

Network Working Group
Request for Comments: 1559
Obsoletes: 1289
Category: Standards Track

J. Saperia
Digital Equipment Corporation
December 1993

DECnet Phase IV MIB Extensions

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction	1
2. The Network Management Framework	2
2.1 Object Definitions	2
3. Selected Objects	3
4. Textual Conventions	4
5. Definitions	4
6. Changes from RFC 1289	67
7. Acknowledgements	68
8. References	68
9. Security Considerations	69
10. Author's Address	69

1. Introduction

This memo defines a set of DECnet Phase IV extensions that have been created for the Internet MIB. It reflects changes which are the result of operational experience based on RFC 1289.

When used in conjunction with the structure of management information (STD 16, RFC 1155), the management information base for network management of TCP/IP-based internets (STD 17, RFC 1213) and the Simple Network Management Protocol (STD 15, RFC 1157), it will be possible to provide integrated network management of combined TCP/IP and DECnet Phase IV based internets. This document was produced by the DECnet Phase IV MIB working group of the Internet Engineering Task Force (IETF).

With the adoption of The Simple Network Management Protocol (STD 15, RFC 1157), the management information base for network management of TCP/IP-based internets (STD 17, RFC 1213), and the structure of

management information (STD 16, RFC 1155), by the Internet, and a large number of vendor implementations of these standards in commercially available products, it became possible to provide a higher level of effective network management in TCP/IP-based internets than previously available. With the growth in the use of these standards, network managers desired to use this environment as a base for providing integrated network management of multi-protocol networks.

DECnet Phase IV is one widely used protocol which often coexists in IP-based internets. This memo provides the mechanisms by which IP-based management stations can effectively manage DECnet Phase IV based systems (especially router products) in an integrated fashion through the use of the standard Internet SMI, MIB and Simple Network Management Protocol.

DECnet Phase IV objects have been defined to be used in conjunction with the Internet MIB to allow access and control of these new objects by the Internet community. Additional support for other DECnet-based protocols such as RBMS (Remote Bridge Management Software) or other Digital Equipment Corporation specific hardware platforms is not included in this document.

2. The Network Management Framework

The Internet-standard Network Management Framework consists of three components. They are:

- o STD 16, RFC 1155 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management. STD 16, RFC 1212 defines a more concise description mechanism, which is wholly consistent with the SMI.
- o STD 17, RFC 1213 defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- o STD 15, RFC 1157 which defines the SNMP, the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1 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, each object type is named by an

OBJECT IDENTIFIER, an administratively assigned name. 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 refer to the object type.

3. Selected Objects

The objects included in this memo have been created from the DIGITAL Network Architecture Network Management Functional Specification Version 4.0.0, dated July 1983. An attempt has been made to provide a reasonable ordering of these variables into groups. These groups are:

- System Group
- Network Management Group
- Session Group
- End Group
- Routing Group
- Circuit Group
- DDCMP Group
- DDCMP Multipoint Control Group
- Ethernet Group
- Counters Group
- Adjacency Group
- Line Group
- Non Broadcast Line Group
- Area Group

An effort has also been made to preserve the original syntax of each object wherever possible, for example, a DECnet Phase IV object is Executor State. This was originally coded as a NICE (Network Information and Control Exchange) data type which is a coded single field object of 1 byte in length. When converted for inclusion into the Internet MIB using the Internet SMI, it became an enumerated integer.

All objects in this memo are described using the standard Internet SMI and BER of STD 16, RFC 1155. A complete description of an object will include the name, syntax and encoding. Just as with objects supported in the MIB (STD 17, RFC 1213), an object name is identified with an object identifier which has been administratively assigned. This identifies an Object Type. When an object type is combined with a specific instance, the particular object is uniquely identified. The use of Object Descriptors in this memo is consistent with that of STD 17, RFC 1213 - they are text strings meant to be read by humans. The descriptors have been taken from the original DIGITAL Network Architecture Network Management Functional Specification Version 4.0.0 Dated July 1983 which defined DECnet Phase IV objects. These

names were then massaged to put them in a form as consistent as possible with object type names listed in the standard Internet MIB. Object definition information is also taken directly from the Network Architecture Network Management Functional Specification cited above wherever possible. In this document, EXECUTOR is intended to reference only the DECnet software and is not intended to effect any other protocols which may be running on the system.

4. Textual Conventions

New datatypes have been introduced as a textual conventions in this DECnet Phase IV MIB document. The purpose of these additions is to facilitate understanding of new objects in this MIB. No changes to the SMI or the SNMP are necessary to support these conventions which are described in 5 (Definitions).

5. Definitions

```
DECNET-PHIV-MIB DEFINITIONS ::= BEGIN

IMPORTS
    Gauge
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212
    mib-2, DisplayString
        FROM RFC1213-MIB;

-- DECnet Phase-IV MIB

    phiv          OBJECT IDENTIFIER ::= { mib-2 18 }

-- textual conventions

PhivAddr ::= OCTET STRING (SIZE (2))
-- This data type is intended as a short word representation of
-- standard DECnet Phase IV addresses. DECnet addresses are
-- hierarchically structured numbers assigned to a particular
-- DECnet node. The address is structured so that the area
-- number is contained in the most significant 6 bits of the
-- first octet. The next 2 bits of the first octet contain
-- the first two bits of the host address. The remainder of
-- the host address is contained in the second octet.

PhivCounter ::= INTEGER
-- This data type has been created for DECnet counters. These
-- counters latch at their maximum specified value until either
-- the system is restarted, or they are reset to zero by the user
```

-- or management software.

InterfaceIndex ::= INTEGER

-- The range of ifIndex, i.e., (1..2147483647)

-- groups in the decnetiv mib

phivSystem	OBJECT IDENTIFIER ::= { phiv 1 }
phivManagement	OBJECT IDENTIFIER ::= { phiv 2 }
session	OBJECT IDENTIFIER ::= { phiv 3 }
end	OBJECT IDENTIFIER ::= { phiv 4 }
routing	OBJECT IDENTIFIER ::= { phiv 5 }
circuit	OBJECT IDENTIFIER ::= { phiv 6 }
ddcmp	OBJECT IDENTIFIER ::= { phiv 7 }
control	OBJECT IDENTIFIER ::= { phiv 8 }
ethernet	OBJECT IDENTIFIER ::= { phiv 9 }
counters	OBJECT IDENTIFIER ::= { phiv 10 }
adjacency	OBJECT IDENTIFIER ::= { phiv 11 }
line	OBJECT IDENTIFIER ::= { phiv 12 }
nonBroadcastLine	OBJECT IDENTIFIER ::= { phiv 14 }
area	OBJECT IDENTIFIER ::= { phiv 15 }

-- System Group

-- The implementation of the System Group is mandatory for

-- all systems.

phivSystemState OBJECT-TYPE

SYNTAX INTEGER {

on (1),

off (2),

shut (3),

restricted (4)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This represents the operational state of the executor node.

The possible states are:

ON Allows logical links.

OFF Allows no new links, terminates existing links, and stops routing traffic through.

SHUT Allows no new logical links, does not destroy existing logical links, and goes to the OFF state when all logical links are gone.

RESTRICTED Allows no new incoming logical links from other nodes.

NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155).

::= { phivSystem 1 }

phivExecIdent OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..32))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This is a text string that describes the executor node (for example, 'Research Lab'). The string is up to 32 characters of any type."

::= { phivSystem 2 }

-- Network Management Group

-- The implementation of the Network Management Group is
-- mandatory for all systems which contain a DECnet-style
-- management version.

phivMgmtMgmtVers OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This is the read-only Network Management Version, consisting of the version number, the Engineering Change Order (ECO) number, and the user ECO number (for example, 3.0.0). This parameter applies to the executor node only."

::= { phivManagement 1 }

-- Session Layer Group

-- The implementation of the Session Layer Group is optional.
-- A system can be said to implement this group if and only if
-- all objects in this group are implemented.

phivSessionSystemName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Name to be associated with the node identification. Only one name can be assigned to a node address or a circuit identification. No name should be used more than once in a DECnet network. Node-name is one to six upper case alphanumeric characters with at least one alpha character. A length of 0 indicates no name."

::= { session 1 }

phivSessionInTimer OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum duration between the time a connect is received for a process at the executor node and the time that process accepts or rejects it. If the connect is not accepted or rejected by the user within the number of seconds specified, Session Control rejects it for the user. A value of 0 indicates no timer is running."

::= { session 2 }

phivSessionOutTimer OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the duration between the time the executor requests a connect and the time that connect is acknowledged by the destination node. If the connect is not acknowledged within the number of seconds specified, Session Control returns an error. A value of 0 indicates no timer is running."

::= { session 3 }

-- End Communication Layer Group

-- The implementation of the End Communication Layer Group is optional.

-- A system can be said to implement this group if and only if

-- all objects in this group are implemented.

-- Remote State Table

phivEndRemoteTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivEndRemoteEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

```
    "Information about the state of sessions between the
    node under study and the nodes found in the table."
 ::= { end 1 }
```

phivEndRemoteEntry OBJECT-TYPE

SYNTAX PhivEndRemoteEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Information about a particular remote node as seen
from the end communication layer."

INDEX { phivEndRemoteHostNodeID }

::= { phivEndRemoteTable 1 }

PhivEndRemoteEntry ::=

SEQUENCE {

phivEndRemoteHostNodeID

PhivAddr,

phivEndRemoteState

INTEGER,

phivEndCircuitIndex

INTEGER,

phivEndActiveLinks

INTEGER,

phivEndDelay

INTEGER

}

phivEndRemoteHostNodeID OBJECT-TYPE

SYNTAX PhivAddr -- OCTET STRING (SIZE (2))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value is the address of the remote node to be
evaluated."

