

Internet Draft
Expires: May 2002

David Zelig
Sharon Mantin
Corrigent Systems LTD.

Thomas D. Nadeau
Cisco Systems, Inc.

Dave Danenberg
Litchfield Communications, Inc.

A. Malis
Vivace Networks, Inc.
November 2001

Pseudo Wire (PW) Management Information Base

[draft-zelig-pw-mib-01.txt](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>.

Copyright (C) The Internet Society (2001). All rights reserved.

1 Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for modeling of Pseudo Wire (PW) services on a general Packet Switched Net (PSN).

Pseudo Wire (PW) Management
Information Base

November 2001

Table of Contents

1	Abstract.....	1
2	Introduction.....	2
3	Terminology.....	3
4	The SNMP Management Framework.....	3
4.1	Object Definitions.....	4
5	Feature Checklist.....	4
6	PW-MIB usage.....	5
6.1	PW-MIB usage.....	5
6.2	Example of MIB usage.....	5
7	Object definitions.....	8
8	Security Considerations.....	32
9	References.....	33
10	Author's Addresses.....	35
11	Full Copyright Statement.....	36

[2](#) Introduction

This document describes a model for managing pseudo wire services for transmission over a packet Switched Network (PSN). This MIB module is generic and common to all types of PSN.

This document is closely related to [[FRMWK](#)], describing the transport and encapsulation of L1 and L2 services over any type of PSN and MPLS respectively, creating a Pseudo Wire (PW) service. This document describes the MIB objects that define the generic pseudo wire association to the PSN and pseudo wire configurations that are not specific to the carried service. A PW type field will be used to point to the relevant service MIB tables, described in other documents. Such document is available currently for CEP services [[CEPMIB](#)]. A similar PSN type field will point to PSN specific modules, for example [[PW MPLS MIB](#)].

Conventions used in this document

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](#) [[BCP14](#)].

Pseudo Wire (PW) Management
Information Base

November 2001

[3](#) Terminology

This document uses terminology from the document describing the PW framework [[FRMWK](#)].

"Adaptation" refers to the method of adapting a "foreign" communications protocol such that it can be carried by a packet switched net (the PSN). For example, in a CEP service the foreign protocol is SONET/SDH and the PSN is MPLS.

"Outbound" references the traffic direction where a PW payload is received, adapted to PSN, assigned a VC label, and sent into the PSN.

Conversely, "inbound" is the direction where packets are received from the PSN, packet payloads are reassembled back into the foreign service, and sent to the user as the original format.

"PSN Tunnel" is a general term indicating a virtual connection between the two PW edge devices. In practice, it is not limited to Path oriented types of PSNs like MPLS; an example is IP PSN.

Since a PW service may be bi-directional, some PW services will require two uni-directional tunnels from a uni-directional oriented PSN (MPLS for example).

[4](#) The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in [RFC 2571](#) [[RFC2571](#)].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, [RFC 1155](#) [[RFC1155](#)], STD 16, [RFC 1212](#) [[RFC1212](#)] and [RFC 1215](#)

[[RFC1215](#)]. The second version, called SMIV2, is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].

o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901](#) [[RFC1901](#)] and [RFC 1906](#) [[RFC1906](#)]. The third version of the message protocol is called SNMPv3 and described in [RFC 1906](#) [[RFC1906](#)], [RFC 2572](#) [[RFC2572](#)] and [RFC 2574](#) [[RFC2574](#)].

o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, [RFC 1157](#) [[RFC1157](#)]. A second set of protocol operations and associated PDU formats is described in [RFC 1905](#) [[RFC1905](#)].

o A set of fundamental applications described in [RFC 2573](#) [[RFC2573](#)] and the view-based access control mechanism described in [RFC 2575](#) [[RFC2575](#)].

A more detailed introduction to the current SNMP Management Framework can be found in [RFC 2570](#) [[RFC2570](#)]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI. This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

[4.1](#) Object Definitions

Managed objects are accessed via a virtual information store,

termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, an OBJECT IDENTIFIER, an administratively assigned name, names each object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to also refer to the object type.

[5](#) Feature Checklist

The PW MIB (PW-MIB) is designed to satisfy the following requirements and constraints:

- The MIB is designed to be extensible to all types of PSNs. PSNs specific MIBs will be defined in additional documents.
- The MIB supports manually configured VCs.
- The MIB supports the VC ID and Group ID.
- The MIB supports point-to-point PW connections. Point-to-multipoint connections are for future study.
- The MIB enables the use of any emulated service. Emulated service specific MIBs will be defined in additional documents.
- The MIB enables both strict and loose incoming VC lookup. In strict mode, only VC carried inside explicitly configured or

signaled tunnels are accepted in path oriented PSNs. In no path oriented PSNs, the source address of the recieved packet must be checked in strict mode.

[6](#) PW-MIB usage

The MIB structure for defining a PW service is composed from three types of modules.

The first type is the PW-MIB module, which configures general parameters of the VC that are common to all types of emulated services and PSNs.

The second type of modules is per PSN module. There is a different module for each type of PSN. These modules associate the VC with one or more "tunnels" that carry the service over the PSN. These modules are defined in other documents; see for example [[PW MPLS MIB](#)].

The third type of modules is service-specific module, which is emulated signal type dependent. These modules are defined in other documents; see for example [[CEP MIB](#)].

[PWTC] defines some of the object types used in these modules.

