) }

CapwapBaseAuthenMethodTC ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Represents the authentication credential type
for WTP.

The following enumerated values are supported:

other(1) - Other method, for example, vendor specific.
clear(2) - Clear text and no authentication,
x509(3) - X.509 Certificate authentication,
psk(4) - Pre-Shared Secret authentication,

As mandatory requirement, CAPWAP control channel
authentication should use DTLS, and either by certificate or
PSK. For data channel, DTLS is optional."

SYNTAX INTEGER { other(1), clear(2), x509(3), psk(4) }

-- Top level components of this MIB

-- Notifications


```
capwapBaseNotifications OBJECT IDENTIFIER
    ::= { capwapBaseMIB 0 }
-- Tables, Scalars
capwapBaseObjects OBJECT IDENTIFIER
    ::= { capwapBaseMIB 1 }
-- Conformance
capwapBaseConformance OBJECT IDENTIFIER
    ::= { capwapBaseMIB 2 }

-- AC Objects Group

capwapBaseAc OBJECT IDENTIFIER
    ::= { capwapBaseObjects 1 }

capwapBaseWtpSessions OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Represents the total number of WTPs which are connecting to
        AC."
    REFERENCE
        "Section 4.6.1. of CAPWAP Protocol Specification, RFC xxx."
    ::= { capwapBaseAc 1 }

capwapBaseWtpSessionsLimit OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Represents the maximum number of WTP sessions supported by
        the AC."
    REFERENCE
        "Section 4.6.1. of CAPWAP Protocol Specification, RFC xxx."
    ::= { capwapBaseAc 2 }

capwapBaseStationSessions OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Represents the total number of stations which are accessing
        the wireless service provided by the AC."
    REFERENCE
        "Section 4.6.1. of CAPWAP Protocol Specification, RFC xxx."
    ::= { capwapBaseAc 3 }
```


capwapBaseStationSessionsLimit OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the maximum number of station sessions supported by the AC."

REFERENCE

"[Section 4.6.1](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAc 4 }

capwapBaseDataChannelSecOptions OBJECT-TYPE

SYNTAX BITS { other(0), clear(1), dtls(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the security policy supported for CAPWAP data channel.

The AC MAY support more than one option, represented by the bit field below.

other(0) - Other method, for example, vendor specific.

clear(1) - Clear Text,

dtls(2) - DTLS."

REFERENCE

"[Section 4.6.1](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAc 5 }

capwapBaseWtpAuthenOptions OBJECT-TYPE

SYNTAX BITS { unused(0), x509(1), psk(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the authentication credential type supported by the AC for control channel.

The AC MAY support more than one option, represented by the bit field below.

unused(0) - This bit is unused

x509(1) - X.509 Certificate Based

psk(2) - Pre-Shared Secret."

REFERENCE

"[Section 4.6.1](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAc 6 }

capwapBaseWtpFallbackEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents whether to enable or disable automatic CAPWAP fallback in the event that a WTP detects its preferred AC, and is not currently connected to it."

REFERENCE

"[Section 4.6.42](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAc 7 }

-- capwapBaseAcNameListTable table

capwapBaseAcNameListTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseAcNameListEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of objects that display AC name list."

::= { capwapBaseAc 8 }

capwapBaseAcNameListEntry OBJECT-TYPE

SYNTAX CapwapBaseAcNameListEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of objects that display AC name list."

INDEX { capwapBaseAcNameListId }

::= { capwapBaseAcNameListTable 1 }

CapwapBaseAcNameListEntry ::= SEQUENCE {

capwapBaseAcNameListId Unsigned32,

capwapBaseAcNameListName OCTET STRING,

capwapBaseAcNameListPriority Unsigned32 }

capwapBaseAcNameListId OBJECT-TYPE

SYNTAX Unsigned32 (1..255)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Represents a unique id for a AC Name list."

::= { capwapBaseAcNameListEntry 1 }

capwapBaseAcNameListName OBJECT-TYPE

SYNTAX OCTET STRING(SIZE(512))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the name of an AC."

REFERENCE

"[Section 4.6.5](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAcNameListEntry 2 }

capwapBaseAcNameListPriority OBJECT-TYPE

SYNTAX Unsigned32 (1..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the priority order of the preferred AC.

For instance, the value of one (1) is used to set the primary AC, the value of two (2) is used to set the secondary, etc."

REFERENCE

["Section 4.6.5.](#) of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseAcNameListEntry 3 }

-- End of AC Objects Group

-- WTP Objects Group

capwapBaseWtps OBJECT IDENTIFIER

::= { capwapBaseObjects 2 }

-- capwapBaseWtpStateTable table

capwapBaseWtpStateTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseWtpStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of objects that display WTP CAPWAP FSM state."

::= { capwapBaseWtps 1 }

capwapBaseWtpStateEntry OBJECT-TYPE

SYNTAX CapwapBaseWtpStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of objects that display WTP CAPWAP FSM state."

INDEX { capwapBaseWtpId }

::= { capwapBaseWtpStateTable 1 }

CapwapBaseWtpStateEntry ::= SEQUENCE {

capwapBaseWtpId CapwapBaseWtpIdTC,

capwapBaseWtpIpAddressType InetAddressType,

capwapBaseWtpIpAddress InetAddress,

capwapBaseWtpPhyAddress PhysAddress,

capwapBaseWtpState INTEGER }

capwapBaseWtpId OBJECT-TYPE

SYNTAX CapwapBaseWtpIdTC

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Represents the unique identifier of a WTP."
::= { capwapBaseWtpStateEntry 1 }

capwapBaseWtpIpAddressType OBJECT-TYPE

SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the type of IP address of a WTP."
::= { capwapBaseWtpStateEntry 2 }

capwapBaseWtpIpAddress OBJECT-TYPE

SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the IP address of a WTP."
::= { capwapBaseWtpStateEntry 3 }

capwapBaseWtpPhyAddress OBJECT-TYPE

SYNTAX PhysAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the PHY address of a WTP."
::= { capwapBaseWtpStateEntry 4 }

capwapBaseWtpState OBJECT-TYPE

SYNTAX INTEGER {
 dtls(1), join(2), image(3), configure(4),
 dataCheck(5), run(6), clear(7), unknown(8)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the various possible CAPWAP FSM states of WTP
 The following enumerated values are supported:
 dtls(1) - DTLS negotiation states, which include
 DTLS Setup, Authorize, DTLS Connect.
 join(2) - WTP is joining with AC,
 image(3) - WTP is downloading software,
 configure(4) - WTP is getting configuration from AC,
 dataCheck(5) - AC is waiting for the Data Channel Keep
 Alive Packet.
 run(6) - WTP comes to run state,

clear(7) - WTP recoveries default configuration.
 unknown(8) - Operator already prepare configuration
 for WTP, while WTP has not contact with AC
 till now."

