

Driver SNMPManager

Nome do Arquivo	SNMPManager.dll
Fabricante	Simple Network Management Protocol
Equipamentos	
Protocolo	SNMP V1 e V2C
Versão	2.0.29
Última Atualização	10/08/2026
Plataforma	Win32 e E3 versão 3.2 ou superior
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31301

Introdução

O Driver SNMPManager comunica com Agentes SNMP através do protocolo SNMP (*Simple Network Management Protocol*). Pela nomenclatura SNMP, **Agentes** são dispositivos de rede, tais como *switches*, roteadores, multiplexadores e *no-breaks*, entre outros, originadores de informações que se deseja gerenciar, e que são portanto enviadas a um sistema **Gerente**, neste caso o Driver SNMPManager.

Atualmente este Driver suporta as versões **V1**, **V2C** e **V3** da especificação SNMP, permitindo os seguintes recursos:

- Comunicação com até 50 equipamentos usando um mesmo Driver (DLL)
- Importação ou criação de Tags dos equipamentos a partir de arquivos no formato **MIB** (*Management Information Base*)
- Leitura assíncrona de variáveis agrupando vários pedidos em uma consulta (*polling*)
- Escrita síncrona de parâmetros
- Suporte a mensagens não solicitadas, ou seja, recepção de *traps* SNMP

NOTA

Ao atingir o limite de 50 equipamentos, este Driver não cria mais instâncias. Se há Tags configurados para estes *hosts*, os valores destes Tags são inválidos. Para endereçar mais Agentes, utilize uma nova instância deste Driver.

Parâmetros de Configuração

Os parâmetros **[P]** não são utilizados por este Driver. As configurações acompanham as da biblioteca **IOKit** da **Elipse Software**, acrescentando as abas de propriedades específicas, a aba **SNMP Device Config**, que configura os dispositivos ou Agentes com os quais ocorre a comunicação, e a aba **SNMP General**, que configura os demais parâmetros deste Driver. Para mais informações sobre as outras abas da janela de configuração deste Driver, consulte o tópico **Documentação das Interfaces de Comunicação**.

Aba SNMP Device Config

Selecione a aba **SNMP Device Config** para configurar os Agentes ou dispositivos, conforme a figura a seguir.

Driver SNMPManager v2.0.27 (IOKit v2.0.147)

SNMP Device Config | SNMP General | Setup | Ethernet

Server Name	IP Address

Add Update Delete

Device: DEVICE001
 IP: 127.0.0.1
 Port: 161
 Max Requests per Packet: 1
 Version: V1
 Read Community: public
 Write Community: public
 MIB Parser: smidump

SNMP v3

Username:
 Context Name:
 Security Level: NoAuthNoPriv
 Auth Passphrase:
 Auth Style: HMAC-MD5
 Priv Passphrase:
 Priv Encrypt Style: DES

Browse MIBs Extra Modules:
 Disable: ☐

OK Cancel Apply

Aba SNMP Device Config

As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba SNMP Device Config

OPÇÃO	DESCRIÇÃO
Add	Adiciona um dispositivo ou Agente
Update	Atualiza as configurações para o dispositivo ou Agente selecionado. Alterações realizadas sem a confirmação através desta opção são descartadas

OPÇÃO	DESCRIÇÃO
Delete	Remove o dispositivo ou Agente selecionado
Device	Nome simbólico de um dispositivo ou Agente. Este nome é utilizado apenas para identificar um dispositivo para este Driver, mapeando-o para o endereço IP e a porta UDP/IP configurados nas opções IP e Port , respectivamente, além de outras configurações
IP	Informa o endereço IP de um Agente
Port	Informa a porta UDP/IP onde um Agente espera por uma conexão. A porta UDP/IP padrão é a 161, mas alguns equipamentos ou simuladores podem usar outras portas UDP/IP
Max Requests per Packet	Quantidade máxima de pedidos de variáveis, ou OIDs, a serem realizados em uma única mensagem para um Agente. Quanto maior o número de pedidos que possam ser realizados a cada comunicação, mais rápida e eficiente é a varredura geral das variáveis configuradas para um equipamento. NOTA: As variáveis que não são declaradas para recepção exclusiva de <i>Traps</i> utilizam o método de <i>polling</i> , respeitando sempre o número máximo de variáveis a cada pedido. Consulte o tópico Referência de Tags para mais informações
Version	Versão do protocolo SNMP usada por um Agente. Atualmente este Driver suporta as versões V1 , V2C e V3
Read Community	Texto usado como parâmetro nas mensagens SNMP para indicar o grupo ou comunidade de variáveis ao qual um pedido pertence. A comunidade mais usual é a pública (<i>public</i>), mas também podem existir variáveis da comunidade privada (<i>private</i>) ou outras criadas por outras empresas ou usuários. Neste caso, digite o nome de uma comunidade. No caso de se utilizar mais de uma comunidade, os nomes devem ser separadas por ponto e vírgula, como por exemplo "public;communityA;communityB". A comunidade que é usada para cada Tag, que só pode ser uma, é definida pelo parâmetro <i>N2</i> , em que o valor 0 (zero) é a primeira comunidade desta lista, o valor 1 (um) é a segunda comunidade desta lista e assim por diante
Write Community	Informe a comunidade padrão que é usada para escrita. Só é aceito um valor de comunidade para escrita
MIB Parser	Até a versão 2.0.21 , este Driver utilizava apenas o interpretador smidump para arquivos no formato MIB , gerando a visualização dos Tags que podem ser importados para uma aplicação pelo Tag Browser. A partir da versão 2.0.22 , também é possível usar o interpretador snmptranslate , que pode gerar resultados diferentes dependendo do arquivo no formato MIB