::= { phivEndRemoteEntry 1 }

phivEndRemoteState OBJECT-TYPE

SYNTAX INTEGER {

on (1),

off (2),

shut (3),

restricted (4)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This represents the operational state of the remote node

being evaluated.
The possible states are:

ON	Allows logical links.
OFF	Allows no new links, terminates existing links, and stops routing traffic through.
SHUT	Allows no new logical links, does not destroy existing logical links, and goes to the OFF state when all logical links are gone.
RESTRICTED	Allows no new incoming logical links from other nodes.

NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."

::= { phivEndRemoteEntry 2 }

phivEndCircuitIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A unique index value for each known circuit used to communicate with the remote node. This is the same value as phivCircuitIndex."

::= { phivEndRemoteEntry 3 }

phivEndActiveLinks OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This read-only parameter represents the number of active logical links from the executor to the destination node."

::= { phivEndRemoteEntry 4 }

phivEndDelay OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This read-only parameter is the average round trip delay in seconds to the destination node. This parameter is kept on a remote node basis."

::= { phivEndRemoteEntry 5 }

-- End System Counter Table

```
phivEndCountTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivEndCountEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about the counters associated with each end
        system that is known to the entity. These counters
        reflect totals from the perspective of the executor
        node."
    ::= { end 2 }
```

```
phivEndCountEntry OBJECT-TYPE
    SYNTAX PhivEndCountEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about a particular session between two end
        systems."
    INDEX { phivEndCountHostNodeID }
    ::= { phivEndCountTable 1 }
```

```
PhivEndCountEntry ::=
    SEQUENCE {
        phivEndCountHostNodeID
        PhivAddr,
        phivEndCountSecsLastZeroed
        PhivCounter,
        phivEndCountUsrBytesRec
        PhivCounter,
        phivEndCountUsrBytesSent
        PhivCounter,
        phivEndUCountUsrMessRec
        PhivCounter,
        phivEndCountUsrMessSent
        PhivCounter,
        phivEndCountTotalBytesRec
        PhivCounter,
        phivEndCountTotalBytesSent
        PhivCounter,
        phivEndCountTotalMessRec
        PhivCounter,
        phivEndCountTotalMessSent
        PhivCounter,
        phivEndCountConnectsRecd
        PhivCounter,
        phivEndCountConnectsSent
        PhivCounter,
```

```
        phivEndCountReponseTimeouts
            PhivCounter,
        phivEndCountRecdConnectResErrs
            PhivCounter
    }

phivEndCountHostNodeID OBJECT-TYPE
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value is the address of the remote node to be
        evaluated."
    ::= { phivEndCountEntry 1 }

phivEndCountSecsLastZeroed OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value is the number of seconds that have elapsed
        since the counters for the node in this table row were
        last set to zero. This counter is located in the
        network management layer, but is returned with the
        end system information which follows."
    ::= { phivEndCountEntry 2 }

phivEndCountUsrBytesRec OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of user bytes received from the target host."
    ::= { phivEndCountEntry 3 }

phivEndCountUsrBytesSent OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of user bytes sent to the target host."
    ::= { phivEndCountEntry 4 }

phivEndUCountUsrMessRec OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```

"Number of user messages received from the target host."
 ::= { phivEndCountEntry 5 }

phivEndCountUsrMessSent OBJECT-TYPE
 SYNTAX PhivCounter (0..2147483647)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Number of user messages sent to the target host."
 ::= { phivEndCountEntry 6 }

phivEndCountTotalBytesRec OBJECT-TYPE
 SYNTAX PhivCounter (0..2147483647)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Number of bytes received from the target host."
 ::= { phivEndCountEntry 7 }

phivEndCountTotalBytesSent OBJECT-TYPE
 SYNTAX PhivCounter (0..2147483647)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Number of bytes sent to the target host."
 ::= { phivEndCountEntry 8 }

phivEndCountTotalMessRec OBJECT-TYPE
 SYNTAX PhivCounter (0..2147483647)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Number of messages received from the target host."
 ::= { phivEndCountEntry 9 }

phivEndCountTotalMessSent OBJECT-TYPE
 SYNTAX PhivCounter (0..2147483647)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Number of messages sent to the target host."
 ::= { phivEndCountEntry 10 }

phivEndCountConnectsRecd OBJECT-TYPE
 SYNTAX PhivCounter (0..65535)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION

```
        "Number of connects received from the target host."
 ::= { phivEndCountEntry 11 }

phivEndCountConnectsSent OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of connects sent to the target host."
 ::= { phivEndCountEntry 12 }

phivEndCountReponseTimeouts OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of response timeouts."
 ::= { phivEndCountEntry 13 }

phivEndCountRecdConnectResErrs OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of received connect resource errors."
 ::= { phivEndCountEntry 14 }

-- additional End System objects

phivEndMaxLinks OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value represents the maximum active logical
         link count allowed for the executor."
 ::= { end 3 }

phivEndNSPVers OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This read-only parameter represents the version number
         of the node End Communication S/W. The format is
         version number, ECO, and user ECO, e.g., 4.1.0"
 ::= { end 4 }
```

phivEndRetransmitFactor OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum number of times the source End Communication at the executor node will restart the retransmission timer when it expires. If the number is exceeded, Session Control disconnects the logical link for the user."

::= { end 5 }

phivEndDelayFact OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This is the number by which to multiply one sixteenth of the estimated round trip delay to a node to set the retransmission timer to that node."

::= { end 6 }

phivEndDelayWeight OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This number represents the weight to apply to a current round trip delay estimate to a remote node when updating the estimated round trip delay to a node. On some systems the number must be 1 less than a power of 2 for computational efficiency."

::= { end 7 }

phivEndInactivityTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum duration of inactivity (no data in either direction) on a logical link before the node checks to see if the logical link still works. If no activity occurs within the minimum number of seconds, End Communication generates artificial traffic to test the link (End Communication specification)."

::= { end 8 }

```
phivEndCountZeroCount OBJECT-TYPE
    SYNTAX INTEGER {
        other (1),
        reset (2)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "When this value is set to 2, all of the counters in
        the End System Counter Table are set to zero."
    ::= { end 9 }

phivEndMaxLinksActive OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value represents the high water mark for the
        number of links that were active at any one time."
    ::= { end 10 }

-- Routing Layer Group

-- The implementation of the Routing Layer Group is mandatory for
-- all systems that implement level 1 routing layer
-- communications.

phivRouteBroadcastRouteTimer OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value determines the maximum time in seconds
        allowed between Routing updates on Ethernet
        circuits. When this timer expired before a routing
        update occurs, a routing update is forced. With a
        standard calculation, Routing also uses this timer
        to enforce a minimum delay between routing updates."
    ::= { routing 1 }

phivRouteBuffSize OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This parameter value determines the maximum size of
        a Routing message. It therefore determines the maximum
        size message that can be forwarded. This size includes
```

protocol overhead down to and including the End Communication layer, plus a constant value of 6. (This value of 6 is included to provide compatibility with the parameter definition in Phase III, which included the Routing overhead.) It does not include Routing or Data link overhead (except for the constant value of 6). There is one buffer size for all circuits.

NOTE: The BUFFER SIZE defines the maximum size messages that the Routing layer can forward. The SEGMENT BUFFER SIZE (defined below) defines the maximum size messages that the End Communication layer can transmit or receive. The SEGMENT BUFFER SIZE is always less than or equal to the BUFFER SIZE. Normally the two parameters will be equal. They may be different to allow the network manager to alter buffer sizes on all nodes without interruption of service. They both include an extra 6 bytes for compatibility with Phase III."

::= { routing 2 }

phivRouteRoutingVers OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This read-only parameter identifies the executor node's Routing version number. The format is version number, ECO, and user ECO, e.g., 4.1.0"

::= { routing 3 }

phivRouteMaxAddr OBJECT-TYPE

SYNTAX INTEGER (1..1023)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the largest node number and, therefore, number of nodes that can be known about by the executor node's home area."

::= { routing 4 }

phivRouteMaxBdcastNonRouters OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum total number of nonrouters the executor node can have on its Ethernet"


```
    circuits."  
 ::= { routing 5 }
```

phivRouteMaxBdcastRouters OBJECT-TYPE

```
SYNTAX INTEGER (0..65535)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "This value represents the maximum total number of  
    routers the executor node can have on its Ethernet  
    circuits."  
 ::= { routing 6 }
```

phivRouteMaxBufs OBJECT-TYPE

```
SYNTAX INTEGER (1..65535)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "This value represents the maximum number of transmit  
    buffers that Routing may use for all circuits."  
 ::= { routing 7 }
```

phivRouteMaxCircuits OBJECT-TYPE

```
SYNTAX INTEGER (1..65535)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "This value represents the maximum number of Routing  
    circuits that the executor node can know about."  
 ::= { routing 8 }
```

phivRouteMaxCost OBJECT-TYPE

