Schneider Private MIB

A

About this Chapter

This chapter provides the detailed tree structure of the Schneider private MIB and a description of its services.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
The Schneider Private MIB	394
Schneider Private MIB Tree Structure	396
MIB Subtree Description	404
Switch Subtree Description	405
Port 502 Messaging Subtree Description	406
I/O Scanning Subtree Description	407
Global Data Subtree Description	408
Web Subtree Description	409
Address Server Subtree Description	410
Equipment Profile Subtree Description	411
Time Management Subtree Description	413
Email Subtree Description	414
Transparent Factory MIB Version	415
Private Traps and MIB Files	416

The Schneider Private MIB

Introduction

A MIB (Management Information Base) is an element used in network management. Network management services are based on the need to monitor and manage:

- performance
- fault occurrences
- security

NOTE: The Transparent Factory private MIB does not define specific management applications and policies.

Each MIB contains a finite number of objects. Use the SNMP manager's GET and SET to retrieve system information and to set system environment variables.

Schneider Private MIB

The Transparent Factory SNMP-embedded component controls the Schneider private MIB function. This private MIB, and its associated services, manages all system components. The private MIB provides the data to manage the main Transparent Factory communication services for all the communication components of the Transparent Factory architecture, including:

- Ethernet communication modules (NOE, ETY, M1E, etc.)
- CPUs with Ethernet communication ports

Elsewhere in this guide is the detailed tree structure of the transparentFactory-Ethernet MIB *(see page 396)*.

Private MIB Identifier

Schneider Electric obtained a Private Enterprise Number (PEN) from the Internet Assigned Numbers Authority (IANA). That number represents a subtree in the SNMP MIB, a number that is a unique identifier used for Groupe Schneider.

The object identifier for the root of the Groupe Schneider subtree is **1.3.6.1.4.1.3833** and represents a path to the subtree as follows:

```
iso(1)
        org(3)
                dod(6)
                        internet(1)
                                private(4)
                                        enterprises(1)
                                                groupeschneider(3833)
                                                        transparentFactoryEthernet(1)
```

Under the Groupe Schneider private MIB is a TFE private MIB, transparentFactoryEthernet(1).

Schneider Private MIB Tree Structure

Introduction

This topic outlines the tree structure for the private Schneider MIB (Schneider TFE-V01-04.mib) for all Transparent Ready products.

The groupeschneider (3833) subtree is the root of Groupe Schneider's private MIB in the Structure of Management Information (SMI) used by SNMP and defined in RFC-1155, a specification that defines the structure and identification of management information for TCP/IP-based networks.