REFERENCE

"[Section 2.3.1](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpStateEntry 5 }

-- End of capwapBaseWtpStateTable Table

-- capwapBaseWtpTable Table

capwapBaseWtpTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseWtpEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of objects that display and control WTPs in
 running state. Values of all read-write objects in this
 table are persistent at restart/reboot."

::= { capwapBaseWtps 2 }

capwapBaseWtpEntry OBJECT-TYPE

SYNTAX CapwapBaseWtpEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of objects that display and control a WTP in
 running state."

INDEX { capwapBaseWtpCurrId }

::= { capwapBaseWtpTable 1 }

CapwapBaseWtpEntry ::= SEQUENCE {

capwapBaseWtpCurrId	CapwapBaseWtpIdTC,
capwapBaseWtpPhyIndex	PhysicalIndex,
capwapBaseWtpName	SnmpAdminString,
capwapBaseWtpLocation	SnmpAdminString,
capwapBaseWtpBaseMacAddress	PhysAddress,
capwapBaseTunnelModeOptions	CapwapBaseTunnelModeTC,
capwapBaseMacTypeOptions	CapwapBaseMacTypeTC,
capwapBaseWtpDiscoveryType	INTEGER,
capwapBaseWtpRadiosInUseNum	Unsigned32,
capwapBaseWtpRadioNumLimit	Unsigned32,
capwapBaseWtpStaticIpEnable	TruthValue,
capwapBaseWtpStaticIpType	InetAddressType,
capwapBaseWtpStaticIp	InetAddress,
capwapBaseWtpNetmask	InetAddress,


```
capwapBaseWtpGateway      InetAddress,
capwapBaseWtpRetransmitCount Unsigned32 }
```

capwapBaseWtpCurrId OBJECT-TYPE

SYNTAX CapwapBaseWtpIdTC

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Represents the unique identifier of a WTP Which is
in running state."

::= { capwapBaseWtpEntry 1 }

capwapBaseWtpPhyIndex OBJECT-TYPE

SYNTAX PhysicalIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the unique physical index of a physical entity
in the ENTITY-MIB. The information such as software version
of specific WTP could be accessed through the index."

::= { capwapBaseWtpEntry 2 }

capwapBaseWtpName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the name of a WTP."

REFERENCE

"[Section 4.6.44](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 3 }

capwapBaseWtpLocation OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the location of a WTP."

REFERENCE

"[Section 4.6.29](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 4 }

capwapBaseWtpBaseMacAddress OBJECT-TYPE

SYNTAX PhysAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the WTP's Base MAC Address, which MAY be assigned

to the primary Ethernet interface."

REFERENCE

"[Section 4.6.39](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 5 }

capwapBaseTunnelModeOptions OBJECT-TYPE

SYNTAX CapwapBaseTunnelModeTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the tunneling modes of operation supported by the WTP.

The WTP MAY support more than one option, represented by the bit field below.

localBridging(1) - Local Bridging Mode,
dot3Tunnel(2) - 802.3 Frame Tunnel Mode,
nativeTunnel(3) - Native Frame Tunnel Mode."

REFERENCE

"[Section 4.6.42](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 6 }

capwapBaseMacTypeOptions OBJECT-TYPE

SYNTAX CapwapBaseMacTypeTC

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the MAC mode of operation supported by the WTP.

The following enumerated values are supported:

localMAC(0) - Local-MAC Mode,
splitMAC(1) - Split-MAC Mode,
both(2) - Both Local-MAC Mode and Split-MAC Mode."

REFERENCE

"[Section 4.6.43](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 7 }

capwapBaseWtpDiscoveryType OBJECT-TYPE

SYNTAX INTEGER {
 unknown(0), staticConfig(1), dhcp(2), dns(3), acRef(4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents how WTP discovers the AC.

The following enumerated values are supported:

unknown(0) - Unknown,
staticConfig(1) - Static Configuration,
dhcp(2) - DHCP,
dns(3) - DNS,

acRef(4) - AC Referral."

REFERENCE

"[Section 4.6.21](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 8 }

capwapBaseWtpRadiosInUseNum OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of radios in use in the WTP."

REFERENCE

"[Section 4.6.40](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 9 }

capwapBaseWtpRadioNumLimit OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the maximum radio number supported by the WTP."

REFERENCE

"[Section 4.6.40](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 10 }

capwapBaseWtpStaticIpEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents whether the WTP should use a static IP address or not. A value of false disables the static IP address, while a value of true enables it."

REFERENCE

"[Section 4.6.47](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 11 }

capwapBaseWtpStaticIpType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents whether WTP uses IPV4 or IPV6 static IP address."

::= { capwapBaseWtpEntry 12 }

capwapBaseWtpStaticIp OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When capwapBaseWtpStaticIpEnable is true, it represents the static IP address to assign to the WTP."

REFERENCE

"[Section 4.6.47](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 13 }

capwapBaseWtpNetmask OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When capwapBaseWtpStaticIpEnable is true, it represents the netmask to assign to the WTP."

REFERENCE

"[Section 4.6.47](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 14 }

capwapBaseWtpGateway OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When capwapBaseWtpStaticIpEnable is true, it represents the gateway to assign to the WTP."