```
SYNTAX INTEGER (1..1022)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "This value represents the maximum total path cost  
    allowed from the executor to any node within an area.  
    The path cost is the sum of the circuit costs along  
    a path between two nodes. This parameter defines the  
    point where the executor node's Routing routing  
    decision algorithm declares another node unreachable  
    because the cost of the least costly path to the  
    other node is excessive. For correct operation, this  
    parameter must not be less than the maximum path cost  
    of the network."  
 ::= { routing 9 }
```

phivRouteMaxHops OBJECT-TYPE

SYNTAX INTEGER (1..30)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum number of routing hops allowable from the executor to any other reachable node within an area. (A hop is the logical distance over a circuit between two adjacent nodes.) This parameter defines the point where the executor node's Routing routing decision algorithm declares another node unreachable because the length of the shortest path between the two nodes is too long. For correct operation, this parameter must not be less than the network diameter. (The network diameter is the reachability distance between the two nodes of the network having the greatest reachability distance, where reachability distance is the length the shortest path between a given pair of nodes.)"

::= { routing 10 }

phivRouteMaxVisits OBJECT-TYPE

SYNTAX INTEGER (1..63)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum number of nodes a message coming into the executor node can have visited. If the message is not for this node and the MAXIMUM VISITS number is exceeded, the message is discarded. The MAXIMUM VISITS parameter defines the point where the packet lifetime control algorithm discards a packet that has traversed too many nodes. For correct operation, this parameter must not be less than the maximum path length of the network. (The maximum path length is the routing distance between the two nodes of the network having the greatest routing distance, where routing distance is the length of the least costly path between a given pair of nodes.)"

::= { routing 11 }

phivRouteRoutingTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value determines the maximum time in seconds allowed between Routing updates on non-Ethernet

circuits. When this timer expires before a routing update occurs, a routing update is forced."
 ::= { routing 12 }

phivRouteSegBuffSize OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This parameter value determines the maximum size of an end-to-end segment. The size is a decimal integer in the range 1-65535. This size is in bytes. This size includes protocol overhead down to and including the End Communication layer, plus a constant value of 6. (This value of 6 is included to provide compatibility with the BUFFER SIZE parameter definition.) It does not include Routing or Data link overhead (except for the constant value of 6)."

::= { routing 13 }

phivRouteType OBJECT-TYPE

SYNTAX INTEGER {
 routing-III (1),
 nonrouting-III (2),
 area (3),
 routing-IV (4),
 nonrouting-IV (5)
}

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This parameter indicates the type of the executor node. The node-type is one of the following:

routing-III
nonrouting-III
routing-IV
nonrouting-IV
area

A routing node has full routing capability. A nonrouting node contains a subset of the Routing routing modules. The III and IV indicate the DNA phase of the node. Nonrouting nodes can deliver and receive packets to and from any node, but cannot route packets from other nodes through to other nodes. An area node routes between areas. Refer to the Routing specification for details.

For adjacent nodes, this is a read-only parameter that indicates the type of the reachable adjacent node.

NOTE: The ROUTING-III and NONROUTING-III values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155)"

::= { routing 14 }

phivRouteCountAgedPktLoss OBJECT-TYPE

SYNTAX PhivCounter (0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of aged packet losses."

::= { routing 15 }

phivRouteCountNodeUnrPktLoss OBJECT-TYPE

SYNTAX PhivCounter (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of node unreachable packet losses."

::= { routing 16 }

phivRouteCountOutRngePktLoss OBJECT-TYPE

SYNTAX PhivCounter (0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of node out-of-range packet losses."

::= { routing 17 }

phivRouteCountOverSzePktLoss OBJECT-TYPE

SYNTAX PhivCounter (0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of Oversized packet losses."

::= { routing 18 }

phivRouteCountPacketFmtErr OBJECT-TYPE

SYNTAX PhivCounter (0..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of packet format errors."

::= { routing 19 }

```
phivRouteCountPtlRteUpdtLoss OBJECT-TYPE
    SYNTAX PhivCounter (0..127)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of partial routing update losses."
    ::= { routing 20 }

phivRouteCountVerifReject OBJECT-TYPE
    SYNTAX PhivCounter (0..127)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of verification rejects."
    ::= { routing 21 }

-- Level 1 Routing Table

phivLevel1RouteTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivLevel1RouteEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about the currently known DECnet Phase
         IV Routes."
    ::= { routing 22 }

phivLevel1RouteEntry OBJECT-TYPE
    SYNTAX PhivLevel1RouteEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about the currently known DECnet Phase
         IV Routes."
    INDEX { phivLevel1RouteNodeAddr }
    ::= { phivLevel1RouteTable 1 }

PhivLevel1RouteEntry ::=
    SEQUENCE {
        phivLevel1RouteNodeAddr
            PhivAddr,
        phivLevel1RouteCircuitIndex
            INTEGER,
        phivLevel1RouteCost
            INTEGER,
        phivLevel1RouteHops
            INTEGER,
        phivLevel1RouteNextNode
```

```
        PhivAddr
    }

phivLevellRouteNodeAddr OBJECT-TYPE
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value is the address of the node about which
        routing information is contained in this level 1
        routing table."
    ::= { phivLevellRouteEntry 1 }

phivLevellRouteCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique index value for each known circuit. This is
        the index to the circuit state table and is the same
        value as phivCircuitIndex."
    ::= { phivLevellRouteEntry 2 }

phivLevellRouteCost OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This read-only parameter represents the total cost
        over the current path to the destination node. Cost is
        a positive integer value associated with using a
        circuit. Routing routes messages (data) along the path
        between two nodes with the smallest cost. COST is kept
        on a remote node basis."
    ::= { phivLevellRouteEntry 3 }

phivLevellRouteHops OBJECT-TYPE
    SYNTAX INTEGER (0..127)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This read-only parameter represents the number of hops
        over to a destination node. A hop is Routing value
        representing the logical distance between two nodes in
        a network. HOPS is kept on a remote node basis."
    ::= { phivLevellRouteEntry 4 }

phivLevellRouteNextNode OBJECT-TYPE
```

```
SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This read-only value indicates the next node on the
    circuit used to get to the node under scrutiny
    (next hop)."
```

::= { phivLevellRouteEntry 5 }

-- Additional routing parameters

phivRouteCountZeroCount OBJECT-TYPE

```
    SYNTAX INTEGER {
        other (1),
        reset (2)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "When this value is set to 2, the following objects are
        set to Zero: phivRouteCountAgedPktLoss,
        phivRouteCountNodeUnrPktLoss,
        phivRouteCountOutRngePktLoss,
        phivRouteCountOverSzePktLoss,
        phivRouteCountPacketFmtErr,
        phivRouteCountPtlRteUpdtLoss, and
        phivRouteCountVerifReject."
```

::= { routing 23 }

phivRouteSystemAddr OBJECT-TYPE

```
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS obsolete
    DESCRIPTION
        "DECnet Phase IV node address."
```

::= { routing 24 }

phivRouteRoutingType OBJECT-TYPE

```
    SYNTAX INTEGER {
        routing-III (1),
        nonrouting-III (2),
        area (3),
        routing-IV (4),
        nonrouting-IV (5)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
```

"This read-write parameter indicates the type of the executor node. The node-type is one of the following:

```
routing-III
nonrouting-III
routing-IV
nonrouting-IV
area
```

A routing node has full routing capability. A nonrouting node contains a subset of the Routing routing modules. The III and IV indicate the DNA phase of the node. Nonrouting nodes can deliver and receive packets to and from any node, but cannot route packets from other nodes through to other nodes. An area node routes between areas. Refer to the Routing specification for details.

For adjacent nodes, this is a read-only parameter that indicates the type of the reachable adjacent node.

NOTE: The ROUTING-III and NONROUTING-III values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155)"

```
::= { routing 25 }
```

phivRouteSystemAddress OBJECT-TYPE

SYNTAX PhivAddr -- OCTET STRING (SIZE (2))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"DECnet Phase IV node address."

```
::= { routing 26 }
```

-- Circuit Group

-- The implementation of the Circuit Group is mandatory for
-- all systems.

-- Circuit Parameters Table

phivCircuitParametersTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivCircuitParametersEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Information about the parameters associated with all
circuits currently known."

```
::= {circuit 1 }
```



```
phivCircuitParametersEntry OBJECT-TYPE
    SYNTAX PhivCircuitParametersEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Parameters information about all circuits currently
        known."
    INDEX { phivCircuitIndex }
    ::= { phivCircuitParametersTable 1 }
```

```
PhivCircuitParametersEntry ::=
    SEQUENCE {
        phivCircuitIndex
            INTEGER,
        phivCircuitLineIndex
            INTEGER,
        phivCircuitCommonState
            INTEGER,
        phivCircuitCommonSubState
            INTEGER,
        phivCircuitCommonName
            DisplayString,
        phivCircuitExecRecallTimer
            INTEGER,
        phivCircuitCommonType
            INTEGER,
        phivCircuitService
            INTEGER,
        phivCircuitExecCost
            INTEGER,
        phivCircuitExecHelloTimer
            INTEGER
    }
```

```
phivCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique index value for each known circuit."
    ::= { phivCircuitParametersEntry 1 }
```