OPÇÃO	DESCRIÇÃO
Browse MIBs	Abre uma caixa de diálogo para selecionar arquivos no formato MIB para importação e que fornecem padrões de Tags
Extra Modules	Lista dos módulos importados usando a opção Browse MIBs
Disable	Desabilita este dispositivo, não estabelecendo comunicação com este Driver ao iniciar, e também esconde um Agente na lista de dispositivos da janela Tag Browser
SNMP v3 - UserName	Nome de usuário a ser utilizado quando a versão selecionada é V3 . Deve ser compatível com o nome de usuário definido em um Agente
SNMP v3 - Context Name	Nome de contexto a ser utilizado quando a versão selecionada é V3 . Deve ser compatível com o nome de contexto definido em um Agente
SNMP v3 - Security Level	Define as opções de autenticação e privacidade. A opção de autenticação utiliza um mecanismo de verificação (<i>hash</i>) para saber se uma mensagem não foi alterada no processo de transmissão. A opção de privacidade significa que, além da autenticação, os dados estão criptografados e portanto não visíveis para quem observa a comunicação. Só é possível ter privacidade se há autenticação, por isto as opções possíveis são NoAuthNoPriv : Não há nenhuma funcionalidade de segurança e o protocolo se comporta de forma similar à versão V2C , AuthNoPriv : Utiliza autenticação sem privacidade ou AuthPriv : Utiliza autenticação e privacidade
SNMP v3 - Auth Passphrase	Informa a senha utilizada para a autenticação. Esta senha deve coincidir com a senha definida em um Agente
SNMP v3 - Auth Style	Algoritmo para autenticação. Este algoritmo é usado também para transformar as senhas de autenticação e de privacidade em uma chave privada, de forma a não expor esta senha em formato de texto. As opções disponíveis são HMAC-MD5 , HMAC-SHA1-96 ou HMAC-SHA2-256
SNMP v3 - Priv Passphrase	Informa a senha utilizada para a privacidade. Esta senha deve coincidir com a senha definida em um Agente
SNMP v3 - Priv Encrypt Style	Algoritmo para criptografia. As opções disponíveis são DES , 3DES , AES128 , AES196 ou AES256

Informações Importantes sobre Algoritmos de Autenticação e Criptografia no Protocolo SNMP V3

Atualmente a especificação oficial do protocolo SNMP define o uso apenas dos algoritmos **MD5** e **SHA1-96** para autenticação e os algoritmos **DES** e **AES128** para criptografia. Entretanto, com o tempo novos algoritmos foram desenvolvidos e que utilizam chaves maiores para evitar a violação da comunicação, como por exemplo utilizando o método de força bruta, e muitos fabricantes adotam por conta própria estes novos algoritmos.

No caso de selecionar uma opção de autenticação que gere uma chave menor que o mínimo necessário para um algoritmo de criptografia, este Driver utiliza um método de extensão de chave, descrito no documento Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in "Outside" CBC Mode e utilizado pela Cisco e outros fabricantes. Note que este mecanismo não é utilizado quando a autenticação é igual a **SHA2-256**, pois esta autenticação já produz uma chave de tamanho adequado. As tabelas a seguir contêm os tamanhos padrão de chaves para cada algoritmo de criptografia.

Tamanhos de chaves de autenticação

AUTENTICAÇÃO	TAMANHO
MD5	16 bytes
SHA1-96	20 bytes
SHA2-256	32 bytes

Tamanhos de chaves de criptografia

CRIPTOGRAFIA	TAMANHO
DES	Necessita de uma chave de 16 bytes
3DES	Necessita de uma chave de 32 bytes
AES128, AES192 e AES256	Necessitam de chaves de 16, 24 e 32 bytes, respectivamente