REFERENCE

"[Section 4.6.47](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 15 }

capwapBaseWtpRetransmitCount OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of retransmissions for a given CAPWAP packet."

REFERENCE

"[Section 4.8.8](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpEntry 16 }

-- End of capwapBaseWtpTable table

-- capwapBaseRadioBindTable Table

capwapBaseRadioBindTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseRadioBindEntry

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "A table of objects that display the mapping relationship
 between specific interface of 'WTP Virtual Radio Interface'
 ifType and PHY radio. The mapping relationship in this table
 is persistent at restart/reboot."
 ::= { capwapBaseWtps 3 }

capwapBaseRadioBindEntry OBJECT-TYPE
SYNTAX CapwapBaseRadioBindEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "A set of objects that display the mapping relationship
 between 'WTP Virtual Radio' and PHY radio."
INDEX { capwapBaseWtpId, capwapBaseRadioId }
 ::= { capwapBaseRadioBindTable 1 }

CapwapBaseRadioBindEntry ::= SEQUENCE {
 capwapBaseRadioId CapwapBaseRadioIdTC,
 capwapBaseVirtualRadioIfIndex InterfaceIndex,
 capwapBaseRadioWirelessBinding INTEGER
}

capwapBaseRadioId OBJECT-TYPE
SYNTAX CapwapBaseRadioIdTC
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "Represents the identifier of a PHY radio on a WTP, and only
 requires unique on a WTP.
 For example, WTP A and WTP B will use same value of
 capwapBaseRadioId for their first radio."
REFERENCE
 "[Section 4.3](#). of CAPWAP Protocol Specification, RFC xxx."
 ::= { capwapBaseRadioBindEntry 1 }

capwapBaseVirtualRadioIfIndex OBJECT-TYPE
SYNTAX InterfaceIndex
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the index value that uniquely identifies a
 'WLAN Virtual Radio Interface'. The interface identified by a
 particular value of this index is the same interface as
 identified by the same value of ifIndex.
 Before WTPs connect to AC and get configuration, operator

will prepare configuration for them. On the AC, there are interfaces of 'WTP Virtual Radio Interface' ifType which represent PHY radio interfaces on the WTP.

As most MIBs use ifIndex to identify an interface for configuration and statistic data, for example, IEEE 802.11 MIB. It will be very easy to reuse other MIBs such as IEEE 802.11 MIB by 'WTP Virtual Radio Interface'."

::= { capwapBaseRadioBindEntry 2 }

capwapBaseRadioWirelessBinding OBJECT-TYPE

SYNTAX INTEGER { reserved0(0), dot11(1), reserved2(2), epc(3) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the wireless binding type for radio.

The following enumerated values are supported:

reserved0(0) - Reserved,
dot11(1) - IEEE 802.11,
reserved2(2) - Reserved,
epc(3) - EPCGlobal."

REFERENCE

"[Section 4.3](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseRadioBindEntry 3 }

-- End of capwapBaseRadioBindTable Table

-- capwapBaseStationTable Table

capwapBaseStationTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseStationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of objects that display stations which are accessing the wireless service provided by the AC."

::= { capwapBaseWtps 4 }

capwapBaseStationEntry OBJECT-TYPE

SYNTAX CapwapBaseStationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of objects that display a station which is associated with the specific radio on the WTP."

INDEX { capwapBaseWtpCurrId,
capwapBaseRadioId,
capwapBaseStationId }


```
::= { capwapBaseStationTable 1 }
```

```
CapwapBaseStationEntry ::= SEQUENCE {  
    capwapBaseStationId          CapwapBaseStationIdTC,  
    capwapBaseStationAddedTime   DateAndTime,  
    capwapBaseStationVlanName    SnmpAdminString  
}
```

```
capwapBaseStationId OBJECT-TYPE  
    SYNTAX          CapwapBaseStationIdTC  
    MAX-ACCESS      not-accessible  
    STATUS          current  
    DESCRIPTION  
        "Represents the unique identifier of the station."  
    REFERENCE  
        "Section 4.6.8. of CAPWAP Protocol Specification, RFC xxx."  
    ::= { capwapBaseStationEntry 1 }
```

```
capwapBaseStationAddedTime OBJECT-TYPE  
    SYNTAX          DateAndTime  
    MAX-ACCESS      read-only  
    STATUS          current  
    DESCRIPTION  
        "Represents the time when the station is added."  
    REFERENCE  
        "Section 4.6.8. of CAPWAP Protocol Specification, RFC xxx."  
    ::= { capwapBaseStationEntry 2 }
```

```
capwapBaseStationVlanName OBJECT-TYPE  
    SYNTAX          SnmpAdminString  
    MAX-ACCESS      read-only  
    STATUS          current  
    DESCRIPTION  
        "Represents VLAN name to which the station is associated."  
    REFERENCE  
        "Section 4.6.8. of CAPWAP Protocol Specification, RFC xxx."  
    ::= { capwapBaseStationEntry 3 }
```

```
-- End of capwapBaseStationTable Table
```

```
-- capwapBaseWtpRebootStatTable
```

```
capwapBaseWtpRebootStatsTable OBJECT-TYPE  
    SYNTAX          SEQUENCE OF CapwapBaseWtpRebootStatsEntry  
    MAX-ACCESS      not-accessible  
    STATUS          current  
    DESCRIPTION
```


"A table of objects that display WTPs' reboot statistics."
REFERENCE
"Section 4.6.46. of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseWtps 5 }

capwapBaseWtpRebootStatsEntry OBJECT-TYPE

SYNTAX CapwapBaseWtpRebootStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of objects that display the reboot statistic data
of a WTP."

REFERENCE

"Section 4.6.46. of CAPWAP Protocol Specification, RFC xxx."

INDEX { capwapBaseWtpCurrId }

::= { capwapBaseWtpRebootStatsTable 1 }

CapwapBaseWtpRebootStatsEntry ::= SEQUENCE {

capwapBaseWtpRebootCount Counter32,

capwapBaseWtpInitCount Counter32,

capwapBaseWtpLinkFailureCount Counter32,

capwapBaseWtpSwFailureCount Counter32,

capwapBaseWtpHwFailureCount Counter32,

capwapBaseWtpOtherFailureCount Counter32,

capwapBaseWtpUnknownFailureCount Counter32,

capwapBaseWtpLastFailureType INTEGER

}

capwapBaseWtpRebootCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of reboots that have occurred due to a
WTP crash. A value of 65535 implies that this information is
not available on the WTP."