```
phivCircuitLineIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The line on which this circuit is active.  This is
```

the same as the ifIndex."
 ::= { phivCircuitParametersEntry 2 }

phivCircuitCommonState OBJECT-TYPE

SYNTAX INTEGER {
 on (1),
 off (2),
 service (3),
 cleared (4)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the circuit's Network Management operational state. NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."

::= { phivCircuitParametersEntry 3 }

phivCircuitCommonSubState OBJECT-TYPE

SYNTAX INTEGER {
 starting (1),
 reflecting (2),
 looping (3),
 loading (4),
 dumping (5),
 triggering (6),
 autoservice (7),
 autoloading (8),
 autodumping (9),
 autotriggering (10),
 synchronizing (11),
 failed (12),
 running (13)
}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the circuit's Network Management operational and service substate. NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."

::= { phivCircuitParametersEntry 4 }

phivCircuitCommonName OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The name of the circuit entry in the table, for example, SVA-0 or in a level 2 router ASYNC-8 or ETHER-1)."
 ::= { phivCircuitParametersEntry 5 }

phivCircuitExecRecallTimer OBJECT-TYPE

SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION
"This parameter represents the minimum number of seconds to wait before restarting the circuit. A value of 0 indicates not timer is running."
 ::= { phivCircuitParametersEntry 6 }

phivCircuitCommonType OBJECT-TYPE

SYNTAX INTEGER {
ddcmp-point (1),
ddcmp-control (2),
ddcmp-tributary (3),
x25 (4),
ddcmp-dmc (5),
ethernet (6),
ci (7),
qp2-dte20 (8),
bisync (9),
other (14),
fddi (15)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Represents the type of the circuit. For X.25 circuits, the value must be set to X25. For DDCMP and Ethernet circuits it is read only and is the same value as the protocol of the associated line.
NOTE: Values 1 - 5 are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."
 ::= { phivCircuitParametersEntry 7 }

phivCircuitService OBJECT-TYPE

SYNTAX INTEGER {
enabled (1),
disabled (2)
}
ACCESS read-write
STATUS mandatory

DESCRIPTION

"This value indicates whether or not Network Management allows service operations on a circuit. The values for service-control are as follows:

ENABLED SERVICE state and/or service functions are allowed.

DISABLED SERVICE state and/or service functions are not allowed.

NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."

::= { phivCircuitParametersEntry 8 }

phivCircuitExecCost OBJECT-TYPE

SYNTAX INTEGER (1..25)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the routing cost of the circuit. Routing sends messages along the path between two nodes having the smallest cost."

::= { phivCircuitParametersEntry 9 }

phivCircuitExecHelloTimer OBJECT-TYPE

SYNTAX INTEGER (1..8191)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value determines the frequency of Routing Hello messages sent to the adjacent node on the circuit."

::= { phivCircuitParametersEntry 10 }

-- Circuit Counters Table

phivCircuitCountTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivCircuitCountEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Information about the counters associated with all circuits currently known."

::= { circuit 2 }

phivCircuitCountEntry OBJECT-TYPE

SYNTAX PhivCircuitCountEntry

```

ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Counter information about all circuits currently known"
INDEX      { phivCircuitIndex }
 ::= { phivCircuitCountTable 1 }

```

```

PhivCircuitCountEntry ::=
SEQUENCE {
    phivCircuitCountSecLastZeroed
        PhivCounter,
    phivCircuitCountTermPacketsRecd
        PhivCounter,
    phivCircuitCountOriginPackSent
        PhivCounter,
    phivCircuitCountTermCongLoss
        PhivCounter,
    phivCircuitCountCorruptLoss
        PhivCounter,
    phivCircuitCountTransitPksRecd
        PhivCounter,
    phivCircuitCountTransitPkSent
        PhivCounter,
    phivCircuitCountTransitCongestLoss
        PhivCounter,
    phivCircuitCountCircuitDown
        PhivCounter,
    phivCircuitCountInitFailure
        PhivCounter,
    phivCircuitCountAdjDown
        PhivCounter,
    phivCircuitCountPeakAdj
        PhivCounter,
    phivCircuitCountBytesRecd
        PhivCounter,
    phivCircuitCountBytesSent
        PhivCounter,
    phivCircuitCountDataBlocksRecd
        PhivCounter,
    phivCircuitCountDataBlocksSent
        PhivCounter,
    phivCircuitCountUsrBuffUnav
        PhivCounter
}

```

```

phivCircuitCountSecLastZeroed OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only

```

STATUS mandatory

DESCRIPTION

"Number of seconds since the circuit counters for this circuit were last zeroed."

::= { phivCircuitCountEntry 1 }

phivCircuitCountTermPacketsRecd OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of terminating packets received on this circuit."

::= { phivCircuitCountEntry 2 }

phivCircuitCountOriginPackSent OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of originating packets sent on this circuit."

::= { phivCircuitCountEntry 3 }

phivCircuitCountTermCongLoss OBJECT-TYPE

SYNTAX PhivCounter (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of terminating congestion losses on this circuit."

::= { phivCircuitCountEntry 4 }

phivCircuitCountCorruptLoss OBJECT-TYPE

SYNTAX PhivCounter (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of corruption losses on this circuit."

::= { phivCircuitCountEntry 5 }

phivCircuitCountTransitPksRecd OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of Transit packets received on this circuit."

::= { phivCircuitCountEntry 6 }

phivCircuitCountTransitPkSent OBJECT-TYPE

```
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Number of transit packets sent on this circuit."
 ::= { phivCircuitCountEntry 7 }
```

```
phivCircuitCountTransitCongestLoss OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of transit congestion losses on this circuit."
    ::= { phivCircuitCountEntry 8 }
```

```
phivCircuitCountCircuitDown OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of circuit downs on this circuit."
    ::= { phivCircuitCountEntry 9 }
```

```
phivCircuitCountInitFailure OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of Initialization failures on this circuit."
    ::= { phivCircuitCountEntry 10 }
```

```
phivCircuitCountAdjDown OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This counter indicates the number of adjacency losses
        that result from any of the following:
        Node listener timeout
        Invalid data received at node listener
        Unexpected control (initialization or verification)
        message received
        Routing message received with a checksum error
        Node identification from a routing message or a
        Hello message that is not the one expected Hello
        message received indicating that connectivity
        became one-way
        Adjacency idled."
```

```
 ::= { phivCircuitCountEntry 11 }

phivCircuitCountPeakAdj OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This counter indicates the maximum number of nodes
         that are up on the circuit."
    ::= { phivCircuitCountEntry 12 }

phivCircuitCountBytesRecd OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of bytes received on this circuit."
    ::= { phivCircuitCountEntry 13 }

phivCircuitCountBytesSent OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of bytes sent on this circuit."
    ::= { phivCircuitCountEntry 14 }

phivCircuitCountDataBlocksRecd OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks received on this circuit."
    ::= { phivCircuitCountEntry 15 }

phivCircuitCountDataBlocksSent OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of data blocks sent on this circuit."
    ::= { phivCircuitCountEntry 16 }

phivCircuitCountUsrBuffUnav OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```



```
        "Number of user buffer unavailable errors."
 ::= { phivCircuitCountEntry 17 }

-- Additional Circuit Parameters

phivCircuitOrigQueueLimit OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This parameter indicates the maximum number of
        originating packets that may be outstanding on this
        circuit. This does not include route-thru traffic."
 ::= { circuit 3 }

phivCircuitCountZeroCount OBJECT-TYPE
    SYNTAX INTEGER {
        other (1),
        reset (2)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "When this value is set to 2, all of the counters in the
        Circuit Counter Table are set to zero."
 ::= { circuit 4 }

-- DDCMP Circuit Group

-- The implementation of the DDCMP Circuit Group is optional.
-- A system can be said to implement this group if and only if
-- all objects in this group are implemented.

-- DDCMP Parameters Table

phivDDCMPCircuitParametersTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivDDCMPCircuitParametersEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Information about DDCMP circuit parameters."
 ::= { ddcmp 1 }

phivDDCMPCircuitParametersEntry OBJECT-TYPE
    SYNTAX PhivDDCMPCircuitParametersEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
```

```

        "Parameters information about DDCMP circuits currently
        known."
INDEX      { phivDDCMPCircuitIndex }
 ::= { phivDDCMPCircuitParametersTable 1 }

PhivDDCMPCircuitParametersEntry ::=
SEQUENCE {
    phivDDCMPCircuitIndex
        INTEGER,
    phivDDCMPCircuitAdjNodeAddr
        INTEGER,
    phivDDCMPCircuitTributary
        INTEGER
}

phivDDCMPCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique index value for each known DDCMP circuit.
        This is the same value as phivCircuitIndex."
    ::= { phivDDCMPCircuitParametersEntry 1 }

phivDDCMPCircuitAdjNodeAddr OBJECT-TYPE
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The address of the adjacent node."
    ::= { phivDDCMPCircuitParametersEntry 2 }

phivDDCMPCircuitTributary OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value represents the Data Link physical tributary
        address of the circuit."
    ::= { phivDDCMPCircuitParametersEntry 3 }

-- DDCMP Circuit Counter Table

phivDDCMPCircuitCountTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivDDCMPCircuitCountEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION

```

```

    "Information about the DDCMP counters associated with all
    circuits currently known."
 ::= { ddcmp 2 }

```

```

phivDDCMPCircuitCountEntry OBJECT-TYPE
    SYNTAX PhivDDCMPCircuitCountEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Counter information about DDCMP circuits now known"
    INDEX { phivCircuitIndex }
    ::= { phivDDCMPCircuitCountTable 1 }

```

```

PhivDDCMPCircuitCountEntry ::=
    SEQUENCE {
        phivDDCMPCircuitErrorsInbd
            PhivCounter,
        phivDDCMPCircuitErrorsOutbd
            PhivCounter,
        phivDDCMPCircuitRmteReplyTimeouts
            PhivCounter,
        phivDDCMPCircuitLocalReplyTimeouts
            PhivCounter,
        phivDDCMPCircuitRmteBuffErrors
            PhivCounter,
        phivDDCMPCircuitLocalBuffErrors
            PhivCounter,
        phivDDCMPCircuitSelectIntervalsElap
            PhivCounter,
        phivDDCMPCircuitSelectTimeouts
            PhivCounter
    }

```

```

phivDDCMPCircuitErrorsInbd OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of Data errors inbound."
    ::= { phivDDCMPCircuitCountEntry 1 }

