

# scspAtmarpHFSMTable

```
ScspAtmarpHFSMEntry ::=
  SEQUENCE {
    scspHFSMHFSMState           ScspHFSMStateType,
    scspHFSMHelloIn            Counter32,
    scspHFSMHelloOut           Counter32,
    scspHFSMHelloInvalidIn     Counter32,
    scspHFSMHelloInterval      Integer32,
    scspHFSMDeadFactor          Integer32,
    scspHFSMFamilyID           Integer32,
    scspAtmarpHFSMRowStatus    RowStatus
  }
```

# scspAtmarpPeerTable

ScspAtmarpPeerEntry ::=

SEQUENCE {

scspAtmarpPeerIndex

Integer32,

scspAtmarpPeerIPAddr

IpAddress,

scspAtmarpPeerAtmAddr

AtmAddr,

scspAtmarpPeerVCType

AtmConnKind,

scspAtmarpPeerVPI

SCSPVPIInteger,

scspAtmarpPeerVCI

SCSPVCIInteger,

scspAtmarpPeerDCSID

OCTET STRING(SIZE(0..63)),

scspAtmarpPeerRowStatus

RowStatus

}

# scspAtmarpLSTable

ScspAtmarpLSEntry ::=

SEQUENCE {

scspAtmarpLSLSIPAddr

IpAddress,

scspAtmarpLSLSAtmAddr

AtmAddr,

scspAtmarpLSRowStatus

RowStatus

}

# scspAtmarpServerGroupTable

ScspAtmarpServerGroupEntry ::=

SEQUENCE {

scspAtmarpServerGroupNetMask

IpAddress,

scspAtmarpServerGroupSubnetAddr

IpAddress,

scspAtmarpServerGroupRowStatus

RowStatus

}

# **SCSP ATMARP dependent MIB**

## **Four MIB tables:**

server group table, local server table, peer server table, and HFSM table.

## **Three MIB traps:**

scspHFSMDown, scspHFSMWaiting, and scspHFSMBiConn to notify the status of HFSM.

```
scspDCSCSURequestOut      Counter32,  
scspDCSCSUReplyOut       Counter32,  
scspDCSCSUReplyIn        Counter32,  
scspDCSCSUInvalidRequestIn Counter32,  
scspDCSCSUInvalidReplyIn Counter32,  
scspDCSCSAIn             Counter32,  
scspDCSCSAOut            Counter32,  
scspDCSCSAREXmtd         Counter32,  
scspDCSCSAREXmtQDepth    Integer32,  
scspDCSRowStatus         RowStatus
```

```
}
```

# scspDCSTable

```
ScspDCSEntry ::= SEQUENCE {  
    scspDCSID                OCTET STRING(SIZE(0..63)),  
    scspDCSCAFSMState       ScspCAFSMStateType,  
    scspDCSCASequence       Integer32,  
    scspDCSCAIn             Counter32,  
    scspDCSCAOut            Counter32,  
    scspDCSCAInvalidIn     Counter32,  
    scspDCSCADuplicateIn   Counter32,  
    scspDCSMSState         INTEGER,  
    scspDCSCSUSIn          Counter32,  
    scspDCSCSUSOut         Counter32,  
    scspDCSCSUSInvalidIn   Counter32,  
    scspDCSCSUSRequestIn   Counter32,  
}
```

# scspLSTable

```
ScspLSEntry ::= SEQUENCE {  
    scspLSID                OCTET STRING(SIZE(0..63)),  
    scspLSHelloInterval    Integer32,  
    scspLSDeadFactor       Integer32,  
    scspLSCAReXmInterval   Integer32,  
    scspLSCSUSReXmtInterval Integer32,  
    scspLSCSUSReXmtInterval Integer32,  
    scspLSCSAMaxReXmt      Integer32,  
    scspLSRowStatus        RowStatus  
}
```



# scspServerGroupTable

```
ScspServerGroupEntry ::= SEQUENCE {  
    scspServerGroupID          Integer32,  
    scspServerGroupPID        ScspPIDType,  
    scspServerGroupRowStatus   RowStatus  
}
```

# **SCSP Protocol Independent MIB**

**Three MIB Tables:**

**scspServerGroupTable, scspLSTable, and scspDC-  
STable.**

**One SNMP trap scspCSAReXmExceed trap event  
used to notify management that CSA exchange  
between LS and DCS has timed out.**