[6.1](#) PW-MIB usage

- The VC table (pwVcTable) is used for all VC types (ATM, FR, Ethernet, SONET, etc.). This table contains high level generic parameters related to the VC creation. A row is created by the operator for each PW service.
- Based on the PSN type defined for the VC, rows are created in PSN specific module (for example [[PW MPLS MIB](#)]) and associated to the VC table by a common VC index.
- Based on the VC type defined for the VC, rows are created in service-specific module (for example [[CEP MIB](#)]) and associated to the VC table by a common VC index.
- The MIB includes performance parameters collection common to all types of PW.

[6.2](#) Example of MIB usage

In this section we provide an example of using the MIB objects described in [section 8](#) to set up a CEP VC over MPLS PSN. While this example is not meant to illustrate every permutation of the MIB, it is intended as an aid to understanding some of the key concepts. It is meant to be read after going through the MIB itself.

In this example a PW service for CEP is configured over an MPLS PSN (MPLS-TE tunnel). It uses LDP as in [[TRANS](#)] for service set-up.

In PW-MIB:

In pwVcTable:

```
{
    pwVcIndex                5,

    pwVcType                  CEP,
    pwVcOwner                  signaling,
    pwVcPsnType                mpls,
    pwVcPriority                0,
    pwVcInboundMode            loose,

    pwVcPeerAddrType           ipv4(2),
    pwVcPeerIpv4Addr            1.4.3.2, -- In this case equal to the
                                   -- peer tunnel IP address

    pwVcPeerIpv6Addr           0,

    pwVcID                     10,
    pwVcLocalGroupID           12,

    pwVcControlWord             false, -- Control word not to be sent
    pwVcLocalIfMtu              0,      -- Do not send ifMtu parameter
    pwVcLocalIfString           false, -- Do not signal if string

    pwVcRemoteGroupID           0xFFFF, -- Will be received by
                                   -- maintenance protocol
    pwVcRemoteControlWord       notKnownYet,
    pwVcRemoteIfMtu             0,
    pwVcRemoteIfString          "",

    pwVcOutboundVcLabel         0xFFFF, -- Will be received by
                                   -- maintenance protocol
    pwVcInboundVcLabel          0xFFFF, -- Will be set by signaling

    pwVcName                    "Example of CEP VC",
    pwVcDescr                   "",
    ..

    pwVcAdminStatus             up,
    ..
    ..
}
```

The agent now create a row in pwVcMplsTable [PWMPLS] based on the VcIndex, that is configured with MPLS specific values:

In pwVcMplsTable:

Pseudo Wire (PW) Management
Information Base

November 2001

```
{
    pwVcMplsMplsType      mplsTe,
    pwVcMplsExpBitsMode   outerTunnel,
    pwVcMplsExpBits       0,
    pwVcMplsTtl           2,
    ...
}
```

The operator now associates the VC with an outgoing TE tunnel:
In pwVcMplsOutboundTable:

```
{
    pwVcMplsOutBoundLsrXcIndex      0, -- MPLS-TE
    pwVcMplsOutboundTunnelIndex     500,
    pwVcMplsOutboundTunnelInstance  0,
    pwVcMplsOutboundTunnelLclLSR    1.2.3.4, -- Always
                                     -- the LSR ID of the current node.
    pwVcMplsOutboundTunnelPeerLSR   1.4.3.2
    pwVcMplsOutboundIfIndex         0, -- MPLS-TE
    ..
}
```

pwVcMplsInboundTable is not used because loose LDP set-up is used.

pwVcMplsMappingTable entry will be created by the agent once the LDP control session will be finished and will enable easy lookup for the VcIndex from knowledge of VC label or tunnel.

Pseudo Wire (PW) Management
Information Base

November 2001

[7](#) Object definitions

```
PW-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE,  
    experimental, Integer32, Counter32, Unsigned32,  
    Counter64, TimeTicks  
    FROM SNMPv2-SMI
```

```
    MODULE-COMPLIANCE, OBJECT-GROUP  
    FROM SNMPv2-CONF
```

```
    TruthValue, RowStatus, StorageType,  
    TimeStamp  
    FROM SNMPv2-TC
```

```
    SnmpAdminString  
    FROM SNMP-FRAMEWORK-MIB
```

```
    PwVcIndex, PwGroupID, PwVcID  
    FROM PW-TC-MIB
```

```
    InetAddressIPv4, InetAddressIPv6  
    FROM INET-ADDRESS-MIB
```

```
;
```

```
pwVcMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "200111071200Z" -- 7 November 2001 12:00:00 EST  
    ORGANIZATION "Pseudo Wire Edge to Edge Emulation (PWE3) Working
```

Group"

CONTACT-INFO

"

David Zelig
Postal: Corrigent Systems
126, Yigal Alon St.
Tel Aviv, ISRAEL
Phone: +972-3-6945273
E-mail: davidz@corrigent.com

Thomas D. Nadeau
Postal: Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Tel: +1-978-244-3051
Email: tnadeau@cisco.com

Dave Danenberg
Postal: Litchfield Communications, Inc.

Zelig et al

Expires March 2002

[page 8]

Pseudo Wire (PW) Management
Information Base

November 2001

76 Westbury Park Rd
Princeton Building East
Watertown, CT 06795
Tel: +1-860-945-1573 x3180
Email: dave_danenberg@litchfieldcomm.com

Andrew G. Malis
Postal: Vivace Networks, Inc.
2730 Orchard Parkway
San Jose, CA 95134
Email: Andy.Malis@vivacenetworks.com

Sharon Mantin
Postal: Corrigent Systems
126, Yigal Alon St.
Tel Aviv, ISRAEL
Phone: +972-3-6948608
E-mail: sharonm@corrigent.com

The PWE3 Working Group (email distribution pwe3@ietf.org)

"

DESCRIPTION

"This MIB contains managed object definitions for Pseudo Wire operation as in: Pate, P., et al, <[draft-pate-pwe3-framework-01.txt](#)>, Xiao, X., et al, <[draft-ietf-pwe3-requirements-00.txt](#)>, Martini, L., et al, <[draft-martini-l2circuit-trans-mpls-06.txt](#)>, and Martini, L., et al, <[draft-martini-l2circuit-encap-mpls-02.txt](#)>.