REFERENCE

"Section 4.6.46. of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 1 }

capwapBaseWtpInitCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of reboots that have occurred at the
request of a CAPWAP protocol message, such as a change in
configuration that required a reboot or an explicit CAPWAP

protocol reset request. A value of 65535 implies that this information is not available on the WTP."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 2 }

capwapBaseWtpLinkFailureCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that a CAPWAP protocol connection with an AC has failed due to link failure."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 3 }

capwapBaseWtpSwFailureCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that a CAPWAP protocol connection with an AC has failed due to software related reasons."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 4 }

capwapBaseWtpHwFailureCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that a CAPWAP protocol connection with an AC has failed due to hardware related reasons."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 5 }

capwapBaseWtpOtherFailureCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that a CAPWAP protocol connection with an AC has failed due to known reasons, other

than AC initiated, link, software or hardware failure."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 6 }

capwapBaseWtpUnknownFailureCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that a CAPWAP protocol connection with an AC has failed for unknown reasons."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 7 }

capwapBaseWtpLastFailureType OBJECT-TYPE

SYNTAX INTEGER { notSupport(0),
acInit(1),
linkFailure(2),
swFailure(3),
hwFailure(4),
otherFailure(5),
unknown(255)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the failure type of the most recent WTP failure. The following enumerated values are supported:

notSupport(0) - Not Supported,
acInit(1) - AC Initiated,
linkFailure(2) - Link Failure,
swFailure(3) - Software Failure,
hwFailure(4) - Hardware Failure,
otherFailure(5) - Other Failure,
unknown(255) - Unknown (e.g., WTP doesn't keep track of info)."

REFERENCE

"[Section 4.6.46](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseWtpRebootStatsEntry 8 }

-- End of capwapBaseWtpRebootStatsTable table

-- capwapBaseRadioStatsTable table

capwapBaseRadioStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF CapwapBaseRadioStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "A table of objects that display statistics on radios behavior,
 and reasons why the WTP radio has been reset."
REFERENCE
 "[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseWtps 6 }

capwapBaseRadioStatsEntry OBJECT-TYPE
SYNTAX CapwapBaseRadioStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "A set of objects that display the statistic data of
 events happened on a specific radio of a WTP."
INDEX { capwapBaseWtpCurrId, capwapBaseRadioId }
::= { capwapBaseRadioStatsTable 1 }

CapwapBaseRadioStatsEntry ::= SEQUENCE {
 capwapBaseRadioResetCount Counter32,
 capwapBaseRadioSwFailCount Counter32,
 capwapBaseRadioHwFailCount Counter32,
 capwapBaseRadioOtherFailCount Counter32,
 capwapBaseRadioUnknownFailCount Counter32,
 capwapBaseRadioConfigUpdateCount Counter32,
 capwapBaseRadioChannelChangeCount Counter32,
 capwapBaseRadioBandChangeCount Counter32,
 capwapBaseRadioCurrentNoiseFloor Integer32,
 capwapBaseRadioDecryptErrorCount Counter32,
 capwapBaseRadioLastFailType INTEGER
}

capwapBaseRadioResetCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Represents the number of times that that the radio has been
 reset."
REFERENCE
 "[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseRadioStatsEntry 1 }

capwapBaseRadioSwFailCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio has failed due to software related reasons."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseRadioStatsEntry 2 }

capwapBaseRadioHwFailCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio has failed due to hardware related reasons."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseRadioStatsEntry 3 }

capwapBaseRadioOtherFailCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio has failed due to known reasons, other than software or hardware failure."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseRadioStatsEntry 4 }

capwapBaseRadioUnknownFailCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio has failed for unknown reasons."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseRadioStatsEntry 5 }

capwapBaseRadioConfigUpdateCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio configuration has been updated."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseRadioStatsEntry 6 }

capwapBaseRadioChannelChangeCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio channel has been changed."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseRadioStatsEntry 7 }

capwapBaseRadioBandChangeCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of times that the radio has changed frequency bands."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseRadioStatsEntry 8 }

capwapBaseRadioCurrentNoiseFloor OBJECT-TYPE

SYNTAX Integer32

UNITS "dBm"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the noise floor of the radio receiver in units of dBm."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseRadioStatsEntry 9 }

capwapBaseRadioDecryptErrorCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of decryption errors that occurred on the WTP. Note that this field is only valid in cases where the WTP provides encryption/decryption services."