```

```

phivDDCMPCircuitErrorsOutbd OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of outbound data errors."
    ::= { phivDDCMPCircuitCountEntry 2 }

```

```
phivDDCMPCircuitRmteReplyTimeouts OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of remote reply timeouts."
    ::= { phivDDCMPCircuitCountEntry 3 }

phivDDCMPCircuitLocalReplyTimeouts OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of local Reply timeouts."
    ::= { phivDDCMPCircuitCountEntry 4 }

phivDDCMPCircuitRmteBuffErrors OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of remote reply time out errors."
    ::= { phivDDCMPCircuitCountEntry 5 }

phivDDCMPCircuitLocalBuffErrors OBJECT-TYPE
    SYNTAX PhivCounter (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of local buffer errors."
    ::= { phivDDCMPCircuitCountEntry 6 }

phivDDCMPCircuitSelectIntervalsElap OBJECT-TYPE
    SYNTAX PhivCounter (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Selection intervals that have elapsed."
    ::= { phivDDCMPCircuitCountEntry 7 }

phivDDCMPCircuitSelectTimeouts OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of selection timeouts."
    ::= { phivDDCMPCircuitCountEntry 8 }
```

-- DDCMP Line Count Table

phivDDCMPLineCountTable OBJECT-TYPE
SYNTAX SEQUENCE OF PhivDDCMPLineCountEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "The DDCMP Line Count Table."
 ::= { ddcmp 3 }

phivDDCMPLineCountEntry OBJECT-TYPE
SYNTAX PhivDDCMPLineCountEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "There is one entry in the table for each line."
INDEX { phivDDCMPLineCountIndex }
 ::= { phivDDCMPLineCountTable 1 }

PhivDDCMPLineCountEntry ::=
SEQUENCE {
 phivDDCMPLineCountIndex
 InterfaceIndex,
 phivDDCMPLineCountDataErrsIn
 PhivCounter,
 phivDDCMPLineCountRmteStationErrs
 PhivCounter,
 phivDDCMPLineCountLocalStationErrs
 PhivCounter
}

phivDDCMPLineCountIndex OBJECT-TYPE
SYNTAX InterfaceIndex
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The line on which this entry's equivalence is
 effective. The interface identified by a particular
 value of this index is the same interface as
 identified by the same value of phivLineIndex.
 This value is the ifIndex."
 ::= { phivDDCMPLineCountEntry 1 }

phivDDCMPLineCountDataErrsIn OBJECT-TYPE
SYNTAX PhivCounter (0..255)
ACCESS read-only
STATUS mandatory

DESCRIPTION

"Number of data errors inbound."
 ::= { phivDDCMPLineCountEntry 2 }

phivDDCMPLineCountRmteStationErrs OBJECT-TYPE

SYNTAX PhivCounter (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Number of remote station errors."
 ::= { phivDDCMPLineCountEntry 3 }

phivDDCMPLineCountLocalStationErrs OBJECT-TYPE

SYNTAX PhivCounter (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Number of local station errors."
 ::= { phivDDCMPLineCountEntry 4 }

-- DDCMP Multipoint Circuit Control Group

-- The implementation of the DDCMP Multipoint Circuit Control
-- Group is optional. A system can be said to implement this group
-- if and only if all objects in this group are implemented.

phivControlSchedTimer OBJECT-TYPE

SYNTAX INTEGER (50..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This value represents the number of milliseconds
between recalculation of tributary polling priorities."
DEFVAL { 200 }
 ::= { control 1 }

phivControlDeadTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This value represents the number of milliseconds
between polls of one of the set of dead
tributaries."
DEFVAL { 10000 }
 ::= { control 2 }

phivControlDelayTimer OBJECT-TYPE

```

SYNTAX INTEGER (1..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This value represents the minimum number of
    milliseconds to delay between polls. The delay timer
    limits the effect of a very fast control station on
    slow tributaries."
 ::= { control 3 }

```

phivControlStreamTimer OBJECT-TYPE

```

SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This value represents the number of milliseconds a
    tributary or a half duplex remote station is
    allowed to hold the line.

    NOTE: This parameter can also be applied to
    half-duplex lines of type DDCMP POINT."
DEFVAL { 6000 }
 ::= { control 4 }

```

-- DDCMP Multipoint Circuit Control Parameters Table

phivControlParametersTable OBJECT-TYPE

```

SYNTAX SEQUENCE OF PhivControlParametersEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Information about control circuit parameters."
 ::= { control 5 }

```

phivControlParametersEntry OBJECT-TYPE

```

SYNTAX PhivControlParametersEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Parameters information about control circuits
    currently known."
INDEX { phivControlCircuitIndex }
 ::= { phivControlParametersTable 1 }

```

```

PhivControlParametersEntry ::=
    SEQUENCE {
        phivControlCircuitIndex
            INTEGER,

```

```
    phivControlBabbleTimer
        INTEGER,
    phivControlMaxBufs
        INTEGER,
    phivControlMaxTransmits
        INTEGER,
    phivControlDyingBase
        INTEGER,
    phivControlDyingIncrement
        INTEGER,
    phivControlDeadThreshold
        INTEGER,
    phivControlDyingThreshold
        INTEGER,
    phivControlInactTreshold
        INTEGER,
    phivControlPollingState
        INTEGER,
    phivControlPollingSubState
        INTEGER,
    phivControlTransTimer
        INTEGER
}
```

```
phivControlCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique index value for each known multipoint
        control circuit.
        This is the same value as phivCircuitIndex."
    ::= { phivControlParametersEntry 1 }
```

```
phivControlBabbleTimer OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value represents the number of milliseconds that a
        selected tributary or remote half-duplex station is
        allowed to transmit."
    DEFVAL { 6000 }
    ::= { phivControlParametersEntry 2 }
```

```
phivControlMaxBufs OBJECT-TYPE
    SYNTAX INTEGER (1..254)
    ACCESS read-write
```


STATUS mandatory

DESCRIPTION

"This value represents the maximum number of buffers the tributary can use from a common buffer pool. If not set, there is no common buffer pool and buffers are explicitly supplied by the higher level. Count is a decimal integer in the range 1-254."

::= { phivControlParametersEntry 3 }

phivControlMaxTransmits OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the maximum number of data messages that can be transmitted at one time. Count is a decimal integer in the range 1-255."

DEFVAL { 4 }

::= { phivControlParametersEntry 4 }

phivControlDyingBase OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the base priority to which a tributary is reset each time it has been polled. A separate base can be set for each of the indicated polling states. Base is a decimal integer in the range 0-255. If not set, the defaults are: active, 255; inactive, 0; and dying, 0."

::= { phivControlParametersEntry 5 }

phivControlDyingIncrement OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the increment added to the tributary priority each time the scheduling timer expires. If not set, the defaults are: active, 0; inactive, 64; and dying, 16."

::= { phivControlParametersEntry 6 }

phivControlDeadThreshold OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the number of times to poll the active, inactive, or dying tributary before changing its polling state to dead because of receive timeouts. Count is a decimal integer in the range 0-255."

DEFVAL { 8 }

::= { phivControlParametersEntry 7 }

phivControlDyingThreshold OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the number of times to poll the active or inactive tributary before changing its polling state to dying because of receive timeouts. Count is a decimal integer in the range 0-255."

DEFVAL { 2 }

::= { phivControlParametersEntry 8 }

phivControlInactTreshold OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the number of times to poll the active tributary before changing its polling state to inactive because of no data response. Count is a decimal integer in the range 0-255."

DEFVAL { 8 }

::= { phivControlParametersEntry 9 }

phivControlPollingState OBJECT-TYPE

SYNTAX INTEGER {
 automatic (1),
 active (2),
 inactive (3),
 dying (4),
 dead (5)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the state of the tributary relative to the multipoint polling algorithm. If not set the default is AUTOMATIC. The possible states are:

AUTOMATIC

The tributary's state is allowed to vary according to the operation of the polling algorithm.

ACTIVE/INACTIVE/DYING/DEAD

The tributary is locked in the specified state.

NOTE: These values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155."

```
::= { phivControlParametersEntry 10 }
```

phivControlPollingSubState OBJECT-TYPE

```
SYNTAX INTEGER {
    active (1),
    inactive (2),
    dying (3),
    dead (4)
}
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"This value represents the tributary's state as determined by the polling algorithm. This applies only when the polling state is AUTOMATIC and is read-only to Network Management. Polling-substate is one of ACTIVE, INACTIVE, DYING, or DEAD. It is displayed as a tag on the polling state, for example: AUTOMATIC-INACTIVE."

```
::= { phivControlParametersEntry 11 }
```

phivControlTransTimer OBJECT-TYPE

```
SYNTAX INTEGER (0..65535)
```

```
ACCESS read-write
```

```
STATUS mandatory
```

```
DESCRIPTION
```

"This value represents the number of milliseconds to delay between data message transmits. Milliseconds is a decimal integer in the range 0-65535."