Tree Structure

```
groupeschneider (3833)
(1) transparentFactoryEthernet
|---(1) switch
|------(14) saConfiguration
|----------(1) saChassis
|----------(2) saAgent
|----------(3) saUserGroup
|----------(5) saRingRedundancy
|----------(7) saLLDP
|------(15) saPlatform4
|----------(1) saPlatform4BasicL2
|---(2) Port502Messaging
|------(1) port502Status
|------(2) port502SupportedProtocol
|------(3) port502IpSecurity
|------(4) port502MaxConn
|------(5) port502LocalConn
|------(6) port502RemConn
|------(7) port502IpSecurityTable
|----------(1) port502IpSecurityEntry
|--------------(1) attemptFails
|--------------(2) ipSourceAddress
|------(8) port502ConnTable
|----------(1) port502ConnEntry
```

```
|----------(1) port502ConnLocalPort
|----------(2) port502ConnRemAddress
|----------(3) port502ConnRemPort
|----------(4) port502ConnType
|----------(5) port502ConnMsgIn
|----------(6) port502ConnMsgOut
|----------(7) port502ConnMsgErr
|----------(8) port502XwayNet
|----------(9) port502XwayStation
|------(9) port502MsgIn
|------(10) port502MsgOut
|------(11) port502MsgOutErr
|------(12) port502AddStackStat
|------(13) port502AddStackStatTable
|------------(1) port502AddStackStatEntry
|----------------(1) port502AddStackStatIndex
|----------------(2) port502PeaKTcpRetransSegs
|---(3) ioScanning
|------(1) ioScanStatus
|------(2) ioScanMaxDevice
|------(3) ioScanPolledDevice
|------(4) ioScanTransSend
|------(5) ioScanGlbHealth
|------(6) ioScanningDeviceTable
|----------(1) ioScanDeviceEntry
|--------------(1) IoScanDeviceRemAddress
|--------------(2) IoScanDeviceHealth
|--------------(3) IoScanDeviceRate
|--------------(4) ioScanInputLocalAddress
|--------------(5) ioScanOutputLocalAddress
|---(4) globalData
|------(1) glbDataStatus
|------(2) glbDataMaxPub
```

```
|------(3) glbDataMaxSub
|------(4) glbDataPub
|------(5) glbDataSub
|------(6) glbDataPubErr
|------(7) glbDataSubErr
|------(8) glbDataGlbSubHealth
|------(9) glbDataPubTable
|----------(1) glbDataPubEntry
|--------------(1) glbDataPubSourceAddress
|--------------(2) glbDataPubHostId
|--------------(3) glbDataPubNetId
|--------------(4) glbDataPubGroupId
|--------------(5) glbDataPubCnt
|--------------(6) glbDataPubErrCnt
|--------------(7) glbDataPubDistribRate
|--------------(8) glbDataPubDuplicateErr
|------(10) glbDataSubTable
|--------------(1) glbDataSubEntry
|------------------(1) glbDataSubSourceAddress
|------------------(2) glbDataSubHostId
|------------------(3) glbDataSubNetId
|------------------(4) glbDataSubGroupId
|------------------(5) glbDataSubCnt
|------------------(6) glbDataSubErrCnt
|------------------(7) glbDataMinimumSeparation
|------------------(8) glbDataHealth
|------------------(9) glbDataHealthTimeOut
|------------------(10) glbDataLastRecErr
|---(5) Web
|------(1) webStatus
|------(2) webPassword
|------(3) webSuccessfullAccess
|------(4) webFailedAttempts
```

```
|---(6) addressServer
|------(1) addressServerStatus
|---(7) equipmentProfile
|------(1) profileProductName
|------(2) profileVersion
|------(3) profileCommunicationServices
|------(4) profileGlobalStatus
|------(5) profileConfigMode
|------(6) profileRoleName
|------(7) profileBandwidthMgt
|------(8) profileBandwidthDistTable
|---------(1) profileBandwidthDistEntry
|------------(1) bandwidthDistributionIndex
|------------(2) port502Bandwidth
|------------(3) ioScanningBandwidth
|------------(4) globalDataBandwidth
|------------(5) otherBandwidth
|------(9) profileLedDisplayTable
|---------(1) profileLedDisplayEntry
|------------(1) ledIndex
|------------(2) ledName
|------------(3) ledDescr
|------------(4) ledState