The indexes for this MIB are also used to index the PSN-specific tables and the VC-specific tables. The VC Type dictates which VC-specific MIB to use. For example, a 'cep' VC Type requires the use the configuration and status tables within the CEP-MIB.

This MIB enable the use of any underlying packet switched network (PSN). Specific tables for the MPLS PSN is currently defined in a separate PW-MPLS-MIB. Tables to support other PSNs (IP, L2TP for example) will be added to this MIB in future revisions."

-- Revision history.

REVISION

"200111071200Z" -- 7 November 2001 12:00:00 EST

DESCRIPTION

"Changes from -00 draft:

- 1) Remove VcInstance from tables indexes in order to simplify the MIB.
- 2) Removing APS 1+1 table.
- 3) Changing hdlcCisco to hdlc in pwVcType.

- 4) Add description of vc label at pending PW signaling stage.
 - 5) Add interval valid object in interval performance tables.
 - 6) Remove VC APS notification.
 - 7) Change 'conservative'/'liberal' to 'strict'/'loose'.
 - 8) Add objects for interface MTU, use of control word, interface string.
 - 9) Ordering of objects based on functionality.
 - 10) Update operation of pwVcOperStatus.
- "

REVISION

"200107111200Z" -- 11 July 2001 12:00:00 EST

DESCRIPTION

"[draft-00](#) version"

::= { experimental xxx }

-- Top-level components of this MIB.

-- Notifications

-- Tables, Scalars

pwVcObjects OBJECT IDENTIFIER
::= { pwVcMIB 1 }

-- Conformance

pwVcConformance OBJECT IDENTIFIER
::= { pwVcMIB 2 }

-- PW Virtual Connection (VC) Table

pwVcIndexNext OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains an appropriate value to be used for pwVcIndex when creating entries in the pwVcTable. The value 0 indicates that no unassigned entries are available. To obtain the value of pwVcIndex for a new entry in the pwVcTable, the manager issues a management protocol retrieval operation to obtain the current value of pwVcIndex. After each retrieval operation, the agent should modify the value to reflect the next unassigned index. After a manager retrieves a value the agent will determine through its local policy when this index value will be made available for reuse."

::= { pwVcObjects 1 }

pwVcTable OBJECT-TYPE

SYNTAX SEQUENCE OF PwVcEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "This table specifies information for connecting various
 emulated services to various tunnel type."
 ::= { pwVcObjects 2 }

pwVcEntry OBJECT-TYPE
 SYNTAX PwVcEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "A row in this table represents an emulated virtual
 connection (VC) across a packet network. It is indexed by
 pwVcIndex, which Uniquely identifying a singular
 connection."
 INDEX { pwVcIndex }

 ::= { pwVcTable 1 }

PwVcEntry ::= SEQUENCE {

 pwVcIndex PwVcIndex,

 pwVcType INTEGER,
 pwVcOwner INTEGER,
 pwVcPsnType INTEGER,
 pwVcPriority Unsigned32,
 pwVcInboundMode INTEGER,

 pwVcPeerAddrType INTEGER,
 pwVcPeerIpv4Addr InetAddressIPv4,
 pwVcPeerIpv6Addr InetAddressIPv6,

 pwVcID PwVcID,
 pwVcLocalGroupID PwGroupID,
 pwVcControlWord TruthValue,
 pwVcLocalIfMtu Unsigned32,
 pwVcLocalIfString TruthValue,

 pwVcRemoteGroupID PwGroupID,
 pwVcRemoteControlWord INTEGER,
 pwVcRemoteIfMtu Unsigned32,
 pwVcRemoteIfString SnmpAdminString,

 pwVcOutboundVcLabel Unsigned32,
 pwVcInboundVcLabel Unsigned32,

Pseudo Wire (PW) Management
Information Base

November 2001

pwVcName	SnmpAdminString,
pwVcDescr	SnmpAdminString,
pwVcCreateTime	TimeStamp,
pwVcUpTime	TimeTicks,
pwVcAdminStatus	INTEGER,
pwVcOperStatus	INTEGER,
pwVcOutboundOperStatus	INTEGER,
pwVcInboundOperStatus	INTEGER,
pwVcRowStatus	RowStatus,
pwVcStorageType	StorageType

}

pwVcIndex OBJECT-TYPE

SYNTAX PwVcIndex
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"Index for the conceptual row identifying a VC within
this PW Emulation VC table."

::= { pwVcEntry 1 }

pwVcType OBJECT-TYPE

SYNTAX INTEGER {
frameRelay(1),
atmAal5Vcc(2),
atmTransparent(3),
ethernetVLAN(4),
ethernet(5),
hdlc(6),
ppp(7),
cep(8),
atmVccCell(9),
atmVpcCell(10),
transparentLanService(11)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This value dictates what service-specific tables are

indexed

by pwVcIndex. For example, if set for 'cep' pwVcIndex indexes the pwVcCepTable in [[CEPMIB](#)].

"

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-07](#)>.

Note: as specified in l2circuit-trans: It is REQUIRED to assign the same VC ID, and VC type for a given circuit in both directions."