```
DEFVAL { 0 }
```

```
::= { phivControlParametersEntry 12 }
```

```
-- Ethernet Group
```

```
-- The implementation of the Ethernet Group is mandatory
```

```
-- for all systems which support ethernet links.
```

-- Ethernet Parameters Table

phivEthLinkParametersTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivEthLinkParametersEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Information about ethernet link parameters."

::= { ethernet 1 }

phivEthLinkParametersEntry OBJECT-TYPE

SYNTAX PhivEthLinkParametersEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Parameter information about ethernet links currently known."

INDEX { phivEthLinkIndex }

::= { phivEthLinkParametersTable 1 }

PhivEthLinkParametersEntry ::=

SEQUENCE {

phivEthLinkIndex

INTEGER,

phivEthDesigRouterNodeAddr

PhivAddr,

phivEthMaxRouters

INTEGER,

phivEthRouterPri

INTEGER,

phivEthHardwareAddr

OCTET STRING

}

phivEthLinkIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The circuit over which this links information is collected. This is the same as phivCircuitIndex."

::= { phivEthLinkParametersEntry 1 }

phivEthDesigRouterNodeAddr OBJECT-TYPE

SYNTAX PhivAddr -- OCTET STRING (SIZE (2))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value is the address of the designated router."
 ::= { phivEthLinkParametersEntry 2 }

phivEthMaxRouters OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This parameter is the maximum number of routers (other than the executor itself) allowed on the circuit by Routing for circuits that are owned by the executor node."

::= { phivEthLinkParametersEntry 3 }

phivEthRouterPri OBJECT-TYPE

SYNTAX INTEGER (0..127)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This parameter is the priority that this router is to have in the selection of designated router for the circuit on circuits that are owned by the executor node."

DEFVAL { 64 }

::= { phivEthLinkParametersEntry 4 }

phivEthHardwareAddr OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This read-only parameter is the address that is associated with the line device hardware as seen by the DECnet Software. This value is not the same as ifPhysAddress."

::= { phivEthLinkParametersEntry 5 }

-- Counters Group

-- The implementation of the Counters Group is optional.

-- A system can be said to implement this group if and only if
-- all objects in this group are implemented.

-- Counters Table

phivCountersCountTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivCountersCountEntry

ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "Information about ethernet link counters."
 ::= { counters 1 }

phivCountersCountEntry OBJECT-TYPE
SYNTAX PhivCountersCountEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "Counter information about ethernet links currently
 known."
INDEX { phivCountersIndex }
 ::= { phivCountersCountTable 1 }

PhivCountersCountEntry ::=
SEQUENCE {
 phivCountersIndex
 InterfaceIndex,
 phivCountersCountBytesRecd
 PhivCounter,
 phivCountersCountBytesSent
 PhivCounter,
 phivCountersCountDataBlocksRecd
 PhivCounter,
 phivCountersCountDataBlocksSent
 PhivCounter,
 phivCountersCountEthUsrBuffUnav
 PhivCounter,
 phivCountersCountMcastBytesRecd
 PhivCounter,
 phivCountersCountDataBlksRecd
 PhivCounter,
 phivCountersCountDataBlksSent
 PhivCounter,
 phivCountersCountMcastBlksRecd
 PhivCounter,
 phivCountersCountBlksSentDef
 PhivCounter,
 phivCountersCountBlksSentSingleCol
 PhivCounter,
 phivCountersCountBlksSentMultCol
 PhivCounter,
 phivCountersCountSendFailure
 PhivCounter,
 phivCountersCountCollDetectFailure
 PhivCounter,

```
    phivCountersCountReceiveFailure
        PhivCounter,
    phivCountersCountUnrecFrameDest
        PhivCounter,
    phivCountersCountDataOver
        PhivCounter,
    phivCountersCountSysBuffUnav
        PhivCounter,
    phivCountersCountUsrBuffUnav
        PhivCounter
}
```

phivCountersIndex OBJECT-TYPE

SYNTAX InterfaceIndex

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The interface to which these counters apply. This is the same interface as identified by the same value of phivLineIndex. This value is the ifIndex."

::= { phivCountersCountEntry 1 }

phivCountersCountBytesRecd OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of bytes received over this link."

::= { phivCountersCountEntry 2 }

phivCountersCountBytesSent OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of bytes sent over this link."

::= { phivCountersCountEntry 3 }

phivCountersCountDataBlocksRecd OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only

STATUS obsolete

DESCRIPTION

"Number of data blocks received over this link."

::= { phivCountersCountEntry 4 }

phivCountersCountDataBlocksSent OBJECT-TYPE

SYNTAX PhivCounter (0..2147483647)

ACCESS read-only
STATUS obsolete
DESCRIPTION
 "Number of data blocks sent over this link."
 ::= { phivCountersCountEntry 5 }

phivCountersCountEthUsrBuffUnav OBJECT-TYPE
SYNTAX PhivCounter (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of user buffer unavailable errors over this
 link."
 ::= { phivCountersCountEntry 6 }

phivCountersCountMcastBytesRecd OBJECT-TYPE
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of multicast bytes received over this link."
 ::= { phivCountersCountEntry 7 }

phivCountersCountDataBlksRecd OBJECT-TYPE
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of data blocks received over this link."
 ::= { phivCountersCountEntry 8 }

phivCountersCountDataBlksSent OBJECT-TYPE
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of data blocks sent over this link."
 ::= { phivCountersCountEntry 9 }

phivCountersCountMcastBlksRecd OBJECT-TYPE
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "Number of multicast blocks received over this link."
 ::= { phivCountersCountEntry 10 }

phivCountersCountBlksSentDef OBJECT-TYPE


```
SYNTAX PhivCounter (0..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Number of blocks sent, initially deferred over this
    link."
 ::= { phivCountersCountEntry 11 }
```

```
phivCountersCountBlksSentSingleCol OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of blocks sent, single collision over this link."
    ::= { phivCountersCountEntry 12 }
```

```
phivCountersCountBlksSentMultCol OBJECT-TYPE
    SYNTAX PhivCounter (0..2147483647)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of blocks sent, multiple collisions over this
        link."
    ::= { phivCountersCountEntry 13 }
```

```
phivCountersCountSendFailure OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of send failures over this link."
    ::= { phivCountersCountEntry 14 }
```

```
phivCountersCountCollDetectFailure OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of collision detect check failures over this
        link."
    ::= { phivCountersCountEntry 15 }
```

```
phivCountersCountReceiveFailure OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of receive failures over this link."
```

```
 ::= { phivCountersCountEntry 16 }

phivCountersCountUnrecFrameDest OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of unrecognized frame destinations over this
        link."
    ::= { phivCountersCountEntry 17 }

phivCountersCountDataOver OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of data overruns over this link."
    ::= { phivCountersCountEntry 18 }

phivCountersCountSysBuffUnav OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of system buffer unavailables over this link."
    ::= { phivCountersCountEntry 19 }

phivCountersCountUsrBuffUnav OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of user buffer unavailables."
    ::= { phivCountersCountEntry 20 }

-- Adjacency Group

-- The implementation of the Adjacency Group is mandatory for all
-- conformant implementations of this memo.

-- The phivAdjTable has been made obsolete it has been replaced with
-- the phivAdjNodeTable.

phivAdjTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivAdjEntry
    ACCESS not-accessible
    STATUS obsolete
    DESCRIPTION
```

```
    "The Adjacency Table."
 ::= { adjacency 1 }

phivAdjEntry OBJECT-TYPE
    SYNTAX PhivAdjEntry
    ACCESS not-accessible
    STATUS obsolete
    DESCRIPTION
        "There is one entry in the table for each adjacency."
    INDEX { phivAdjCircuitIndex }
    ::= { phivAdjTable 1 }

PhivAdjEntry ::=
    SEQUENCE {
        phivAdjCircuitIndex
            INTEGER,
        phivAdjNodeAddr
            PhivAddr,
        phivAdjBlockSize
            INTEGER,
        phivAdjListenTimer
            INTEGER (1..65535),
        phivAdjCircuitEtherServPhysAddr
            OCTET STRING,
        phivAdjType
            INTEGER,
        phivAdjState
            INTEGER,
        phivAdjPriority
            INTEGER,
        phivAdjExecListenTimer
            INTEGER (1..65535)
    }
phivAdjCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS obsolete
    DESCRIPTION
        "A unique index value for each known circuit."
    ::= { phivAdjEntry 1 }

phivAdjNodeAddr OBJECT-TYPE
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS obsolete
    DESCRIPTION
        "The address of the adjacent node."
    ::= { phivAdjEntry 2 }
```

phivAdjBlockSize OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This read-only parameter is the block size that was negotiated with the adjacent Routing layer during Routing initialization over a particular circuit. It includes the routing header, but excludes the data link header. This parameter is qualified by ADJACENT NODE."

::= { phivAdjEntry 3 }

phivAdjListenTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This value determines the maximum number of seconds allowed to elapse before Routing receives some message (either a Hello message or a user message) from the adjacent node on the circuit. It was agreed during Routing initialization with the adjacent Routing layer. This parameter is qualified by ADJACENT NODE."

::= { phivAdjEntry 4 }

phivAdjCircuitEtherServPhysAddr OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (6))

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This parameter indicates the Ethernet physical address of an adjacent node that is being serviced on this circuit. This parameter is a qualifier for SERVICE SUBSTATE."

::= { phivAdjEntry 5 }

phivAdjType OBJECT-TYPE

SYNTAX INTEGER {

routing-III (1),

nonrouting-III (2),

area (3),

routing-IV (4),

nonrouting-IV (5)

}

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This parameter indicates the type of adjacency.

For adjacent nodes, this is a read-only parameter that indicates the type of the reachable adjacent node.

NOTE: The routing-III and nonrouting-III values are incremented by one compared to the standard DECnet values in order to maintain compliance with RFC 1155)"

::= { phivAdjEntry 6 }

phivAdjState OBJECT-TYPE