|------(10) profileSlot
|------(11) profileCPUType
|------(12) profileTrapTableEntriesMax
|------(13) profileTrapTable
|---------(1) profileTrapEntry
|------------(1) trapCommunityName
|------------(2) remoteIpAddress
|------------(3) authenticationTrap
|------------(4) port502Trap
|------------(5) ioScanningTrap
```

```
|------------(6) globalDataTrap
|------------(7) webTrap
|------------(8) addressServerTrap
|------------(9) profileTrap
|------------(10) timeManagementTrap
|------------(11) emailTrap
|------(14) profileSpecificId
|------(15) profileIpAddress
|------(16) profileIpNetMask
|------(17) profileIpGateway
|------(18) profileMacAddress
|------(19) profileImplementationClass
|------(100) premiumProfile
|------(101) quantumProfile
|------------(100) qnoe
|---------------(1) qNoeCommand
|------(102) microProfile
|------------(100) mEtz
|---------------(1) etzIpMgtStatus
|---------------(2) etzIpMgtDhcpTries
|---------------(3) etzIpMgtDhcpMode
|---------------(4) etzRepUserBkups
|---------------(5) etzRepAutoBkups
|---------------(6) etzRepStatus
|---------------(7) etzRepTFPcnxErrors
|---------------(8) etzRepTFPxferErrors
|------(103) momentumIoProfile
|------------(1) momentumIoBaseType
|------------(2) momentumIoBaseName
|------------(3) momentumIoMasterIPTable
|---------------(1) momentumIoMasterIPEntry
|------------------(1) momentumIoMasterIPValue
|------------(4) momentumIoModuleTimeOut
```

```
|------------(5) momentumIoASCIIModuleHeader
|------------(6) momentumIoReservationTime
|------------(7) momentumIoInputDataTable
|---------------(1) momentumIoInputDataEntry
|------------------(1) momentumIoInputDataIndex
|------------------(2) momentumIoInputDataValues
|------------------(3) momentumIoInputDataWords
|------------------(4) momentumIoInputDataPoints
|------------(8) momentumIoOutputDataTable
|---------------(1) momentumIoOutputDataEntry
|------------------(1) momentumIoOutputDataIndex
|------------------(2) momentumIoOutputDataValues
|------------------(3) momentumIoOutputDataWords
|------------------(4) momentumIoOutputDataPoints
|------(104) momentumM1eProfile
|------(105) advantysProfile
|------(106) gatewayProfile
|------(107) modiconM340Profile
|------(255) tfProducts
|------------(1) ety
|------------(2) noe
|------------(3) etz
|------------(4) momentumIo
|------------(5) momentumM1e
|------------(6) altivar
|------------(7) stbNip
|------------(8) tsxntp
|------------(9) nwm
|------------(10) wmy
|------------(11) quantumPLC
|------------(12) premiumPLC
|------------(13) etg
|------------(14) egx
```

```
|------------(15) ecc
|------------(16) cev
|------------(17) inducteIXGKS
|------------(18) ositrackTAP
|------------(19) twidoPLC
|------------(20) modiconM340PLC
|------------(21) modiconM340DPLC
|------------(22) modiconM340CPLC
|------------(23) modiconM340NOE
|---(8) timeManagement
|------(1) ntp
|--------(1) ntpStatus
|--------(2) ntpSrvAddr
|--------(3) ntpLnkSrvStatus
|--------(4) ntpReqCnt
|--------(5) ntpRespCnt
|--------(6) ntpErrCnt
|--------(7) ntpDate
|--------(8) ntpTime
|--------(9) ntpTimeZone
|--------(10) ntpDSTStatus
|--------(11) ntpLastErr
|---(9) email
|------(1) smtp
|---------(1) emailTable
|------------(1) emailEntry
|--------------(1) emailIndex
|--------------(2) smtpStatus
|--------------(3) smtpSrvAddr
|--------------(4) smtpMailSentCnt
|--------------(5) smtpErrCnt
|--------------(6) smtpLastErr
|--------------(7) smtpLastMailElapsedTime
```

```
|--------------(8) smtpLnkSrvStatus
|--------------(9) smtpSrvChkFailCnt
|---(255) tfeMibVersion
|------(1) tfeMibVersionNumber
|------(2) tfeMibVersionDate
```