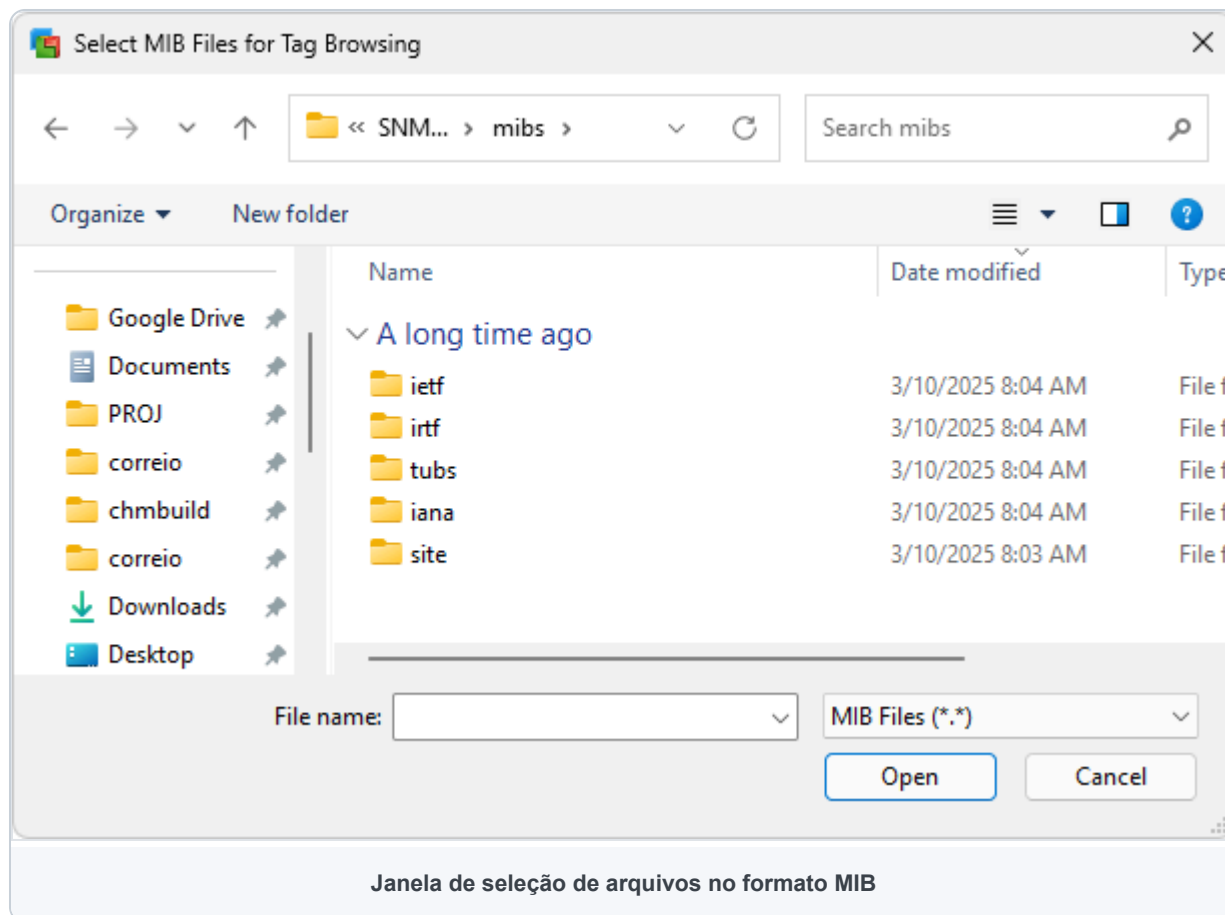
Portanto, ao selecionar a opção de autenticação **MD5** e a opção de criptografia **3DES**, o algoritmo estende a chave automaticamente de 16 para 32 bytes.

Arquivos MIB

Os Tags de um Agente podem ser criados importando arquivos no formato **MIB** (*Management Information Base*), obtidos de domínio público e padronizados conforme as normas IETF, IRTF e IANA, entre outras, ou fornecidos pelos fabricantes dos Agentes e dos equipamentos.

Clique em **Browse MIBs** para abrir uma caixa de diálogo para selecionar arquivos no formato **MIB** para importação. Após selecionar os arquivos, cria-se uma entrada na seção de declaração de dispositivos correspondendo a um Agente, além de uma pasta na janela Tag Browser contendo todos os Tags identificados.

Junto com este Driver é fornecida uma pasta **mibs** com um conjunto bastante completo e atualizado de arquivos no formato **MIB** padrão, conforme mostrado na figura a seguir.



Caso seja necessário importar arquivos no formato **MIB** fornecidos por fabricantes de equipamentos ou por terceiros, estes devem obrigatoriamente ser copiados para a pasta **mibs/site**.

Para que esta importação aconteça, cada arquivo no formato **MIB** deve ter exatamente o mesmo nome do módulo que este arquivo contém, ou seja, o nome encontrado antes da palavra-chave **DEFINITIONS** no início deste arquivo, sem nenhuma extensão ou caracteres adicionais.

Após a seleção dos arquivos no formato **MIB**, a importação é executada e, se bem-sucedida, a lista **Extra Modules** do equipamento selecionado é acrescida dos novos módulos, como uma lista separada por ponto e vírgula. Esta operação equivale a editar manualmente esta lista e clicar em **Update**.

Estes módulos estão disponíveis na janela Tag Browser do **E3**, **Power** ou **Water** como pastas contendo todos os objetos importados e mantendo a hierarquia de módulos, nós, tabelas, linhas, colunas, grupos, notificações e escalares de cada arquivo no formato **MIB** em cada dispositivo.

NOTA

Para adicionar um mesmo arquivo no formato **MIB** em vários equipamentos, acrescente este arquivo na opção **Extra Modules** de cada um destes Agentes.

O processo de importação disponibiliza os Tags com os parâmetros descritos na tabela a seguir.

Parâmetros de Tags do processo de importação

PARÂMETRO	DESCRIÇÃO
Name	Nome simbólico padrão de um escalar ou coluna de tabela, conforme obtido de um arquivo no formato MIB
ParamDevice	Nome do Agente definido pelo usuário na lista de equipamentos ou Agentes
ParamItem	Valor numérico padrão do endereço OID (<i>Object Identifier</i>) já no formato ASN.1 (<i>Abstract Syntax Notation</i>), usado pelo protocolo SNMP e obtido de um arquivo no formato MIB

Quando um Tag referencia um objeto escalar, ou seja, uma instância simples, o endereço OID tem o final ".0". Quando um Tag representa uma coluna de uma tabela, este Tag é criado com um índice padrão 1 (um), indicado por colchetes, ou seja, [1].

As tabelas no protocolo SNMP podem conter n linhas. Os Tags, que são as colunas, são portanto criados com o endereço da linha 1 (um) por padrão. No caso de Tags para outras linhas, modifique o índice ou criar cópias de um Tag apontando para outros índices, conforme os exemplos a seguir.

```
'Tag padrão para uma coluna chamada lldpPortNum:  
lldpPortNum.1 = 1.0.8802.1.1.2.1.3.7.1.1[1]  
  
'Para Tags para outras linhas, podem ser criadas cópias:  
lldpPortNum.0 = 1.0.8802.1.1.2.1.3.7.1.1[0]  
lldpPortNum.2 = 1.0.8802.1.1.2.1.3.7.1.1[2]  
lldpPortNum.3 = 1.0.8802.1.1.2.1.3.7.1.1[3]
```

Caso uma tabela não contenha uma determinada célula, a qualidade deste Tag é configurada como **Ruim**.