REFERENCE

"[Section 4.6.45](#). of CAPWAP Protocol Specification, RFC xxx."


```
::= { capwapBaseRadioStatsEntry 10 }
```

```
capwapBaseRadioLastFailType OBJECT-TYPE
```

```
SYNTAX      INTEGER {
                notSupport(0),
                swFailure(1),
                hwFailure(2),
                otherFailure(3),
                unknown(255)
            }
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Represents the failure type of the most recent radio failure.
    The following enumerated values are supported:
```

```
        notSupport(0)   - Not Supported,
        swFailure(1)    - Software Failure,
        hwFailure(2)    - Hardware Failure,
        otherFailure(3) - Other Failure,
        unknown(255)    - Unknown."
```

```
REFERENCE
```

```
    "Section 4.6.45. of CAPWAP Protocol Specification, RFC xxx."
```

```
::= { capwapBaseRadioStatsEntry 11 }
```

```
-- End of capwapBaseRadioStatsTable table
```

```
-- CAPWAP Base Parameters Group
```

```
capwapBaseParameters OBJECT IDENTIFIER
```

```
::= { capwapBaseObjects 3 }
```

```
capwapBaseDataChannelDeadInterval OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS       "second"
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "Represents the minimum time, in seconds, a WTP MUST wait
    without having received a Data Channel Keep Alive packet
    before the destination for the Data Channel Keep Alive
    packets may be considered dead."
```

```
REFERENCE
```

```
    "Section 4.7.3. of CAPWAP Protocol Specification, RFC xxx."
```

```
::= { capwapBaseParameters 1 }
```

```
capwapBaseDataCheckTimer OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```


UNITS "second"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents The number of seconds the AC will wait for the Data Channel Keep Alive, which is required by the CAPWAP state machine's Data Check state.

The AC resets the state machine if this timer expires prior to transitioning to the next state."

REFERENCE

"[Section 4.7.4](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 2 }

capwapBaseMaxRetransmit OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represent the maximum number of retransmission for a given CAPWAP packet before the link layer considers the peer dead."

REFERENCE

"[Section 4.8.7](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 3 }

capwapBaseRetransmitInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "second"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the minimum time, in seconds, in which a non-acknowledged CAPWAP packet will be retransmitted."

REFERENCE

"[Section 4.7.12](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 4 }

capwapBaseReportInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "second"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the interval for WTP to send Decryption Error report."

REFERENCE

"[Section 4.7.11](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 5 }

capwapBaseEchoInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "second"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the minimum time, in seconds, between sending Echo Request messages to the AC with which the WTP has joined."

REFERENCE

"[Section 4.7.7](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 6 }

capwapBaseIdleTimeout OBJECT-TYPE

SYNTAX Unsigned32

UNITS "second"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Represents the idle timeout value that the WTP SHOULD enforce for its active stations."

REFERENCE

"[Section 4.7.8](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseParameters 7 }

-- End of CAPWAP Base Parameters Group

-- CAPWAP Statistics Group

capwapBaseStats OBJECT IDENTIFIER

::= { capwapBaseObjects 4 }

capwapBaseFailedDTLSAuthFailCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of failed DTLS session establishment attempts due to authentication failures."

REFERENCE

"[Section 4.8.3](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseStats 1 }

capwapBaseFailedDTLSSessionCount OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Represents the number of failed DTLS session establishment attempts."

REFERENCE

"[Section 4.8.4](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseStats 2 }

-- Notifications

capwapBaseChannelUp NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId,
capwapBaseNtfChannelType,
capwapBaseNtfAuthenMethod }

STATUS current

DESCRIPTION

"This notification is sent by AC when a CAPWAP channel established. The notification is separated for data or control channel."

::= { capwapBaseNotifications 1 }

capwapBaseChannelDown NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId,
capwapBaseNtfChannelType,
capwapBaseNtfChannelDownReason }

STATUS current

DESCRIPTION

"This notification is sent by AC when CAPWAP channel becomes down. The notification is separated for data or control channel."

::= { capwapBaseNotifications 2 }

capwapBaseDecryptErrorReport NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId,
capwapBaseNtfRadioId,
capwapBaseNtfStationIdList }

STATUS current

DESCRIPTION

"This notification is generated when a WTP that has occurred decryption error since the last report."

REFERENCE

"[Section 4.6.17](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifications 3 }

capwapBaseJoinFailure NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId, capwapBaseNtfJoinFailureReason }

STATUS current

DESCRIPTION

"This notification is generated when a WTP fails to join."

REFERENCE

"[Section 4.6.34](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseNotifications 4 }

capwapBaseImageUpgradeFailure NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId, capwapBaseNtfImageFailureReason }
STATUS current
DESCRIPTION
"This notification is generated when a WTP fails to update
firmware image."
REFERENCE
"[Section 4.6.34](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseNotifications 5 }

capwapBaseConfigMsgError NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId, capwapBaseNtfConfigMsgErrorType,
capwapBaseNtfMsgErrorElements }
STATUS current
DESCRIPTION
"This notification is generated when a WTP received message
elements in the configuration management messages which it was
unable to apply locally."
REFERENCE
"[Section 4.6.34](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseNotifications 6 }

capwapBaseRadioOperableStatus NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId,
capwapBaseNtfRadioId,
capwapBaseNtfRadioOperStatusFlag,
capwapBaseNtfRadioStatusCause }
STATUS current
DESCRIPTION
"The notification is generated when a radio's operational state
is changed."
REFERENCE
"[Section 4.6.33](#). of CAPWAP Protocol Specification, RFC xxx."
::= { capwapBaseNotifications 7 }

capwapBaseAuthenticationFailure NOTIFICATION-TYPE

OBJECTS { capwapBaseNtfWtpId,
capwapBaseNtfChannelType,
capwapBaseNtfAuthenMethod,
capwapBaseNtfAuthenFailureReason }
STATUS current
DESCRIPTION
"The notification will notify the authentication failure event,
and provides the reason for it."
::= { capwapBaseNotifications 8 }

-- Objects used only in notifications

-- for notifications

capwapBaseNotifyVarObjects OBJECT IDENTIFIER
 ::= { capwapBaseObjects 5 }

capwapBaseNtfWtpId OBJECT-TYPE

SYNTAX CapwapBaseWtpIdTC

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the unique identifier of a WTP."

::= { capwapBaseNotifyVarObjects 1 }

capwapBaseNtfRadioId OBJECT-TYPE

SYNTAX CapwapBaseRadioIdTC

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the identifier of a PHY radio on a WTP, and only requires unique on a WTP.

For example, WTP A and WTP B will use same value of capwapBaseNtfRadioId for their first radio."

REFERENCE

"[Section 4.3](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 2 }

capwapBaseNtfChannelType OBJECT-TYPE

SYNTAX CapwapBaseChannelTypeTC

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the channel type for CAPWAP protocol."

::= { capwapBaseNotifyVarObjects 3 }

capwapBaseNtfAuthenMethod OBJECT-TYPE

SYNTAX CapwapBaseAuthenMethodTC

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents authentication method for Channel."

::= { capwapBaseNotifyVarObjects 4 }

capwapBaseNtfChannelDownReason OBJECT-TYPE

SYNTAX INTEGER { timeout(1),
 rekeyfailure(2),
 acRebootWtp(3),