MIB Subtree Description

Transparent Factory Ethernet Subtree

This topic details some of the objects in the Schneider private MIB tree. The **transparentFactoryEthernet (1)**subtree defines groups that support the TFE services and devices.

Service	Subtree Definition
switch(1) *(see page 405)*	the brand of switches labeled
port502Messaging(2) *(see page 406)*	objects for managing explicit client/server communications to support applications (for example, HMI, SCADA, or programming tools)
ioScanning(3) *(see page 407)*	objects for managing I/O device communications that use the I/O Scanner with the MB/TCP protocol
globalData(4) *(see page 408)*	objects for managing the application coordination service using a publish/subscribe protocol
web(5) *(see page 409)*	objects for managing the activity of the embedded Web servers
addressServer(6) *(see page 410)*	objects for managing the activity of the BOOTP or DHCP servers
equipmentProfile(7) *(see page 411)*	objects for each device type in Transparent Factory Ethernet product portfolio
timeManagement(8) *(see page 413)* (NTP)	objects for managing the UTC time stamp service
email(9) *(see page 414)* (SMTP)	objects for managing the email service
tfeMibVersion(255) *(see page 415)*	the version of the Schneider TFE MIB supported by the product

NOTE: All listed services are not available on all communications modules. Refer to the available services for your module.

When devices are added to the Schneider catalog, the private MIB is extended in the following manner:

- If needed, a Transparent Factory communication-service object is added for the new device in the subtree that corresponds to **equipmentProfile(7)** *(see page 411)*. This subtree can hold as many objects as are required.
- If needed, a new branch is added at the same level as **transparentFactory-Ethernet (1)**. This subtree is created for product-specific objects.

When a new device is added to the catalog a corresponding object description is created in the ASN.1 format. The ASN.1 file(s) are then given to producers of SNMP manager software for inclusion in their products.

Switch Subtree Description

Switch Subtree

The switch (1) subtree, or group, indicates the brand of switches labeled. The following list describes the function of each object.

Service	Indicates . . .
saChassis(1)	configuration of the chassis
saAgent(2)	configuration of Agent
saRingRedundancy(3)	management of Ring Redundancy
saUserGroup(5)	management of user groups
saLLDP(7)	management of proprietary extensions of 802.1AB (station and Media access control Connectivity Discovery)

Port 502 Messaging Subtree Description

Port 502 Messaging Subtree

The port502Messaging (2) subtree, or group, provides connection management and data flow services. The following list describes the function of each object.

Service	Indicates . . .
port502Status(1)	status of the service (idle or operational)
port502SupportedProtocol(2)	supported protocols (MODBUS, X-way, etc.)
port502IpSecurity(3)	status of the Port 502 IP Security service (enabled or disabled)
port502MaxConn(4)	maximum number of TCP connections supported by the Port 502 entity
port502LocalConn(5)	number of TCP connections currently opened by the local Port 502 entity
port502RemConn(6)	number of TCP connections currently opened by the remote entity to the local Port 502 entity
port502IpSecurityTable(7)	a table containing the number of unsuccessful attempts to open a TCP connection from a remote TCP entity
port502ConnTable(8)	a table containing Port 502 TCP specific information (MsgIn, MsgOut)
port502MsgIn(9)	total number of Port 502 messages received from the network
port502MsgOut(10)	total number of Port 502 messages sent from the network
port502MsgOutErr(11)	total number of error messages built by the Port 502 messaging entity and sent to the network
port502AddStackStat(12)	the support of additional Port 502 stack statistics
port502AddStackStatTable(13)	additional stack statistics for Port 502 (optional)

I/O Scanning Subtree Description

I/O Scanning Subtree

The ioScanning (3) subtree, or group, contains the objects related to I/O scanning device management and associated Modbus communications on port 502.

Service	Indicates . . .
ioScanStatus(1)	global status of the I/O scanning service
ioScanMaxDevice(2)	maximum number of devices supported by the I/O scanning entity
ioScanPolledDevice(3)	number of devices currently polled by the I/O scanning entity
ioScanTransSend(4)	total number of transactions sent by the I/O scanning entity
ioScanGlbHealth(5)	global health status for the I/O scanning service
ioScanningDeviceTable(6)	a table containing information on each remote device polled by the I/O scanning entity

Global Data Subtree Description

Global Data Subtree

The globalData (4) subtree, or group, contains the objects related to Global Data.