Zelig et al

Expires March 2002

[page 12]

Pseudo Wire (PW) Management Information Base

November 2001

::= { pwVcEntry 2 }

pwVcOwner OBJECT-TYPE

SYNTAX INTEGER {

manual (1),

maintenanceProtocol (2), -- PW signaling

other (3)

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Set by the operator to indicate the protocol responsible for establishing this VC. Value 'manual' is used in all cases where no maintenance protocol (PW signaling) is used to set-up the VC, i.e. require configuration of entries in the VC tables including VC labels, etc. The value 'maintenanceProtocol' is used in case of standard signaling of the VC for the specific PSN, for example LDP for MPLS PSN as specified in <draft- [draft-martini-l2circuit-trans-mpls](#)> or L2TP control protocol.

Value 'other' is used for other types of signaling."

::= { pwVcEntry 3 }

pwVcPsnType OBJECT-TYPE

SYNTAX INTEGER {

mpls (1),

l2tp (2),

ip (3),

mplsOverIp (4), -- May be combined with 'mpls' in the future

gre (5),

other (6)

```

    }
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "Set by the operator to indicate the PSN type on which this
    VC will be carried. Based on this object, the relevant PSN
    table entries are created in the in the PSN specific MIB
    modules. For example, if mpls(1) is defined, the agent
    create an entry in pwVcMplsTable, which further define the
    MPLS PSN configuration."
 ::= { pwVcEntry 4 }

```

```

pwVcPriority OBJECT-TYPE
SYNTAX          Unsigned32
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "This object define the relative priority of the VC
    in a lowest-to-highest fashion, where 0 is the lowest

```

```

    priority. VCs with the same priority are treated with
    equal priority. Dropped VC will be set 'dormant' (as
    indicated in pwVcOperStatus)."
 ::= { pwVcEntry 5 }

```

```

pwVcInboundMode OBJECT-TYPE
SYNTAX          INTEGER {
    loose (1),
    strict (2)
}
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "This object is used to enable greater security for
    implementation that use global VC label space. In
    strict mode, inbound packets with are accepted only from
    tunnels that are associated to the same VC via the inbound
    tunnel table in the case of MPLS, or as identified by the
    source IP address in case of L2TP or IP PSN. The entries in
    the inbound tunnel table are either explicitly configured
    or implicitly known by the maintenance protocol used for VC
    set-up.

```


If such association is not known, not configured or not desired, loose mode should be configured, and the node should accept the packet based on the VC label only regardless of the outer tunnel used to carry the VC."

::= { pwVcEntry 6 }

pwVcPeerAddrType OBJECT-TYPE

SYNTAX INTEGER {
 ipV4(1),
 ipV6(2),
 notApplicable(3)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Denotes the address type of the peer node maintenance protocol (signaling) address if PW maintenance protocol is used for the VC creation. It should be set to notApplicable' if PE/PW maintenance protocol is not used, i.e. pwVcOwner is set to 'manual'. "

DEFVAL { ipV4 }

::= { pwVcEntry 7 }

pwVcPeerIpv4Addr OBJECT-TYPE

SYNTAX InetAddressIPv4

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If pwVcPeerAddrType is set to ipV4(1), then this value will contain the IPv4 address of the peer node PW/PE maintenance protocol entity. This object is otherwise insignificant and should contain a value of 0."

::= { pwVcEntry 8 }

pwVcPeerIpv6Addr OBJECT-TYPE

SYNTAX InetAddressIPv6

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If pwVc PeerAddrType is set to ipV6(2), then this value

will contain the Ipv6 address of the peer node PW/PE maintenance protocol entity. This object is otherwise insignificant and should contain a value of 0."

::= { pwVcEntry 9 }

pwVcID OBJECT-TYPE

SYNTAX PwVcID
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Used in the outgoing VC ID field within the 'Virtual Circuit FEC Element' when LDP signaling is used or PW ID AVP for L2TP."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>. and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

Note: as specified in l2circuit-trans: It is REQUIRED to assign the same VC ID, and VC type for a given circuit in both directions."

::= { pwVcEntry 10 }

pwVcLocalGroupID OBJECT-TYPE

SYNTAX PwGroupID
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Used in the Group ID field sent to the peer PWES within the maintenance protocol used for VC setup, zero if not used."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)> and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

::= { pwVcEntry 11 }

pwVcControlWord OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Define if the control word will be sent with each packet by the local node."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>"

DEFVAL

{ false }

::= { pwVcEntry 12 }

pwVcLocalIfMtu OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If not equal zero, the optional IfMtu object in the maintenance protocol will be sent with this value, representing the locally supported MTU size over the interface (or the virtual interface) associated with the VC."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

DEFVAL

{ 0 }

::= { pwVcEntry 13 }

pwVcLocalIfString OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Each VC is associated to an interface (or a virtual interface) in the ifTable of the node as part of the service configuration. This object defines if the maintenance protocol will send the interface's name as appears on the ifTable in the name object as part of the maintenance protocol. If set to false, the optional element will not be sent."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

DEFVAL

{ false }

::= { pwVcEntry 14 }

pwVcRemoteGroupID OBJECT-TYPE

SYNTAX PwGroupID

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Obtained from the Group ID field as received via the

Pseudo Wire (PW) Management
Information Base

November 2001

maintenance protocol used for VC setup, zero if not used.
Value of 0xFFFF shall be used if the object is yet to be
defined by the VC maintenance protocol."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

::= { pwVcEntry 15 }

pwVcRemoteControlWord OBJECT-TYPE

SYNTAX INTEGER {
noControlWord (1),
withControlWord (2),
notYetKnown(3)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"If maintenance protocol is used for VC establishment, this
parameter indicates the received status of the control word
usage, i.e. if packets will be received with control word
or not. The value of 'notYetKnown' is used while the
maintenance protocol has not yet received the indication
from the remote node.