```
SYNTAX INTEGER {
    initializing (1),          -- Ethernet one-way
    up (2),                  -- Ethernet two-way
    run (3),                 -- The eight DDCMP/X.25 states
    circuit-rejected (4),
    data-link-start (5),
    routing-layer-initialize (6),
    routing-layer-verify (7),
    routing-layer-complete (8),
    off (9),
    halt (10)
}
```

ACCESS read-only

STATUS obsolete

DESCRIPTION

"This value indicates the state of a router adjacency. On adjacencies over a circuit of type (phivCircuitCommonType) Ethernet, CI, or FDDI, with an adjacent node of type (phivAdjType) ROUTING IV or AREA, this variable is the state of the Ethernet Initialization Layer for this adjacency, and can have values INITIALIZING or UP. (See Section 9.1.1 of DECnet Phase IV Routing Layer Functional Specification.)

On adjacencies over a circuit of type (phivCircuitCommonType) Ethernet, CI, or FDDI, with an adjacent node of type (phivAdjType) NONROUTING IV, this variable will always take on the value UP.

On adjacencies over a circuit of type (phivCircuitCommonType) DDCMP POINT, DDCMP CONTROL, DDCMP TRIBUTARY, DDCMP DMC, or X.25, this variable is the state of the Routing Layer Initialization Circuit State. (See section 7.3, *ibid.*) It can have values between RUN and HALT.

On adjacencies over a circuit of type (phivCircuitCommonType) OTHER, this variable may be

used in a manner consistent with the Initialization Layer used on that circuit."
 ::= { phivAdjEntry 7 }

phivAdjPriority OBJECT-TYPE
 SYNTAX INTEGER (0..255)
 ACCESS read-only
 STATUS obsolete
 DESCRIPTION
 "Priority assigned by the adjacent node for this circuit."
 ::= { phivAdjEntry 8 }

phivAdjExecListenTimer OBJECT-TYPE
 SYNTAX INTEGER (1..65535)
 ACCESS read-only
 STATUS obsolete
 DESCRIPTION
 "This read-only value determines the maximum number of seconds allowed to elapse before Routing receives some message (either a Hello message or a user message) from the adjacent node on the circuit. It was agreed during Routing initialization with the adjacent Routing layer."
 ::= { phivAdjEntry 9 }

-- New Adjacency Table this replaces the phivAdjTable.

phivAdjNodeTable OBJECT-TYPE
 SYNTAX SEQUENCE OF PhivAdjNodeEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "The Adjacent Node Table."
 ::= { adjacency 2 }

phivAdjNodeEntry OBJECT-TYPE
 SYNTAX PhivAdjNodeEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "There is one entry in the table for each adjacency."
 INDEX { phivAdjNodeCircuitIndex, phivAdjAddr }
 ::= { phivAdjNodeTable 1 }

PhivAdjNodeEntry ::=
 SEQUENCE {
 phivAdjNodeCircuitIndex
 INTEGER,

```

    phivAdjAddr
        PhivAddr,
    phivAdjNodeBlockSize
        INTEGER,
    phivAdjNodeListenTimer
        INTEGER,
    phivAdjNodeCircuitEtherServPhysAddr
        OCTET STRING,
    phivAdjNodeType
        INTEGER,
    phivAdjNodeState
        INTEGER,
    phivAdjNodePriority
        INTEGER
}

```

phivAdjNodeCircuitIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A unique index value for each known circuit. This value is the same as phivCircuitIndex and identifies the circuit over which the adjacency is realized."

::= { phivAdjNodeEntry 1 }

phivAdjAddr OBJECT-TYPE

SYNTAX PhivAddr -- OCTET STRING (SIZE (2))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The address of the adjacent node."

::= { phivAdjNodeEntry 2 }

phivAdjNodeBlockSize OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This read-only parameter is the block size that was negotiated with the adjacent Routing layer during Routing initialization over a particular circuit. It includes the routing header, but excludes the data link header. This parameter is qualified by ADJACENT NODE."

::= { phivAdjNodeEntry 3 }

phivAdjNodeListenTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "This value determines the maximum number of seconds
 allowed to elapse before Routing receives some message
 (either a Hello message or a user message) from the
 adjacent node on the circuit. It was agreed during
 Routing initialization with the adjacent Routing layer.
 This parameter is qualified by ADJACENT NODE."
 ::= { phivAdjNodeEntry 4 }

phivAdjNodeCircuitEtherServPhysAddr OBJECT-TYPE
 SYNTAX OCTET STRING (SIZE (6))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "This parameter indicates the Ethernet physical address
 of an adjacent node that is being serviced on this
 circuit. This parameter is a qualifier for SERVICE
 SUBSTATE."
 ::= { phivAdjNodeEntry 5 }

phivAdjNodeType OBJECT-TYPE
 SYNTAX INTEGER {
 routing-III (1),
 nonrouting-III (2),
 area (3),
 routing-IV (4),
 nonrouting-IV (5)
 }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "This parameter indicates the type of adjacency.

 For adjacent nodes, this is a read-only parameter that
 indicates the type of the reachable adjacent node.
 NOTE: The routing-III and nonrouting-III values are
 incremented by one compared to the standard DECnet
 values in order to maintain compliance with RFC 1155)"
 ::= { phivAdjNodeEntry 6 }

phivAdjNodeState OBJECT-TYPE
 SYNTAX INTEGER {
 initializing (1), -- Ethernet one-way
 up (2), -- Ethernet two-way
 run (3), -- The eight DDCMP/X.25 states
 circuit-rejected (4),


```

    data-link-start (5),
    routing-layer-initialize (6),
    routing-layer-verify (7),
    routing-layer-complete (8),
    off (9),
    halt (10)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This value indicates the state of a router adjacency.
    On adjacencies over a circuit of type
    (phivCircuitCommonType) Ethernet, CI, or FDDI, with an
    adjacent node of type (phivAdjNodeType) ROUTING IV or AREA,
    this variable is the state of the Ethernet
    Initialization Layer for this adjacency, and can have
    values INITIALIZING or UP. (See Section 9.1.1 of
    DECnet Phase IV Routing Layer Functional Specification.)

    On adjacencies over a circuit of type
    (phivCircuitCommonType) Ethernet, CI, or FDDI, with an
    adjacent node of type (phivAdjNodeType) NONROUTING IV,
    this variable will always take on the value UP.

    On adjacencies over a circuit of type
    (phivCircuitCommonType) DDCMP POINT, DDCMP CONTROL,
    DDCMP TRIBUTARY, DDCMP DMC, or X.25, this variable is
    the state of the Routing Layer Initialization Circuit
    State. (See section 7.3, ibid.) It can have values
    between RUN and HALT.

    On adjacencies over a circuit of type
    (phivCircuitCommonType) OTHER, this variable may be
    used in a manner consistent with the Initialization
    Layer used on that circuit."
 ::= { phivAdjNodeEntry 7 }

phivAdjNodePriority OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Priority assigned by the adjacent node for this
        circuit."
    ::= { phivAdjNodeEntry 8 }

```

```
-- Line Group

-- The implementation of the Line Group is mandatory for all
-- conformant implementations of this memo.
```

```
phivLineTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivLineEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The Line Table."
    ::= { line 1 }
```

```
phivLineEntry OBJECT-TYPE
    SYNTAX PhivLineEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "There is one entry in the table for each line."
    INDEX { phivLineIndex }
    ::= { phivLineTable 1 }
```

```
PhivLineEntry ::=
    SEQUENCE {
        phivLineIndex
            InterfaceIndex,
        phivLineName
            DisplayString,
        phivLineState
            INTEGER,
        phivLineSubstate
            INTEGER,
        phivLineService
            INTEGER,
        phivLineDevice
            DisplayString,
        phivLineReceiveBufs
            INTEGER,
        phivLineProtocol
            INTEGER,
        phivLineServiceTimer
            INTEGER,
        phivLineMaxBlock
            INTEGER
    }
```

```
phivLineIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
```

ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The line on which this entry's equivalence is effective.
 This is the same as the ifIndex."
::= { phivLineEntry 1 }

phivLineName OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The name of the line on this row of the table."
::= { phivLineEntry 2 }

phivLineState OBJECT-TYPE
SYNTAX INTEGER {
 on (1),
 off (2),
 service (3),
 cleared (4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "This value represents Network Management operational
 state.
 NOTE that these values are incremented by one compared to
 the standard DECnet values."
::= { phivLineEntry 3 }

phivLineSubstate OBJECT-TYPE
SYNTAX INTEGER {
 starting (1),
 reflecting (2),
 looping (3),
 loading (4),
 dumping (5),
 triggering (6),
 auto-service (7),
 auto-loading (8),
 auto-dumping (9),
 auto-triggering (10),
 synchronizing (11),
 failed (12),
 running (13)
}
ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the line's read-only Network Management substate.

NOTE that these values are incremented by one compared to the standard DECnet values."

::= { phivLineEntry 4 }

phivLineService OBJECT-TYPE

SYNTAX INTEGER {

starting (1),

reflecting (2),

looping (3),

other (4)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the line's read-only Network Management service.

NOTE that these values are incremented by one compared to the standard DECnet values and OTHER is a new addition."

::= { phivLineEntry 5 }

phivLineDevice OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..16))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the Physical Link device to be used on the line."

::= { phivLineEntry 6 }

phivLineReceiveBufs OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the number of receive buffers reserved for the line. It is a decimal number in the range 0-65535. 0 is supported for those vendors that do not reserve buffers on a per line basis and use a pool of buffers that can be used by any line."

::= { phivLineEntry 7 }

phivLineProtocol OBJECT-TYPE

SYNTAX INTEGER {

ddcmp-point (1),

```

        ddcmp-control (2),
        ddcmp-tributary (3),
        reserved (4),
        ddcmp-dmc (5),
        olapb (6),
        ethernet (7),
        ci (8),
        qp2 (9),
        other (14),
        fddi (15)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value represents the protocol used on the line
        device.  Note that these values are incremented by
        one compared to the standard DECnet values."
    ::= { phivLineEntry 8 }

```

```

phivLineServiceTimer OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value represents the amount of time in
        milliseconds allowed to elapse before a Data Link
        receive request completes while doing service
        operations."
    ::= { phivLineEntry 9 }

```

```

phivLineMaxBlock OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This value represents the Data Link maximum block
        size on the line."
    ::= { phivLineEntry 10 }

```

-- Non Broadcast Line Group

-- The implementation of the Non Broadcast Line Group is optional.
 -- A system can be said to implement this group if and only if
 -- all objects in this group are implemented.