Service	Indicates . . .
glbDataStatus(1)	global status of the Global Data service
glbDataMaxPub(2)	maximum number of published variables configured by the Global Data entity
glbDataMaxSub(3)	maximum number of subscribed variables configured by the Global Data entity
glbDataPub(4)	total number of publications sent to the network
glbDataSub(5)	total number of subscriptions received from the network
glbDataPubErr(6)	total number of publication errors detected by the local entity
glbDataSubErr(7)	total number of subscription errors detected by the local entity
glbDataGlbSubHealth(8)	global health status of the Global Data service
glbDataPubTable(9)	a table containing information on each published variable (the number of publications, the source IP address, the number of errors, etc.)
glbDataSubTable(10)	a table containing information on each subscribed variable (the number of subscriptions, the source IP address, the number of errors, health, etc.)

Web Subtree Description

Web Subtree

The web (5) subtree, or group, contains the objects related to the Web server service.

Service	Indicates . . .
webStatus(1)	global status of the Web service
webPassword(2)	enable or disable Web passwords
webSuccessfullAccess(3)	total number of successful attempts to access Web site
webFailedAttempts(4)	total number of failed attempts to access Web site

Address Server Subtree Description

Address Server Subtree

The addressServer (6) subtree, or group, contains the objects related to the Address Server. The address server can be either a BOOTP server or a DHCP server.

Service	Indicates . . .
addressServerStatus(1)	global status of the address server service

Equipment Profile Subtree Description

Equipment Profile Subtree

The equipmentProfile (7) subtree contains a set of common objects.

Service	Indicates . . .
profileProductName(1)	the commercial name of the communication product in string form (for example: 140 NOE 771 11, BMX NOE 0100, etc.)
profileVersion(2)	the software version of the communication product in string form (for example, Vx.y or V1.1)
profileCommunicationServices(3)	the communication services supported by the profile (Port502Messaging, I/O scanning Messaging, Global Data, Web, and Address Server)
profileGlobalStatus(4)	the global status of the communication module
profileConfigMode(5)	the IP configuration mode of the communication module
profileRoleName(6)	the role name for the IP address management if it exists (empty string if there is none)
profileBandwidthMgt(7)	the status of Bandwidth Management
profileBandwidthDistTable(8)	the CPU time distribution between Global Data, Port 502 Messaging, I/O scanning
profileLedDisplayTable(9)	a table giving the name and the state of each module's LEDs
profileSlot(10)	the position of the communication module inside the rack if there is one (if there is no rack, the profileSlot value is 0)
profileCPUType(11)	the host for which that communication module is a part when a CPU type exists (if there is no host, the string is empty)
profileTrapTableEntriesMax(12)	the maximum numbers of entries in the Trap Table (equal to the number of possible remote managers)
profileTrapTable(13)	a table allowing you to enable or disable the private traps for each of the communication services
profileSpecificId(14)	a unique Profile Specific Identification inside the equipmentProfile object of the Schneider Transparent Factory MIB (for example, the PLC Premium family is 100)

Service	Indicates . . .
profileIpAddress(15)	the IP address of the SNMP agent
profileIpNetMask(16)	the subnetwork mask associated with the IP address of the SNMP agent (the value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0)
profileIpGateway(17)	the default Gateway IP address of the SNMP agent
profileMacAddress(18)	the Ethernet media-dependent address of the SNMP agent
profileImplementationClass(19)	a textual description of the implementation class supported by the product
premiumProfile(100)	managed products (ETY, ETY port)
quantumProfile(101)	managed products (NOE)
microProfile(102)	managed products (ETZ)
momentumIoProfile(103)	managed products (ENT)
momentumM1eProfile(104)	managed products (M1E)
advantysProfile(105)	managed products (STB NIP)
gatewayProfile(106)	managed products (ETG)
modiconM340profile(107)	managed products (Modicon M340 PLC)
tfProducts(225)	Transparent Factory products

Time Management Subtree Description

Time Management Subtree

The timeManagement (8) subtree contains a set of common NTP objects.