Aba SNMP General

Selecione a aba **SNMP General** para configurar as opções gerais deste Driver.

Driver SNMPManager v2.0.27 (IOKit v2.0.147)

SNMP Device ConfigSNMP GeneralSetupEthernet

General Timeout:5000

☐ Listen for SNMP Traps

Trap Port:162

OKCancelApply

Aba SNMP General

As opções disponíveis nesta aba estão descritas na tabela a seguir.

Opções disponíveis na aba SNMP General

OPÇÃO	DESCRIÇÃO
General Timeout	Tempo de espera, em milissegundos, para uma resposta de uma leitura (<i>SNMP Get</i>) ou escrita (<i>SNMP Set</i>). Este tempo de espera byte a byte é definido na aba Setup da biblioteca IOKit . Ambos os <i>time-outs</i> devem ser configurados, e o valor desta opção deve ser sempre superior ao <i>time-out</i> da biblioteca IOKit . Esta opção pode ser modificada em tempo de execução através de um Tag de configuração chamado SNMP.Timeout . Para mais informações, consulte o tópico Documentação das Interfaces de Comunicação
Listen for SNMP Traps	Instrui este Driver a criar um processo específico para tratar as mensagens não solicitadas (<i>SNMP Trap</i>) enviadas na ocorrência de eventos específicos que podem ser configurados em alguns Agentes. As informações obtidas nas mensagens de <i>trap</i> são repassadas aos Tags com o mesmo Device e Item descrito nesta mensagem. Esta opção pode ser modificada em tempo de execução através de um Tag de configuração chamado SNMP.UseTraps . Para mais informações, consulte o tópico Documentação das Interfaces de Comunicação

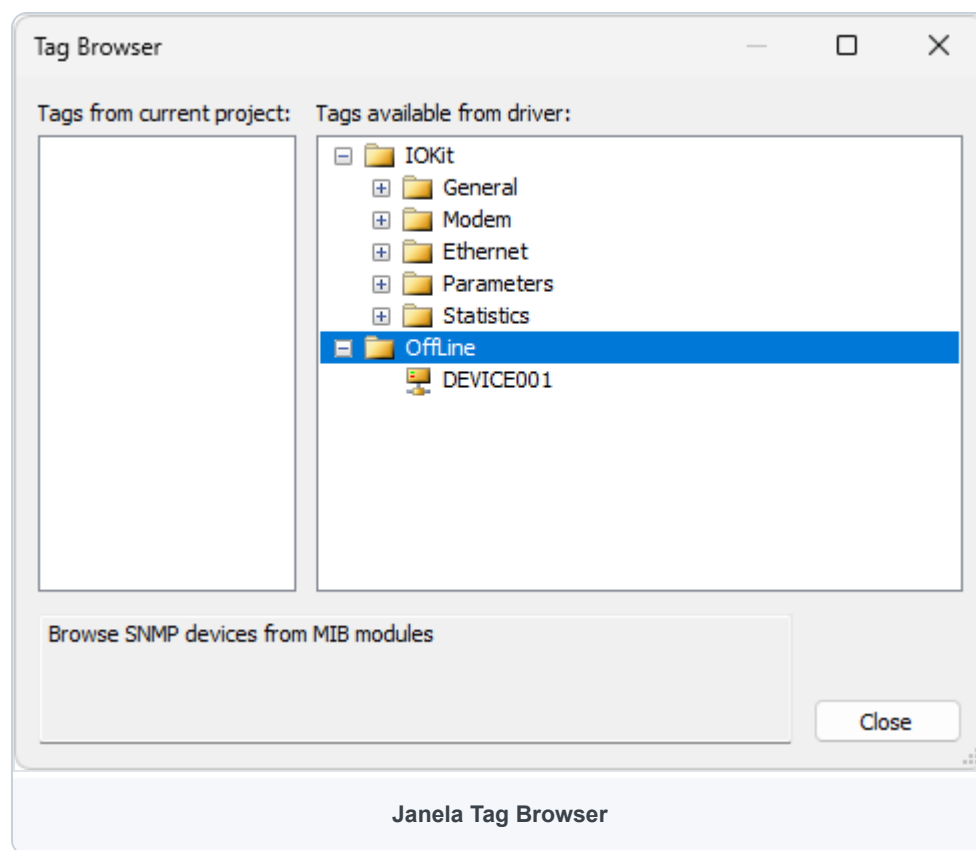
OPÇÃO	DESCRIÇÃO
Trap Port	Indica a porta UDP/IP local onde as mensagens de <i>trap</i> são recebidas. A porta UDP/IP padrão é a 162. Esta opção pode ser modificada em tempo de execução através de um Tag de configuração chamado SNMP.TrapPort . Para mais informações, consulte o tópico Documentação das Interfaces de Comunicação

Janela Tag Browser

Clique em **Tag Browser**



no **E3**, **Power** ou **Water** para abrir a janela da figura a seguir, que permite expandir, listar e arrastar para este Driver os Tags identificados em um Agente. Selecione, na opção MIB Parser de cada equipamento, o interpretador usado, **smidump** ou **snmptranslate**.



Esta janela exibe a pasta **IOKit**, que contém Tags padrão da biblioteca **IOKit** e que permitem ler e escrever parâmetros gerais ou de status, por exemplo, e a pasta **Offline**, que contém os Tags dos Agentes, conforme os módulos importados de arquivos no formato **MIB**.