```
                dtlsError(4),
                maxRetransmit(5) }
MAX-ACCESS    accessible-for-notify
STATUS        current
DESCRIPTION
    "Represents the reason for Channel down.
    The following enumerated values are supported:
        timeout(1)      - The keep alive is timeout,
        rekeyfailure(2) - Rekey process is failed, channel will be
                           broken.
        acRebootWtp(3)  - AC reboot WTP.
        dtlsError(4)    - DTLS notifications: DTLSAborted,
                           DTLSReassemblyFailure, DTLSPeerDisconnect,
                           or frequent DTLSDecapFailure.
        maxRetransmit(5) - The underlying reliable transport's
                           RetransmitCount counter has reached the
                           MaxRetransmit variable"
 ::= { capwapBaseNotifyVarObjects 5 }
```

capwapBaseNtfStationIdList OBJECT-TYPE

```
SYNTAX        OCTET STRING
MAX-ACCESS    accessible-for-notify
STATUS        current
DESCRIPTION
    "Represents the list of station id which use semicolons to
    separate each station id."
REFERENCE
    "Section 4.6.8. of CAPWAP Protocol Specification, RFC xxx."
 ::= { capwapBaseNotifyVarObjects 6 }
```

capwapBaseNtfAuthenFailureReason OBJECT-TYPE

```
SYNTAX        INTEGER {
                keyMismatch(1), invalidCA(2), micError(3),
                timeout(4), unknown(8)
                }
MAX-ACCESS    accessible-for-notify
STATUS        current
DESCRIPTION
    "Represents reason for WTP authorization failure.
    The following enumerated values are supported:
        keyMismatch(1) - WTP's and AC's key is not matched,
        invalidCA(2)   - ca is not valid,
        micError(3)    - detect MIC error,
        timeout(4)     - WaitDTLS Timer is timeout,
        unknown(8)     - Unknown reason."
REFERENCE
    "Section 2.3.1. of CAPWAP Protocol Specification, RFC xxx."
 ::= { capwapBaseNotifyVarObjects 7 }
```


capwapBaseNtfRadioOperStatusFlag OBJECT-TYPE

SYNTAX INTEGER { reserved(0), operable(1), inoperable(2) }

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the operation status of a radio.

The following enumerated values are supported:

reserved(0) - Reserved

operable(1) - To indicate radio is operable,

inoperable(2) - To indicate radio is inoperable, and
capwapBaseNtfRadioStatusCause object will
give reason in details."

REFERENCE

"[Section 4.6.33](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 8 }

capwapBaseNtfRadioStatusCause OBJECT-TYPE

SYNTAX INTEGER {
normal(0), hwError(1), swError(2), adminSet(3)
}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the reason the radio is out of service.

The following enumerated values are supported:

normal(0) - Normal Status,

hwError(1) - Radio Failure,

swError(2) - Software Failure,

adminSet(3) - Administratively Set."

REFERENCE

"[Section 4.6.33](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 9 }

capwapBaseNtfJoinFailureReason OBJECT-TYPE

SYNTAX INTEGER {
unspecified(1), resDepletion(2), unknownSource(3),
incorrectData(4), sessionIdInUse(5), notSupportHw(6),
notSupportBinding(7)
}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the reason of join failure.

The following enumerated values are supported:

unspecified(1) - unspecified failure reason,

resDepletion(2) - Resource Depletion,

unknownSource(3) - Unknown Source,

incorrectData(4) - Incorrect Data,

sessionIdInUse(5) - Session ID already in use,
notSupportHw(6) - WTP Hardware not supported,
notSupportBinding(7) - Binding Not Supported."

REFERENCE

"[Section 4.6.34.](#) of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 10 }

capwapBaseNtfImageFailureReason OBJECT-TYPE

SYNTAX INTEGER {
invalidChecksum(1),
invalidLength(2),
other(3),
inStorage(4)
}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the reason of image failure.

The following enumerated values are supported:

invalidChecksum(1) - Invalid Checksum,
invalidLength(2) - Invalid Data Length,
other(3) - Other Error,
inStorage(4) - Image Already Present."

REFERENCE

"[Section 4.6.34.](#) of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 11 }

capwapBaseNtfConfigMsgErrorType OBJECT-TYPE

SYNTAX INTEGER {
unknownElement(1), unsupportedElement(2),
unknownValue(3), unsupportedValue(4)
}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the type of configuration message error.

The following enumerated values are supported:

unknownElement(1) - Unknown Message Element,
unsupportedElement(2) - Unsupported Message Element,
unknownValue(3) - Unknown Message Element Value,
unsupportedValue(4) - Unsupported Message Element Value."

REFERENCE

"[Section 4.6.35.](#) of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 12 }

capwapBaseNtfMsgErrorElements OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"Represents the message elements sent by the AC in the Configuration Status Response message that caused the error."

REFERENCE

"[Section 4.6.35](#). of CAPWAP Protocol Specification, RFC xxx."

::= { capwapBaseNotifyVarObjects 13 }

-- Module compliance

capwapBaseCompliances OBJECT IDENTIFIER

::= { capwapBaseConformance 1 }

capwapBaseGroups OBJECT IDENTIFIER

::= { capwapBaseConformance 2 }

capwapBaseCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Describes the requirements for conformance to the CAPWAP-BASE-MIB."

MODULE -- this module

MANDATORY-GROUPS { capwapBaseAcNodeGroup,
capwapBaseWtpStateGroup,
capwapBaseWtpsGroup,
capwapBaseRadiosGroup,
capwapBaseStationsGroup
}

GROUP capwapBaseAcNodeGroup2

DESCRIPTION

"The capwapBaseAcNodeGroup2 group is optional."

GROUP capwapBaseAcNameListGroup

DESCRIPTION

"The capwapBaseAcNameListGroup group is optional."

GROUP capwapBaseWtpsGroup2

DESCRIPTION

"The capwapBaseWtpsGroup2 group is optional."

GROUP capwapBaseWtpRebootStatsGroup

DESCRIPTION

"The capwapBaseWtpRebootStatsGroup group is optional."

GROUP capwapBaseRadioStatsGroup

DESCRIPTION

"The capwapBaseRadioStatsGroup group is optional."

GROUP capwapBaseParametersGroup

DESCRIPTION

"The capwapBaseParametersGroup group is optional."

GROUP capwapBaseStatsGroup

DESCRIPTION

"The capwapBaseStatsGroup group is optional."

GROUP capwapBaseNotificationGroup

DESCRIPTION

"The group capwapBaseNotificationGroup is optional."

GROUP capwapBaseNotifyVarGroup

DESCRIPTION

"The capwapBaseNotifyVarGroup group is optional.
If capwapBaseNotificationGroup is supported,
this group must be implemented."

OBJECT capwapBaseRadioWirelessBinding

SYNTAX INTEGER { reserved0(0), reserved2(2) }

DESCRIPTION

"A value other than reserved0(0), reserved2(2) need not be
supported if there is no wireless binding defined for
technologies used."

::= { capwapBaseCompliances 1 }

capwapBaseAcNodeGroup OBJECT-GROUP

OBJECTS {

capwapBaseWtpSessions,
capwapBaseWtpSessionsLimit,
capwapBaseStationSessions,
capwapBaseStationSessionsLimit

}

STATUS current

DESCRIPTION

"The collection of objects which are used to represent
basic properties for AC from CAPWAP protocol perspective."

::= { capwapBaseGroups 1 }

capwapBaseAcNodeGroup2 OBJECT-GROUP

OBJECTS {

capwapBaseDataChannelSecOptions,
capwapBaseWtpAuthenOptions,
capwapBaseWtpFallbackEnable

}

STATUS current

DESCRIPTION

"The collection of objects which are used to represent other properties such as security for AC from CAPWAP protocol perspective."

::= { capwapBaseGroups 2 }

capwapBaseAcNameListGroup OBJECT-GROUP

OBJECTS {

capwapBaseAcNameListName,
capwapBaseAcNameListPriority

}

STATUS current

DESCRIPTION

"The collection of objects which are used to represent AC name list."

::= { capwapBaseGroups 3 }

capwapBaseWtpStateGroup OBJECT-GROUP

OBJECTS {

capwapBaseWtpIpAddressType,
capwapBaseWtpIpAddress,
capwapBaseWtpPhyAddress,
capwapBaseWtpState

}

STATUS current

DESCRIPTION

"The collection of objects which are used to represent WTP state information."

::= { capwapBaseGroups 4 }

capwapBaseWtpsGroup OBJECT-GROUP

OBJECTS {

capwapBaseWtpName,
capwapBaseWtpLocation,
capwapBaseWtpBaseMacAddress,
capwapBaseTunnelModeOptions,
capwapBaseMacTypeOptions,
capwapBaseWtpRadiosInUseNum,
capwapBaseWtpRadioNumLimit

}

STATUS current

DESCRIPTION

"The collection of objects which are used to represent configuration and properties information for WTP in running state."

::= { capwapBaseGroups 5 }

capwapBaseWtpsGroup2 OBJECT-GROUP