Service	Indicates . . .
ntpStatus(1)	the status of the NTP service (not server)
ntpSrvAddr(2)	the IP address of the NTP server in dot notation format
ntpLnkSrvStatus(3)	the status of the link between the module and the NTP server
ntpReqCnt(4)	the number of requests sent to the NTP server
ntpRespCnt(5)	the number of responses received from the NTP server
ntpErrCnt(6)	the total number of communication errors
ntpDate(7)	date of the day
ntpTime(8)	time of the day
ntpTimeZone(9)	current time zone
ntpDSTStatus(10)	daylight saving time status
ntpLastErr(11)	last error code generated by system

Email Subtree Description

Email Subtree

The email(9) subtree contains a set of common SMTP objects.

Service	Indicates . . .
emailIndex(1)	the index value in the email service table
smtpStatus(2)	the status of SMTP service (not server)
smtpSrvAddr(3)	the IP address of SMTP server in dot notation format
smtpMailSentCnt(4)	the total number of emails sent to the network and successfully acknowledged by the server
smtpErrCnt(5)	the total number of email messages that could not be sent to the network or that have been sent but not acknowledged by the server
smtpLastErr(6)	the error code of the last error that occurred while trying to send an email message to the network
smtpLastMailElapsedTime(7)	the number of elapsed seconds since last successful email was sent to the server
smtpLnkSrvStatus(8)	the status of link with SMTP server
smtpSrvChkFailCnt(9)	the number of times the link to SMTP server is detected as ‘down.’

Transparent Factory MIB Version

tfeMibVersion Subtree

This group contains information about the version of the Schneider TFE MIB *(see page 394)* supported by the product.

Service	Indicates . . .
tfeMibVersionNumber(1)	the version of the SchneiderTFE Mib in Vxx.yy form (example V01.04)
tfeMibVersionDate(2)	the date of last update of the SchneiderTFE MIB in ‘ddMmmyy’ form (example: 09Jan06)

Private Traps and MIB Files

Private Traps and MIB Files

Traps are used to signal status changes to the manager while avoiding additional traffic:

- **LEDs** (`profileLED`): This trap is sent if the LED state changes.
- **communications ports** (`port502StatusChange`): This trap is sent if `port502Status` changes.
- **I/O scanning health value** (`ioScanStatusChange`): This trap is sent if `ioScanStatus` changes.
- **global data health value** (`glbDataStatusChange`): This trap is sent if `glbDataStatus` changes.
- **Web service** (`webStatusChange`): This trap is sent if `webStatus` changes.
- **address server** (`addressServerStatusChange`): This trap is sent if `addressServerStatus` changes.
- **NTP service** (see below)
- **SMTP service** (see below)

Private traps can:

- send messages to the two managers whose IP addresses are configured in the SNMP configuration
- use the community name given to this configuration
- enable or disable each of the Transparent Factory Ethernet Private MIB groups listed in the Transparent Factory Ethernet Subtree *(see page 404)*.

Private traps are described in the MIB ASN.1 description, which is contained in a `.mib` text file.

NTP Traps

- **NTP status** (`ntpStatusChange`): This trap is sent if `ntpStatus` changes.
- **server change** (`ntpServerChange`): This trap is sent if the NTP component switches from the Primary NTP server to the standby NTP server or vice versa.
- **link server status change** (`ntpLnkSrvStatusChange`): This trap is sent if the NTP link server status changes.
- **leap second** (`ntpLeapSecond`): This trap is sent when leap seconds are inserted.
- **DST change** (`ntpDSTChange`): This trap notifies the manager that the NTP server time has changed from either:
  - standard time to daylight savings time, or
  - daylight savings time to standard time

SMTP Traps

- **SMTP status change** (`smtpStatusChange`): This trap is sent if `smtpStatus` of the email service referenced by emailIndex changes.
- **SMTP link to server status** (`smtpLnkSrvChange`): This trap is sent when the `smtpLnkSrvStatus` of the email service referenced by `emailIndex` changes. The trap is sent when the service tries to send an email. Every 30 minutes a periodic test checks the connection to the SMTP server.