Para utilizar este Tags em uma aplicação, arraste um Tag ou pasta da lista **Tags available from driver** (*Tags disponibilizados pelo driver*) para a lista **Tags from current project** (*Tags do projeto corrente*).

Referência de Tags

Os parâmetros a seguir são utilizados para os Tags:

- **Device:** Nome de um Agente cadastrado na aba SNMP Device Config
- **Item:** Identificador de um objeto ou OID. Este parâmetro pode conter ainda os seguintes formatadores:
 - "ReadCommunity:OID": Informação sobre uma comunidade específica para leitura
 - "TableOID[Row]": Ao utilizar um bloco apontando para o identificador OID de uma tabela, pode-se especificar entre colchetes uma linha específica desta tabela para ser retornada na leitura, descartando outras linhas encontradas
 - "WalkTable:TableOID": Instrução para obter todas as células de uma tabela SNMP em um único Tag
- EnterpriseOID.<sufixo>: Quando utilizado com mensagens de *trap*, os sufixos a seguir podem ser usados para extrair metadados específicos da mensagem de *trap* recebida:
 - Source: Retorna o endereço IP do Agente que enviou uma mensagem de *trap*
 - Generic: Retorna o código *Generic-Trap*, um tipo genérico da mensagem de *trap* SNMP versão 1, entre 0 (zero) e 6 (seis)
 - Specific: Retorna o código *Specific-Trap*, um tipo específico definido por um fabricante
 - Timestamp: Retorna a estampa de tempo (*sysUpTime*), em segundos, do momento em que uma mensagem de *trap* foi gerada
- **N1:** Se este parâmetro é diferente de 0 (zero), indica que um Tag só é atualizado através de uma mensagem de *trap*, não efetuando a comunicação por *polling* pela taxa de varredura definida neste Tag
- **N2:** Permite indicar uma das comunidades declaradas na opção Read Community de um equipamento, utilizando um índice a partir de 0 (zero) para a leitura. Por exemplo, ao declarar em um equipamento as comunidades "A;B;C" e configurar em um Tag o parâmetro N2 igual a 2 (dois), então a comunidade "C" é usada neste Tag. Este parâmetro é usado apenas para leitura, e a comunidade de escrita é definida na opção Write Community

Opções disponíveis para os parâmetros dos Tags

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
Nome de um Agente, como por exemplo "router-001"	OID, como por exemplo 1.3.6.1.2.1.37.1.2.1.1.1.0	Leitura e escrita	Tag qualquer que é lido de acordo com a varredura configurada. Caso o parâmetro N1 seja diferente de 0 (zero), este Tag só é atualizado quando recebe uma mensagem de <i>trap</i> com a variável indicada. O identificador OID que termina com ".0" representa uma variável escalar. O identificador OID especificado com um índice entre colchetes corresponde a uma coluna de uma tabela, com o índice entre colchetes representando uma linha. Neste caso é realizada uma busca de todos os OIDs lexicograficamente encontrados nesta coluna da tabela,

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			retornando o resultado da mesma forma que na opção WalkTable informada a seguir
Idem ao anterior	Community:OID	Somente leitura	Indique o nome de uma comunidade para ser utilizada nesta leitura, como por exemplo "public:1.3.6.1.2.1.37.1.1.0"
Idem ao anterior	TableOID	Somente leitura	A Janela Tag Browser oferece os objetos do tipo Tabela do protocolo SNMP como Tags Bloco, em que cada Elemento corresponde a uma coluna de uma tabela. O parâmetro Item é configurado com o identificador OID desta tabela. Ao realizar a leitura de um Tag Bloco, este Driver busca e retorna todas as linhas encontradas nesta tabela. O usuário pode tratar o conteúdo de cada linha, com o valor de cada uma das respectivas colunas, usando um evento OnRead deste Tag Bloco. Portanto, se esta tabela tem 5 (cinco) linhas, o evento OnRead deste Tag Bloco é disparado 5 (cinco) vezes
Idem ao anterior	TableOID[Row]	Somente leitura	Da mesma forma que no item anterior, ao adicionar no parâmetro Item de um Tag Bloco associado a uma tabela um índice entre colchetes, a leitura retorna apenas a linha selecionada, caso exista nesta tabela
Idem ao anterior	WalkTable:OID	Somente leitura	Esta instrução indica que este Driver deve varrer todos os identificadores OIDs lexicograficamente encontrados a partir do identificador OID inicial informado no parâmetro Item , usando o comando SNMP GetNextRequest . O resultado é retornado em um Tag como uma String composta de uma sequência de combinações "OID = valor" separadas por ponto e vírgula,

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			como por exemplo "1.3.6.1.2.1.37.1.2.1.33.1 = 1260; 1.3.6.1.2.1.37.1.2.1.33.2 = 'Fast Ethernet Switch'"
Idem ao anterior	EnterpriseOID.Source	Somente leitura	Uso exclusivo com mensagens de <i>trap</i> , ou seja, com o parâmetro <i>N1</i> diferente de 0 (zero). Retorna o endereço IP do Agente que enviou uma mensagem de <i>trap</i> . O valor <i>EnterpriseOID</i> refere-se ao identificador da organização ou fabricante configurado em um equipamento. Este sufixo extrai metadados de uma mensagem de <i>trap</i> recebida, não o valor da variável em si
Idem ao anterior	EnterpriseOID.Generic	Somente leitura	Uso exclusivo com mensagens de <i>trap</i> , ou seja, com o parâmetro <i>N1</i> diferente de 0 (zero). Retorna o código <i>Generic-Trap</i> de uma mensagem SNMP versão 1 (um). Os valores possíveis são 0 : coldStart, 1 : warmStart, 2 : linkDown, 3 : linkUp, 4 : authenticationFailure, 5 : egpNeighborLoss ou 6 : enterpriseSpecific
Idem ao anterior	EnterpriseOID.Specific	Somente leitura	Uso exclusivo com mensagens de <i>trap</i> , ou seja, com o parâmetro <i>N1</i> diferente de 0 (zero). Retorna o código <i>Specific-Trap</i> definido por um fabricante. Este valor identifica tipos de eventos específicos de um equipamento quando o código <i>Generic-Trap</i> é igual a 6 (seis, <i>enterpriseSpecific</i>)
Idem ao anterior	EnterpriseOID.Timestamp	Somente leitura	Uso exclusivo com mensagens de <i>trap</i> , ou seja, com o parâmetro <i>N1</i> diferente de 0 (zero). Retorna a estampa de tempo (<i>sysUpTime</i>), em segundos, do momento em que um evento <i>trap</i> foi gerado em um Agente. Representa o tempo decorrido desde a última reinicialização de um equipamento