```

phivNonBroadcastTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PhivNonBroadcastEntry
    ACCESS not-accessible

```

STATUS mandatory
DESCRIPTION
 "The Non Broadcast Table."
 ::= { nonBroadcastLine 1 }

phivNonBroadcastEntry OBJECT-TYPE
SYNTAX PhivNonBroadcastEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "There is one entry in the table for each
 Non Broadcast line."
INDEX { phivNonBroadcastIndex }
 ::= { phivNonBroadcastTable 1 }

PhivNonBroadcastEntry ::=
SEQUENCE {
 phivNonBroadcastIndex
 InterfaceIndex,
 phivNonBroadcastController
 INTEGER,
 phivNonBroadcastDuplex
 INTEGER,
 phivNonBroadcastClock
 INTEGER,
 phivNonBroadcastRetransmitTimer
 INTEGER
}

phivNonBroadcastIndex OBJECT-TYPE
SYNTAX InterfaceIndex
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "The Non Broadcast line on which this entry's
 equivalence is effective. This is the same value
 as the ifIndex."
 ::= { phivNonBroadcastEntry 1 }

phivNonBroadcastController OBJECT-TYPE
SYNTAX INTEGER {
 normal (1),
 loopback (2),
 other (3)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This value represents the Physical Link hardware controller mode for the line device. The values for controller-mode are:

NORMAL For normal controller operating mode.

LOOPBACK For software controllable loopback of the controller. On those devices that can support this mode, it causes all transmitted messages to be looped back from within the controller itself. This is accomplished without any manual intervention other than the setting of this parameter value.

OTHER indicates function is not supported
Note that these values are incremented by one compared to the standard DECnet values."

::= { phivNonBroadcastEntry 2 }

phivNonBroadcastDuplex OBJECT-TYPE

SYNTAX INTEGER {
 full (1),
 half (2)
}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the Physical Link hardware duplex mode of the line device. The possible modes are:

FULL Full-duplex

HALF Half-duplex

Note that these values are incremented by one compared to the standard DECnet values."

::= { phivNonBroadcastEntry 3 }

phivNonBroadcastClock OBJECT-TYPE

SYNTAX INTEGER {
 external (1),
 internal (2),
 other (3)
}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents the Physical Link hardware clock mode for the line device. The values for clock-mode are:

INTERNAL For software controllable loopback use of the clock. On those devices that can support this mode, it causes the device to supply a clock signal such that a transmitted messages can be looped back from outside the device. This may require manual intervention other than the setting of this parameter value. For example, the operator may have to connect a loopback plug in place of the normal line.

EXTERNAL For normal clock operating mode, where the clock signal is supplied externally to the controller. Note that these values are incremented by one compared to the standard DECnet values."

::= { phivNonBroadcastEntry 4 }

phivNonBroadcastRetransmitTimer OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This value represents number of milliseconds before the Data Link retransmits a block on the line. On half-duplex lines, this parameter is the select timer."

DEFVAL { 3000 }

::= { phivNonBroadcastEntry 5 }

-- Area Parameters Group

-- The implementation of the Area Parameters Group is mandatory
-- for all systems which implement level 2 routing.

phivAreaTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhivAreaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Table of information kept on all areas known to this unit."

::= { area 1 }

phivAreaEntry OBJECT-TYPE

SYNTAX PhivAreaEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"The area routing information."

INDEX { phivAreaNum }


```
::= { phivAreaTable 1 }
```

```
PhivAreaEntry ::=
  SEQUENCE {
    phivAreaNum
      INTEGER,
    phivAreaState
      INTEGER,
    phivAreaCost
      Gauge,
    phivAreaHops
      INTEGER,
    phivAreaNextNode
      PhivAddr,
    phivAreaCircuitIndex
      INTEGER
  }
```

```
phivAreaNum OBJECT-TYPE
  SYNTAX INTEGER (0..64)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This value indicates the area number of this entry."
  ::= { phivAreaEntry 1 }
```

```
phivAreaState OBJECT-TYPE
  SYNTAX INTEGER {
    reachable (4),
    unreachable (5)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This value indicates the state of the area"
  ::= { phivAreaEntry 2 }
```

```
phivAreaCost OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total cost over the current path to the
    destination area. Cost is a value associated with
    using a circuit. Routing routes messages (data)
    along the path between 2 areas with the smallest
    cost."
  ::= { phivAreaEntry 3 }
```

```
phivAreaHops OBJECT-TYPE
    SYNTAX INTEGER (0..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of hops to a destination area. A hop is
        the routing value representing the logical distance
        between two areas in network."
    ::= { phivAreaEntry 4 }

phivAreaNextNode OBJECT-TYPE
    SYNTAX PhivAddr -- OCTET STRING (SIZE (2))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The next node on the circuit used to get to the
        area under scrutiny."
    ::= { phivAreaEntry 5 }

phivAreaCircuitIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique index value for each known circuit."
    ::= { phivAreaEntry 6 }

-- Additional Area Parameters

phivAreaMaxCost OBJECT-TYPE
    SYNTAX INTEGER (1..1022)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value represents the maximum total path cost
        allowed from the executor to any other level 2 routing
        node. The AREA MAXIMUM COST number is decimal in the
        range 1-1022. This parameter is only applicable if
        the executor node is of type AREA."
    ::= { area 2 }

phivAreaMaxHops OBJECT-TYPE
    SYNTAX INTEGER (1..30)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This value represents the maximum number of routing hops
```

allowable from the executor to any other level 2 routing node. This parameter is only applicable if the executor node is of type AREA."
 ::= { area 3 }

phivRouteMaxArea OBJECT-TYPE

SYNTAX INTEGER (1..63)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This value represents the largest area number and, therefore, number of areas that can be known about by the executor node's Routing. This parameter is only applicable if the executor node is of type AREA."

::= { area 4 }

END

6. Changes from RFC 1289

Several changes have been made to this document. These changes include:

- (1) Ranges have been added on all PhivCounter types to remove ambiguity which might otherwise have occurred.
- (2) Made clear that all indexes start with 1 and count up.
- (3) Spelling and typographic changes.
- (4) Changes to improve consistency with other documents including the removal of subranging within definitions of sequences defining table entries.
- (5) Updated compliance text to conform to current practice.
- (6) Fixed discrepancy between description and range clause for phivControlMaxBufs.
- (7) Added a space that was missing between SYNTAX and INTEGER in the phivRouteType object.
- (8) Both phivRouteType and phivRouteSystemAddr have been made obsolete. They have been replaced with phivRouteRoutingType and phivRouteSystemAddress which are both read-write objects.

- (9) A new Adjacency table has been added as adjacency 2. This table is identical to the original except that `phivAdjExecListenTimer` was not carried into the new version. The existing Adjacency table and all objects in it have been made obsolete. The index to the new table is `phivAdjNodeCircuitIndex` and `phivAdjAddr`.
- (10) Objects `phivCountersCountDataBlocksRecd` and `phivCountersCountDataBlocksSent` have both been made obsolete since the DESCRIPTION information overlapped with the `phivCountersCountDataBlksRecd` and `phivCountersCountDataBlksSent` objects which have been retained.
- (11) The following groups have been moved from mandatory to optional status: Session, End, DDCMP, DDCMP Multipoint Circuit Control, Counters, and Non Broadcast Line.

7. Acknowledgements

This document is the result of work undertaken the by DECnet Phase IV MIB working group. In addition, the special contributions and comments of the following members are also acknowledged:

Chris Chiotasso, Sparticus

Steven Hunter, National Energy Research Supercomputer Center,
Lawrence Livermore National Laboratory

8. References

- [1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, NRI, April 1988.
- [2] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [3] McCloghrie K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.
- [4] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

- [5] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.
- [6] Cook, J., Editor, "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 1284, Chipcom Corporation, December 1991.
- [7] Digital Equipment Corporation, "DECnet-ULTRIX NCP Command Reference", Digital Equipment Corporation, Maynard, Massachusetts.
- [8] Digital Equipment Corporation, "DECnet-ULTRIX USE Guide", Digital Equipment Corporation, Maynard, Massachusetts.
- [9] Digital Equipment Corporation, "DECnet DIGITAL Network Architecture, Network Management Functional Specification", Version 4.0.0. Digital Equipment Corporation, Maynard, Massachusetts, July 1983.
- [10] Digital Equipment Corporation, "DECnet DIGITAL Network Architecture, Routing Layer Functional Specification", Version 2.0.0. Digital Equipment Corporation, Maynard, Massachusetts, May 1983.

9. Security Considerations

Security issues are not discussed in this memo.

10. Author's Address

Jon Saperia
Digital Equipment Corporation
153 Taylor Street
M/S TAY2-2/B5
Littleton, MA 01460

Phone: +1 508-952-3171
EMail: saperia@tay.dec.com