```
OBJECTS {
    capwapBaseWtpPhyIndex,
    capwapBaseWtpDiscoveryType,
    capwapBaseWtpStaticIpEnable,
    capwapBaseWtpStaticIpType,
    capwapBaseWtpStaticIp,
    capwapBaseWtpNetmask,
    capwapBaseWtpGateway
}
STATUS current
DESCRIPTION
    "The collection of objects which are used to represent
    configuration and properties information for WTP
    in running state."
::= { capwapBaseGroups 6 }
```

```
capwapBaseRadiosGroup    OBJECT-GROUP
    OBJECTS {
        capwapBaseVirtualRadioIfIndex,
        capwapBaseRadioWirelessBinding
    }
    STATUS current
    DESCRIPTION
        "The collection of objects which are used to represent
        wireless binding type, the mapping relationship between
        'WLAN Virtual Radio Interface' and PHY radio."
    ::= { capwapBaseGroups 7 }
```

```
capwapBaseStationsGroup  OBJECT-GROUP
    OBJECTS {
        capwapBaseStationAddedTime,
        capwapBaseStationVlanName
    }
    STATUS current
    DESCRIPTION
        "The collection of objects which are used to represent
        stations' basic property."
    ::= { capwapBaseGroups 8 }
```

```
capwapBaseWtpRebootStatsGroup  OBJECT-GROUP
    OBJECTS {
        capwapBaseWtpRebootCount,
        capwapBaseWtpInitCount,
        capwapBaseWtpLinkFailureCount,
        capwapBaseWtpSwFailureCount,
        capwapBaseWtpHwFailureCount,
        capwapBaseWtpOtherFailureCount,
        capwapBaseWtpUnknownFailureCount,
```



```
    capwapBaseWtpLastFailureType
}
```

```
STATUS    current
```

```
DESCRIPTION
```

```
    "The collection of objects which are used for collecting
    WTP reboot count, link failure count, hardware failure
    count and so on."
```

```
::= { capwapBaseGroups 9 }
```

```
capwapBaseRadioStatsGroup    OBJECT-GROUP
```

```
OBJECTS {
```

```
    capwapBaseRadioResetCount,
    capwapBaseRadioSwFailCount,
    capwapBaseRadioHwFailCount,
    capwapBaseRadioOtherFailCount,
    capwapBaseRadioUnknownFailCount,
    capwapBaseRadioConfigUpdateCount,
    capwapBaseRadioChannelChangeCount,
    capwapBaseRadioBandChangeCount,
    capwapBaseRadioCurrentNoiseFloor,
    capwapBaseRadioDecryptErrorCount,
    capwapBaseRadioLastFailType
```

```
}
```

```
STATUS    current
```

```
DESCRIPTION
```

```
    "The collection of objects which are used for collecting
    radio reset count, channel change count, hardware failure
    count and so on"
```