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
Idem ao anterior	ServerStatus	Somente leitura	Retorna o status interno da conexão com um Agente. Os valores possíveis são 0 : Sem comunicação ou 1 : Comunicando normalmente
Nome qualquer, que não pode ser uma String vazia	UpdateConfig	Somente escrita	Recria todas as conexões com cada equipamento, a partir da definição corrente de parâmetros, que pode ser alterada através da escrita de um <i>array</i> de parâmetros e valores. Para mais informações, consulte o tópico Atualização de Parâmetros em Tempo de Execução

Tag Bloco para Mensagens de Trap SNMP Versão 1

É possível configurar um Tag Bloco para capturar em uma única leitura todas as informações de uma mensagem de *trap* SNMP versão 1 recebida. Para isto, configure o parâmetro *B1* diferente de 0 (zero) e o parâmetro *Item* com o EnterpriseOID da mensagem de *trap*, sem sufixo. Esta abordagem garante que todos os dados pertencem a um mesmo evento de *trap*, eliminando possíveis inconsistências causadas por leituras individuais em Tags separados. Este Tag Bloco contém os Elementos descritos na tabela a seguir.

Elementos de um Tag Bloco para mensagens de trap SNMP versão 1

ELEMENTO	CONTEÚDO
0	Source : Endereço IP de um Agente que enviou uma mensagem de <i>trap</i>
1	Generic : Código <i>Generic-Trap</i> . Os valores possíveis são 0 : coldStart, 1 : warmStart, 2 : linkDown, 3 : linkUp, 4 : authenticationFailure, 5 : egpNeighborLoss ou 6 : enterpriseSpecific
2	Specific : Código <i>Specific-Trap</i> definido por um fabricante
3	Timestamp : Estampa de tempo (<i>sysUpTime</i>), em segundos, do momento em que uma mensagem de <i>trap</i> foi gerada em um Agente
4, 6, 8, ...	OID/Name de cada variável recebida em uma mensagem de <i>trap</i> (índices pares a partir de quatro)
5, 7, 9, ...	Valor de cada variável recebida em uma mensagem de <i>trap</i> (índices ímpares a partir de cinco)

Atualização de Parâmetros em Tempo de Execução

Para adicionar, remover ou alterar algum parâmetro em tempo de execução de um equipamento ou Agente, execute os procedimentos a seguir:

1. Crie um Tag chamado "UpdateConfig" e informe na propriedade **ParamDevice** o nome de um dispositivo qualquer, como por exemplo "Teste", pois este valor não pode ser uma **String** vazia. Configure a propriedade **ParamItem** com o valor "UpdateConfig".
2. Crie em uma aplicação **E3**, **Power** ou **Water** um script para a escrita dos parâmetros que devem ser atualizados usando o Tag **SetConfigurationParameters** da biblioteca **IOKit**. Para mais informações, consulte o tópico Documentação das Interfaces de Comunicação.
3. Os nomes dos parâmetros utilizados por este Driver estão descritos na tabela a seguir e armazenam o resultado da configuração realizada na aba SNMP Device Config da janela de configuração.

Parâmetros utilizados por este Driver

PARÂMETRO	DESCRIÇÃO
SNMP.DeviceCount	Indica o número de equipamentos de uma lista, em que cada equipamento é endereçado por um índice começando em 0 (zero). Este índice é necessário para todas as demais propriedades
SNMP.Device[Index].Name	Nome de um equipamento conforme o índice, como por exemplo SNMP.Device[0].Name = "MainSwitch"
SNMP.Device[Index].IP	Endereço IP
SNMP.Device[Index].Port	Porta UDP/IP
SNMP.Device[Index].Community	Comunidade de leitura
SNMP.Device[Index].WCommunity	Comunidade de escrita
SNMP.Device[Index].Version	Versão do protocolo SNMP
SNMP.Device[Index].MaxReq	Número máximo de pedidos simultâneos
SNMP.Device[Index].Modules	Lista de módulos MIB
SNMP.Device[Index].Disable	Os valores possíveis são 1 : Desabilita a comunicação com este equipamento ou 0 : Habilita a comunicação com este equipamento

No exemplo de script a seguir, a comunicação é habilitada com um equipamento de índice 1 (um) e configurado o endereço IP deste equipamento. Após enviar estes parâmetros usando uma escrita no Tag **SetConfigurationParameters**, é solicitado o reinício da comunicação escrevendo no Tag **UpdateConfig**.

```
Dim CommArr(1)
CommArr(0) = Array("SNMP.Device[1].Disable",0)
CommArr(1) = Array("SNMP.Device[1].IP","192.168.100.2")
Item("DriverSNMP").Write -1,0,0,3,CommArr
Item("DriverSNMP").Item("UpdateConfig").WriteEx(1)
```

Informações Adicionais sobre o Protocolo SNMP

O protocolo **SNMP** é um protocolo de gerência típica de redes TCP/IP, da camada de aplicação, que facilita o intercâmbio de informação entre dispositivos de rede, tais como roteadores, *switches* e estações de trabalho (*hosts*). Este protocolo possibilita aos administradores de rede gerenciar a conectividade e o desempenho de uma rede, encontrar e resolver eventuais problemas e fornecer informações para o planejamento de expansão, dentre outras possibilidades.