In manual configuration of the VC this parameters indicate
to the local node what is the expected encapsulation for
the received packets.

"

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."

::= { pwVcEntry 16 }

pwVcRemoteIfMtu OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The remote interface MTU as (optionally) received from the
remote node via the maintenance protocol. Should be zero if
this parameter is not available or not used."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."
 ::= { pwVcEntry 17 }

pwVcRemoteIfString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..80))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Zelig et al

Expires March 2002

[page 17]

Pseudo Wire (PW) Management
Information Base

November 2001

"Indicate the interface description string as received by
the maintenance protocol, MUST be NULL string if not
applicable or not known yet."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>
and So, et al, <[draft-so-pwe3-ethernet-00.txt](#)>."
 ::= { pwVcEntry 18 }

pwVcOutboundVcLabel OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The VC label used in the outbound direction. It may be set
up manually if owner is 'manual' or automatically
otherwise. Examples: For MPLS PSN, it represents the 20
bits of VC tag, for L2TP it represent the 16 bits Session
ID.

If the label is not yet known (signaling in process), the
object should return a value of 0xFFFF."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>"
 ::= { pwVcEntry 19 }

pwVcInboundVcLabel OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The VC label used in the inbound direction. It may be set
up manually if owner is 'manual' or automatically
otherwise.

Examples: For MPLS PSN, it represents the 20 bits of VC tag, for L2TP it represent the 16 bits Session ID. If the label is not yet known (signaling in process), the object should return a value of 0xFFFF."

REFERENCE

"Martini, et al, <[draft-martini-l2circuit-trans-mpls-06](#)>"
::= { pwVcEntry 20 }

pwVcName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The canonical name assigned to the VC."

::= { pwVcEntry 21 }

pwVcDescr OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-create

Zelig et al

Expires March 2002

[page 18]

Pseudo Wire (PW) Management
Information Base

November 2001

STATUS current

DESCRIPTION

"A textual string containing information about the VC.
If there is no description this object contains a zero
length string."

::= { pwVcEntry 22 }

pwVcCreateTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"System time when this VC was created."

::= { pwVcEntry 23 }

pwVcUpTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of consecutive ticks this VC has been 'up' in
both directions together (i.e. 'up' is observed in

```

        pwVcOperStatus.)"
 ::= { pwVcEntry 24 }

pwVcAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                        up(1),          -- ready to pass packets
                        down(2),
                        testing(3) -- in some test mode
    }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The desired operational status of this VC."
 ::= { pwVcEntry 25 }

pwVcOperStatus OBJECT-TYPE
    SYNTAX INTEGER {
        up(1),          -- ready to pass packets
        down(2),
        testing(3),      -- in some test mode
        unknown(4),      -- status cannot be determined
        dormant(5),
        notPresent(6),    -- some component is missing
        lowerLayerDown(7) -- down due to the state of
                        -- lower layer interfaces
    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

```

"Indicates the actual combined operational status of this
VC. It is 'up' if both pwVcInboundOperStatus and
pwVcOutboundOperStatus are in 'up' state. For all other
values, if the VCs in both directions are of the same
value it reflects that value, otherwise it is set to the
most severe status out of the two statuses. The order of
severance from most severe to less severe is: unknown,
notPresent, down, lowerLayerDown, dormant, testing, up.
The operator may consult the per direction OperStatus for
fault isolation per direction."
 ::= { pwVcEntry 26 }

```

pwVcInboundOperStatus OBJECT-TYPE

```
SYNTAX INTEGER {
    up(1),                -- ready to pass packets
    down(2),
    testing(3),           -- in some test mode
    unknown(4),           -- status cannot be determined
    dormant(5),
    notPresent(6),        -- some component is missing
    lowerLayerDown(7)     -- down due to the state of
                        -- lower layer interfaces
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the actual operational status of this VC in the inbound direction.

- down: if PW signaling has not yet finished, or indications available at the service level indicate that the VC is not passing packets.
- testing: if AdminStatus at the VC level is set to test.
- dormant: The VC is not available because of the required resources are occupied VC with higher priority VCs .
- notPresent: Some component is missing to accomplish the set up of the VC.
- lowerLayerDown: The underlying PSN is not in OperStatus 'up'.

"

::= { pwVcEntry 27 }

pwVcOutboundOperStatus OBJECT-TYPE

```
SYNTAX INTEGER {
    up(1),                -- ready to pass packets
    down(2),
    testing(3),           -- in some test mode
    unknown(4),           -- status cannot be determined
}
```

```
dormant(5),
notPresent(6),    -- some component is missing
```



```

        lowerLayerDown(7) -- down due to the state of
                           -- lower layer interfaces
    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates the actual operational status of this VC in the
        outbound direction
        - down:          if PW signaling has not yet finished, or
                        indications available at the service
                        level indicate that the VC is not
                        passing packets.
        - testing:       if AdminStatus at the VC level is set to
                        test.
        - dormant:       The VC is not available because of the
                        required resources are occupied VC with
                        higher priority VCs .
        - notPresent:    Some component is missing to accomplish
                        the set up of the VC.
        - lowerLayerDown: The underlying PSN is not in OperStatus
                        'up'."
    ::= { pwVcEntry 28 }

pwVcRowStatus OBJECT-TYPE
    SYNTAX          RowStatus
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "For creating, modifying, and deleting this row."
    ::= { pwVcEntry 29 }

pwVcStorageType OBJECT-TYPE
    SYNTAX          StorageType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This variable indicates the storage type for this
        object."
    ::= { pwVcEntry 30 }

-- End of PW Virtual Connection (VC) Table

-- Vc Performance Table.

pwVcPerfCurrentTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PwVcPerfCurrentEntry
    MAX-ACCESS      not-accessible
    STATUS          current

```

Pseudo Wire (PW) Management
Information Base

November 2001

DESCRIPTION

"This table provides per-VC performance information."