```
::= { capwapBaseGroups 10 }
```

```
capwapBaseParametersGroup    OBJECT-GROUP
```

```
OBJECTS {
```

```
    capwapBaseDataChannelDeadInterval,
    capwapBaseDataCheckTimer,
    capwapBaseRetransmitInterval,
    capwapBaseMaxRetransmit,
    capwapBaseEchoInterval,
    capwapBaseIdleTimeout,
    capwapBaseReportInterval
```

```
}
```

```
STATUS    current
```

```
DESCRIPTION
```

```
    "Objects used for CAPWAP protocol."
```

```
::= { capwapBaseGroups 11 }
```

```
capwapBaseStatsGroup    OBJECT-GROUP
```

```
OBJECTS {
```

```
    capwapBaseWtpRetransmitCount,
```



```
        capwapBaseFailedDTLSAuthFailCount,
        capwapBaseFailedDTLSSessionCount
    }
    STATUS current
    DESCRIPTION
        "Objects used for CAPWAP protocol."
    ::= { capwapBaseGroups 12 }
```

```
capwapBaseNotificationGroup    NOTIFICATION-GROUP
    NOTIFICATIONS {
        capwapBaseChannelUp,
        capwapBaseChannelDown,
        capwapBaseDecryptErrorReport,
        capwapBaseJoinFailure,
        capwapBaseImageUpgradeFailure,
        capwapBaseConfigMsgError,
        capwapBaseRadioOperableStatus,
        capwapBaseAuthenticationFailure
    }
    STATUS current
    DESCRIPTION
        "Collection of notifications in this MIB."
    ::= { capwapBaseGroups 13 }
```

```
capwapBaseNotifyVarGroup      OBJECT-GROUP
    OBJECTS {
        capwapBaseNtfWtpId,
        capwapBaseNtfRadioId,
        capwapBaseNtfChannelType,
        capwapBaseNtfAuthenMethod,
        capwapBaseNtfChannelDownReason,
        capwapBaseNtfStationIdList,
        capwapBaseNtfAuthenFailureReason,
        capwapBaseNtfRadioOperStatusFlag,
        capwapBaseNtfRadioStatusCause,
        capwapBaseNtfJoinFailureReason,
        capwapBaseNtfImageFailureReason,
        capwapBaseNtfConfigMsgErrorType,
        capwapBaseNtfMsgErrorElements
    }
    STATUS current
    DESCRIPTION
        "Objects used for notification."
    ::= { capwapBaseGroups 14 }
```

END

10. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. The followings are the tables and objects and their sensitivity/vulnerability:

- o - Unauthorized changes to the capwapBaseWtpTable, writable objects under capwapBaseAcs group may disrupt allocation of resources in the network.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. The followings are the tables and objects and their sensitivity/vulnerability:

- o - The capwapBaseWtpTable exposes WTP's important information like IP address, MAC type and so on;
- o - The capwapBaseWtpRebootStatTable exposes WTP's failure information;
- o - The capwapBaseRadioStatsTable exposes radio's failure information;

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

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

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate

rights to indeed GET or SET (change/create/delete) them.

11. IANA Considerations

11.1. IANA Considerations for CAPWAP-BASE-MIB

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
-----	-----
capwapBaseMIB	{ mib-2 XXX }

11.2. IANA Considerations for ifType

Require IANA to assign a ifType for 'WTP Virtual Radio Interface' type.

12. Contributors

This MIB is based on contributions from Long Gao.

13. Acknowledgements

The authors wish to thank David Harrington, Yu Liu, Xi Yao, Sachin Dutta, Ju Wang, Yujin Zhao, Haitao Zhang.

14. References

14.1. Normative References

- | | |
|-----------|---|
| [RFC2119] | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14 , RFC 2119 , March 1997. |
| [RFC2578] | McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578 , April 1999. |
| [RFC2579] | McCloghrie, K., Ed., Perkins, D., Ed., and J. |

- Schoenwaelder, Ed.,
"Textual Conventions for
SMIV2", STD 58, [RFC 2579](#),
April 1999.
- [RFC2580] McCloghrie, K., Perkins,
D., and J. Schoenwaelder,
"Conformance Statements for
SMIV2", STD 58, [RFC 2580](#),
April 1999.
- [RFC2863] McCloghrie, K. and F.
Kastenholz, "The Interfaces
Group MIB", [RFC 2863](#),
June 2000.
- [RFC3418] Presuhn, R., "Management
Information Base (MIB) for
the Simple Network
Management Protocol
(SNMP)", STD 62, [RFC 3418](#),
December 2002.
- [RFC4001] Daniele, M., Haberman, B.,
Routhier, S., and J.
Schoenwaelder, "Textual
Conventions for Internet
Network Addresses",
[RFC 4001](#), February 2005.
- [RFC4133] Bierman, A. and K.
McCloghrie, "Entity MIB
(Version 3)", [RFC 4133](#),
August 2005.
- [I-D.ietf-capwap-protocol-specification] Montemurro, M., Stanley,
D., and P. Calhoun, "CAPWAP
Protocol Specification", dr
aft-ietf-capwap-protocol-
specification-13 (work in
progress), September 2008.

[14.2.](#) Informative References

- [RFC3410] Case, J., Mundy, R.,
Partain, D., and B.
Stewart, "Introduction and
Applicability Statements

for Internet-Standard
Management Framework",
[RFC 3410](#), December 2002.

Authors' Addresses

Yang Shi (editor)
H3C Tech. Co., Ltd
Digital Technology Plaza, NO.9 Shangdi 9th Street, Haidian District,
Beijing
China(100085)

Phone: +86 010 82775276
EMail: young@h3c.com

David Perkins (editor)
SNMPinfo
288 Quailbrook Ct San Carlos,
CA 94070
USA

Phone: +1 408 394-8702
EMail: dperkins@snmpinfo.com

Chris Elliott (editor)
Cisco Systems, Inc.
7025 Kit Creek Rd., P.O. Box 14987 Research Triangle Park
27709
USA

Phone: +1 919-392-2146
EMail: chelliott@cisco.com

Yong Zhang (editor)
Fortinet, Inc.
1090 Kifer Road
Sunnyvale, CA 94086
USA

EMail: yzhang@fortinet.com

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