O programa de gerência de redes não segue o modelo Cliente-Servidor convencional, pois para algumas operações a estação de gerenciamento se comporta como Cliente e o dispositivo de rede a ser analisado ou monitorado se comporta como Servidor, enquanto que em uma operação *trap* ocorre o oposto, pois no envio de alarmes é o dispositivo gerenciado que toma a iniciativa da comunicação. Por conta disto, os sistemas de gerência de redes evitam os termos Cliente e Servidor, optando por usar **Gerente** para uma aplicação que executa na estação de gerenciamento e **Agente** para uma aplicação que executa no dispositivo de rede.

Perfil NTCIP

Quando um equipamento suporta o protocolo NTCIP (*National Transportation Communications for Intelligent Transportation System*), algumas funções especiais são disponibilizadas. Este Driver atualmente suporta apenas as funções DMS (*Dynamic Message Sign*).

Ao configurar um dispositivo, se um usuário informa na opção **Extra Modules** o nome de módulos **MIB** que começam com a palavra "NTCIP", então este dispositivo é do tipo **NTCIP**.

Os Tags especiais do protocolo NTCIP estão disponíveis na janela Tag Browser, em que são exibidos tanto os Tags normais do protocolo SNMP quanto os Tags especiais na pasta **NTCIP**.

Tags NTCIP

A tabela a seguir contém a descrição dos Tags suportados por este Driver para o perfil **DMS** do protocolo NTCIP.

Tags suportados para o perfil DMS do protocolo NTCIP

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
Nome de um Agente, como por exemplo "device-001"	NTCIP.DMS.GetCurrentMessage	Somente leitura	Obtém a mensagem atual exibida em um painel. Este Tag Bloco contém 6 (seis) Elementos. Os Elementos são Elemento 1 : Texto exibido no formato MULTI String, Elemento 2 : Tempo

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			restante, em minutos (o valor 65535 indica que não há limite de tempo), Elemento 3: Prioridade, entre 1 (um) e 255, Elemento 4: Status. Os valores possíveis são 1: Not used, 2: Modifying, 3: Validating, 4: Valid, 5: Error, 6: ModifyReq, 7: ValidateReq ou 8: NotUsedReq, Elemento 5: Nome do usuário que inseriu uma mensagem e Elemento 6: Origem de uma mensagem. Os valores possíveis são 1: SourceModeOther, 2: SourceModeLocal, 3: SourceModeExternal, 8: SourceModeCentral, 9: SourceModeTimeBasedScheduler, 10: SourceModePowerRecovery, 11: SourceModeReset, 12: SourceModeCommLoss, 13: SourceModePowerLoss ou 14: SourceModeEndDuration
Idem ao anterior	NTCIP.DMS.GetMessage	Somente leitura	Obtém uma mensagem qualquer, em que o parâmetro <i>B1</i> é o tipo de memória e o parâmetro <i>B2</i> é um índice, entre 1 (um) e 255. Os tipos de memória disponíveis são 2: MemoryTypePermanent, 3: MemoryTypeChangeable, 4: MemoryTypeVolatile, 5: MemoryTypeCurrentBuffer, 6: MemoryTypeSchedule ou 7: MemoryTypeBlank. Este Tag Bloco contém 4 (quatro) Elementos. Os Elementos são Elemento 1: Texto exibido no formato MULTI String, Elemento 2: Prioridade, entre 1 (um) e 255, Elemento 3: Status. Os valores possíveis são 1: Not Used, 2: Modifying, 3: Validating, 4: Valid, 5: Error, 6: ModifyReq, 7: ValidateReq ou 8: NotUsedReq e Elemento 4: Nome do usuário que inseriu uma mensagem

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
Idem ao anterior	NTCIP.DMS.DefineMessage	Somente escrita	Define uma mensagem que posteriormente pode ser ativada. Este Tag Bloco possui 7 (sete) Elementos. Os Elementos são Elemento 1 : Tipo de memória. Os valores possíveis são 2 : MemoryTypePermanent, 3 : MemoryTypeChangeable ou 4 : MemoryTypeVolatile, Elemento 2 : Índice de uma mensagem, entre 1 (um) e 255, Elemento 3 : Texto exibido no formato MULTI String, Elemento 4 : Prioridade, entre 1 (um) e 255, Elemento 5 : Nome do usuário que inseriu uma mensagem, Elemento 6 : Indica se os <i>beacons</i> são acionados ou não. Os valores possíveis são 0 (zero, não aciona) ou 1 (um, aciona), Elemento 7 : Indica se o serviço de pixel é acionado ou não. Os valores possíveis são 0 (zero, não aciona) ou 1 (um, aciona)
Idem ao anterior	NTCIP.DMS.ActivateMessage	Somente escrita	Ativa uma mensagem já existente. Este Tag Bloco possui 4 (quatro) Elementos. Os Elementos são Elemento 1 : Tipo de memória. Os valores possíveis são 2 : MemoryTypePermanent, 3 : MemoryTypeChangeable ou 4 : MemoryTypeVolatile, Elemento 2 : Índice de uma mensagem, entre 1 (um) e 255, Elemento 3 : Prioridade, entre 1 (um) e 255 e Elemento 4 : Duração, em minutos (use o valor 65535 para uma mensagem permanente)
Idem ao anterior	NTCIP.DMS.LongPowerMessage	Leitura ou escrita	Permite ler ou definir uma mensagem ativada após uma queda de energia. Os parâmetros <i>N1</i> e <i>N2</i> são usados somente na escrita, em que o parâmetro <i>N1</i> é o tipo de memória e o parâmetro <i>N2</i> é um índice entre 1 (um) e 255. Os valores possíveis para os tipos de memória são 2 :