::= { pwVcObjects 3 }

pwVcPerfCurrentEntry OBJECT-TYPE

SYNTAX PwVcPerfCurrentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created by the agent for every VC."

INDEX { pwVcIndex }

::= { pwVcPerfCurrentTable 1 }

PwVcPerfCurrentEntry ::= SEQUENCE {

pwVcPerfCurrentInPackets	Counter32,
pwVcPerfCurrentInHCPackets	Counter64,
pwVcPerfCurrentInBytes	Counter32,
pwVcPerfCurrentInHCBytes	Counter64,
pwVcPerfCurrentOutPackets	Counter32,
pwVcPerfCurrentOutHCPackets	Counter64,
pwVcPerfCurrentOutBytes	Counter32,
pwVcPerfCurrentOutHCBytes	Counter64

}

pwVcPerfCurrentInPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets received by the VC in the current 15 minute interval"

::= { pwVcPerfCurrentEntry 1 }

pwVcPerfCurrentInHCPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of packets received by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 2 }

pwVcPerfCurrentInBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of bytes received by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 3 }

pwVcPerfCurrentInHCBytes OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of bytes received by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 4 }

pwVcPerfCurrentOutPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets forwarded by the VC. in the current 15 minute interval"

::= { pwVcPerfCurrentEntry 5 }

pwVcPerfCurrentOutHCPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of packets forwarded by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 6 }

pwVcPerfCurrentOutBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of bytes forwarded by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 7 }

pwVcPerfCurrentOutHCBytes OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of bytes forwarded by the VC in the current 15 minute interval."

::= { pwVcPerfCurrentEntry 8 }

-- End of Vc Perf current Table

-- Vc Performance Interval Table.

pwVcPerfIntervalTable OBJECT-TYPE

Zelig et al

Expires March 2002

[page 23]

Pseudo Wire (PW) Management
Information Base

November 2001

SYNTAX SEQUENCE OF PwVcPerfIntervalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table provides per-VC performance information."

::= { pwVcObjects 4 }

pwVcPerfIntervalEntry OBJECT-TYPE

SYNTAX PwVcPerfIntervalEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table is created by the LSR for every VC. "

INDEX { pwVcIndex, pwVcPerfIntervalNumber }

::= { pwVcPerfIntervalTable 1 }

PwVcPerfIntervalEntry ::= SEQUENCE {

pwVcPerfIntervalNumber	Integer32,
pwVcPerfIntervalValidData	TruthValue,

pwVcPerfIntervalInPackets	Counter32,
---------------------------	------------

pwVcPerfIntervalInHCPackets	Counter64,
pwVcPerfIntervalInBytes	Counter32,
pwVcPerfIntervalInHCBytes	Counter64,
pwVcPerfIntervalOutPackets	Counter32,
pwVcPerfIntervalOutHCPackets	Counter64,
pwVcPerfIntervalOutBytes	Counter32,
pwVcPerfIntervalOutHCBytes	Counter64

}

pwVcPerfIntervalNumber OBJECT-TYPE

SYNTAX Integer32 (1..96)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A number N, between 1 and 96, which identifies the interval for which the set of statistics is available. The interval identified by 1 is the most recently completed 15 minute interval, and the interval identified by N is the interval immediately preceding the one identified by N-1. The minimum range of N is 1 through 4. The default range is 1 to 32. The maximum range of N is 1 through 96. "

REFERENCE

"Tesink, K. 'Definitions of Managed Objects for the SONET/SDH Interface Type', [RFC 2558](#)"

::= { pwVcPerfIntervalEntry 1 }

pwVcPerfIntervalValidData OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This variable indicates if the data for this interval is valid."

::= { pwVcPerfIntervalEntry 2 }

pwVcPerfIntervalInPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets received by the VC in a particular 15-minute interval in the past 24 hours."
 ::= { pwVcPerfIntervalEntry 3 }

pwVcPerfIntervalInHCPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of packets received by the VC in a particular 15-minute interval in the past 24 hours."

::= { pwVcPerfIntervalEntry 4 }

pwVcPerfIntervalInBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of bytes received by the VC in a particular 15-minute interval in the past 24 hours."

::= { pwVcPerfIntervalEntry 5 }

pwVcPerfIntervalInHCBytes OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of bytes received by the VC in a particular 15-minute interval in the past 24 hours."

::= { pwVcPerfIntervalEntry 6 }

pwVcPerfIntervalOutPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets forwarded by the VC in a particular 15-minute interval in the past 24 hours."

```

 ::= { pwVcPerfIntervalEntry 7 }

pwVcPerfIntervalOutHCPackets OBJECT-TYPE
    SYNTAX          Counter64
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "High capacity counter for number of packets forwarded by
         the VC in a particular 15-minute interval in the past 24
         hours."
    ::= { pwVcPerfIntervalEntry 8 }

pwVcPerfIntervalOutBytes OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Number of bytes forwarded by the VC in a particular 15-
         minute interval in the past 24 hours."
    ::= { pwVcPerfIntervalEntry 9 }

pwVcPerfIntervalOutHCBytes OBJECT-TYPE
    SYNTAX          Counter64
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "High capacity counter for number of bytes forwarded by the
         VC in a particular 15-minute interval in the past 24
         hours."

    ::= { pwVcPerfIntervalEntry 10 }

-- End of VC Performance Interval Table

-- VC Performance Total Table.

pwVcPerfTotalTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PwVcPerfTotalEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table provides per-VC Performance information."
    ::= { pwVcObjects 5 }

pwVcPerfTotalEntry OBJECT-TYPE
    SYNTAX          PwVcPerfTotalEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION

```

Pseudo Wire (PW) Management
Information Base

November 2001

"An entry in this table is created by the LSR for every VC."

```
INDEX { pwVcIndex }
 ::= { pwVcPerfTotalTable 1 }
```

```
PwVcPerfTotalEntry ::= SEQUENCE {
    pwVcPerfTotalInPackets          Counter32,
    pwVcPerfTotalInHCPackets        Counter64,
    pwVcPerfTotalInBytes            Counter32,
    pwVcPerfTotalInHCBytes          Counter64,
    pwVcPerfTotalOutPackets         Counter32,
    pwVcPerfTotalOutHCPackets       Counter64,
    pwVcPerfTotalOutBytes           Counter32,
    pwVcPerfTotalOutHCBytes         Counter64
}
```

```
pwVcPerfTotalInPackets OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Number of packets received by the VC."
    ::= { pwVcPerfTotalEntry 1 }
```

```
pwVcPerfTotalInHCPackets OBJECT-TYPE
    SYNTAX          Counter64
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "High capacity counter for number of packets received by the
        VC ."
    ::= { pwVcPerfTotalEntry 2 }
```

```
pwVcPerfTotalInBytes OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Number of bytes received by the VC."
    ::= { pwVcPerfTotalEntry 3 }
```


pwVcPerfTotalInHCBytes OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of bytes received by the VC."

::= { pwVcPerfTotalEntry 4 }

pwVcPerfTotalOutPackets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of packets forwarded by the VC."

::= { pwVcPerfTotalEntry 5 }

pwVcPerfTotalOutHCPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of packets forwarded by the VC ."

::= { pwVcPerfTotalEntry 6 }

pwVcPerfTotalOutBytes OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of bytes forwarded by the VC."

::= { pwVcPerfTotalEntry 7 }

pwVcPerfTotalOutHCBytes OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"High capacity counter for number of bytes forwarded by the VC."

```

 ::= { pwVcPerfTotalEntry 8 }

-- End of VC Perf Total Table

-- Error counter scalar

pwVcPerfTotalErrorPackets OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Counter for number of error at VC level processing, for
         example packets received with unknown VC label."
    ::= { pwVcObjects 6 }

-- Notifications - PW VCs

-- No notifications are defined. Notifications are generated in

```

```

-- the per service MIB modules and/or PSN specific modules.
-- Notification related to the maintenance protocol are FFS.

-- End of notifications.

-- conformance information

-- Note: Conformance at the object access and values level is
-- still FFS, therefore current conformance is defined at the
-- object existence level only.

```

```

pwVcGroups          OBJECT IDENTIFIER ::= { pwVcConformance 1 }
pwVcCompliances     OBJECT IDENTIFIER ::= { pwVcConformance 2 }

```

```

pwModuleCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for agent that support PW
         operation."

```

```

MODULE -- this module
    MANDATORY-GROUPS { pwVcGroup,
                        pwVcPerformanceGroup }

    ::= { pwVcCompliances 1 }

-- Units of conformance.

pwVcGroup OBJECT-GROUP
    OBJECTS {
        pwVcIndexNext,
        pwVcType,
        pwVcOwner,
        pwVcPsnType,
        pwVcPriority,
        pwVcInboundMode,
        pwVcPeerAddrType,
        pwVcPeerIpv4Addr,
        pwVcPeerIpv6Addr,
        pwVcID,
        pwVcLocalGroupID,
        pwVcControlWord,
        pwVcLocalIfMtu,
        pwVcLocalIfString,
        pwVcRemoteGroupID,
        pwVcRemoteControlWord,
        pwVcRemoteIfMtu,
        pwVcRemoteIfString,
        pwVcOutboundVcLabel,

```

```

        pwVcInboundVcLabel,
        pwVcName,
        pwVcDescr,
        pwVcCreateTime,
        pwVcUpTime,
        pwVcAdminStatus,
        pwVcOperStatus,
        pwVcOutboundOperStatus,
        pwVcInboundOperStatus,
        pwVcRowStatus,
        pwVcStorageType
    }

```

```

STATUS    current
DESCRIPTION
    "Collection of objects needed for PW VC
    configuration."
 ::= { pwVcGroups 1 }

```

```

pwVcPerformanceGroup OBJECT-GROUP
OBJECTS {
    pwVcPerfTotalInPackets,
    pwVcPerfTotalInHCPackets,
    pwVcPerfTotalInBytes,
    pwVcPerfTotalInHCBytes,
    pwVcPerfTotalOutPackets,
    pwVcPerfTotalOutHCPackets,
    pwVcPerfTotalOutBytes,
    pwVcPerfTotalOutHCBytes,
    pwVcPerfCurrentInPackets,
    pwVcPerfCurrentInHCPackets,
    pwVcPerfCurrentInBytes,
    pwVcPerfCurrentInHCBytes,
    pwVcPerfCurrentOutPackets,
    pwVcPerfCurrentOutHCPackets,
    pwVcPerfCurrentOutBytes,
    pwVcPerfCurrentOutHCBytes,
    pwVcPerfIntervalValidData,
    pwVcPerfIntervalInPackets,
    pwVcPerfIntervalInHCPackets,
    pwVcPerfIntervalInBytes,
    pwVcPerfIntervalInHCBytes,
    pwVcPerfIntervalOutPackets,
    pwVcPerfIntervalOutHCPackets,
    pwVcPerfIntervalOutBytes,
    pwVcPerfIntervalOutHCBytes,
    pwVcPerfTotalErrorPackets
}

```

```

STATUS    current
DESCRIPTION

```

```
        "Collection of objects needed for PW VC  
        performance."  
 ::= { pwVcGroups 2 }  
  
-- END of PW-MIB  
  
END
```

Pseudo Wire (PW) Management
Information Base

November 2001

8 Security Considerations

There are a number of management objects defined in this MIB that have 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.

No managed objects in this MIB contain sensitive information.

SNMPv1 by itself is not a secure environment. 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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 2574](#) [[RFC2574](#)] and the View-based Access Control Model [RFC 2575](#) [[RFC2575](#)] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, 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.

Pseudo Wire (PW) Management
Information Base

November 2001

9 References

- [CEP] Malis, A., et al, "SONET/SDH Circuit Emulation over Packet (CEP)", Internet Draft <[draft-malis-pwe3-sonet-00](#)>, April 2001.
- [TRANS] Martini et al, "Transport of Layer 2 Frames Over MPLS" <[draft-martini-l2circuit-trans-mpls-07.txt](#)>, July 2000.
- [ENCAP] Martini et al, "Encapsulation Methods for Transport of Layer 2 Frames Over MPLS", <[draft-martini-l2circuit-encap-mpls-03.txt](#)>, July 2001.
- [EtherPW] So et al, "Ethernet Pseudo Wire Emulation Edge-to-Edge (PWE3)", <[draft-so-pwe3-ethernet-00.txt](#)>, October 2001.
- [FRMWK] Pate et al, "Framework for Pseudo Wire Emulation Edge-to-Edge (PWE3)", <[draft-pate-pwe3-framework-01.txt](#)>, July 2001.
- [PWREQ] Xiao et al, "Requirements for Pseudo Wire Emulation Edge-to-Edge (PWE3)", <[draft-ietf-pwe3-requirements-01.txt](#)>, July 2001.
- [CEPMIB] Danenberg et al, "SONET/SDH Circuit Emulation Service Over MPLS (CEM) Management Information Base Using SMIV2", <[draft-danenberg-pw-cem-mib-00](#)>, July 2001.
- [LSRMIB] Srinivasan, C., Viswanathan, A., and Nadeau, T. "MPLS Label Switch Router Management Information Base Using SMIV2", [draft-ietf-mpls-lsr-mib-07.txt](#), January 2001.

- [PWTC] Nadeau, T., et al, " Definitions for Textual Conventions and OBJECT-IDENTITIES for Pseudo-Wires Management", <[draft-Nadeau-pw-tc-mib-00.txt](#)>, July 2001.
- [PWMPISMIB] Zelig et al, "Pseudo Wire (PW) Over MPLS PSN Management Information Base",<[draft-zelig-pw-mpls-mib-00.txt](#)>, November 2001.
- [TEMIB] Srinivasan, C., Viswanathan, A., and Nadeau, T. "MPLS Traffic Engineering Management Information Base Using SMIV2", <[draft-ietf-mpls-te-mib-05.txt](#)>, November 2000.
- [LblStk] Rosen, E., Rekhter, Y., Tappan, D., Farinacci, D., Federokow, G., Li, T., and A. Conta, "MPLS Label Stack Encoding", Internet Draft <[draft-ietf-mpls-Label-encaps-07.txt](#)>, September 1999.

Zelig et al

Expires March 2002

[page 33]

Pseudo Wire (PW) Management
Information Base

November 2001

- [Assigned] Reynolds, J., and J. Postel, "Assigned Numbers", [RFC 1700](#), October 1994. See also: <http://www.isi.edu/in-notes/iana/assignments/smi-numbers>
- [IANAFamily] Internet Assigned Numbers Authority (IANA), ADDRESS FAMILY NUMBERS, (<http://www.isi.edu/in-notes/iana/assignments/address-family-numbers>), for MIB see: <ftp://ftp.isi.edu/mib/ianaaddressfamilynumbers.mib>
- [IFMIB] McCloghrie, K., and F. Kastenholz, "The Interfaces Group MIB using SMIV2", [RFC 2233](#), Nov. 1997
- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2571](#), April 1999.
- [RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-

based Internets", STD 16, [RFC 1155](#), May 1990.

- [RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, [RFC 1212](#), March 1991.
- [RFC1215] M. Rose, "A Convention for Defining Traps for use with the SNMP", [RFC 1215](#), March 1991.
- [RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, [RFC 2579](#), April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, [RFC 2580](#), April 1999.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, [RFC 1157](#), May 1990.
- [RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", [RFC 1901](#), January 1996.

- [RFC1906] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1906](#), January 1996.
- [RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", [RFC 2572](#), April 1999.
- [RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", [RFC 2574](#), April 1999.

- [RFC1905] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", [RFC 1905](#), January 1996.
- [RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", [RFC 2573](#), April 1999.
- [RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", [RFC 2575](#), April 1999.
- [RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", [RFC 2570](#), April 1999.

[@](#) Author's Addresses

David Zelig
Corrigent Systems
126, Yigal Alon st.
Tel Aviv, ISRAEL
Phone: +972-3-6945273
Email: davidz@corrigent.com

Thomas D. Nadeau
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Email: tnadeau@cisco.com

Dave Danenberg
Litchfield Communications, Inc.
76 Westbury Park Rd
Princeton Building East
Watertown, CT 06795
Email: dave_danenberg@litchfieldcomm.com

2730 Orchard Parkway
San Jose, CA 95134
Email: Andy.Malis@vivacenetworks.com

Sharon Mantin
Corrigent Systems
126, Yigal Alon st.
Tel Aviv, ISRAEL
Phone: +972-3-6948608
Email: sharonm@corrigent.com

1 Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.