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			MemoryTypePermanent, 3 : MemoryTypeChangeable ou 4 : MemoryTypeVolatile. Na escrita este Tag aceita qualquer valor para iniciar a operação de escrita. Na leitura este Tag retorna um valor no formato MMXXXXCCCC , em que MM é um tipo de memória com 8 (oito) bits, XXXX é o índice de memória com 16 bits e CCCC é o código CRC (<i>Cyclic Redundancy Code</i> ou Verificação de Redundância Cíclica) de uma mensagem
Idem ao anterior	NTCIP.DMS.CommLostMessage	Leitura ou escrita	Permite ler ou definir qual mensagem é ativada após uma queda de comunicação. Os parâmetros <i>N1</i> e <i>N2</i> são usados apenas na escrita, em que o parâmetro <i>N1</i> é o tipo de memória e o parâmetro <i>N2</i> é um índice entre 1 (um) e 255. Os valores possíveis para os tipos de memória são 2 : MemoryTypePermanent, 3 : MemoryTypeChangeable ou 4 : MemoryTypeVolatile. Na escrita este Tag aceita qualquer valor para iniciar a operação de escrita. Na leitura este Tag retorna um valor no formato MMXXXXCCCC , em que MM é um tipo de memória com 8 (oito) bits, XXXX é o índice de memória com 16 bits e CCCC é o código CRC (<i>Cyclic Redundancy Code</i> ou Verificação de Redundância Cíclica) de uma mensagem

Formato MULTI String

O texto a ser exibido em um painel DMS segue o formato **MULTI String** (*Markup Language for Transportation Information*). Esta linguagem é similar à linguagem HTML, em que texto é transmitido e Tags definem como este texto é exibido. Tags são colocados entre delimitadores, contêm um identificador de 1 (um) ou mais caracteres e quaisquer parâmetros opcionais necessários para este Tag.

Cada Tag no formato **MULTI** começa com um colchete à esquerda ([) e termina com um colchete à direita (]). O identificador de um Tag aparece depois do colchete à esquerda ([) e tem 1 (um) ou mais caracteres, sem distinção entre maiúsculas e minúsculas. Se um Tag tem parâmetros, então estes parâmetros seguem imediatamente após o identificador deste Tag e também não distinguem

entre maiúsculas e minúsculas, exceto quando especificado. Não devem ser inseridos espaços ou outros caracteres separadores entre um identificador e os parâmetros. Alguns Tags podem operar em pares e a notação padrão de um Tag é definida no Tag de abertura. Um Tag de abertura define onde a funcionalidade deste Tag começa. Um Tag de fechamento define onde a funcionalidade deste Tag termina, e é definido como um Tag de abertura com uma barra à esquerda precedendo o identificador, ou seja, se um Tag de abertura de *flash* é igual a **[fl]**, então o Tag de fechamento de *flash* é igual a **[/fl]**. O código a seguir contém um exemplo de uma mensagem no formato **MULTI**:

```
[pt50o0][fo6][j13]This[n120][j13]is[n120][j13]a Test
```

Tags disponíveis no formato MULTI String

TAG DE ABERTURA	TAG DE FECHAMENTO	DESCRIÇÃO
cbx		Cor de fundo de uma mensagem
pbz ou pbr,g,b		Cor de fundo de uma página
cfx ou cfr,g,b		Cor do texto de uma mensagem
crx,y,w,h,r,g, b ou crx,y,w,h,z		Cor de uma área retangular da página atual de uma mensagem
fx,y		Campo incorporado em uma mensagem com base nos dados de um equipamento, como por exemplo um relógio, um sensor de temperatura ou um detector, entre outros
fltxoy ou floytx	/fl	Ativa o piscar de um texto, definindo o número de vezes que o texto pisca e a ordem de piscar (ligar e desligar ou desligar e ligar)
fox ou fox,cccc		Permite selecionar o código de uma fonte para visualizar uma mensagem. O parâmetro opcional cccc indica a versão da fonte
gn ou gn,x,y ou gn,x,y,cccc		Seleciona uma imagem para inserir na mensagem. Uma imagem é tratada como um único caractere imprimível. Pode exigir alguns pixels ou o sinal completo para ser mostrado. O parâmetro opcional cccc indica o identificador de uma imagem
hcx		Valor hexadecimal de um caractere para ser mostrado
jlx		Especifica o alinhamento do texto. Os valores possíveis são left : Alinhamento à esquerda, center : Texto centralizado, right : Alinhamento à direita ou full : Texto justificado

TAG DE ABERTURA	TAG DE FECHAMENTO	DESCRIÇÃO
jpx		Especifica o alinhamento da página. Os valores possíveis são top : Alinhamento pelo topo, middle : Alinhamento pelo centro ou bottom : Alinhamento pela base
msx,y	/msx,y	Especifica um Tag específico de um fabricante
mvtdw,s,r,text		Especifica os parâmetros para um texto que pode ser rolado
nlx		Especifica o início de uma nova linha
np		Especifica o início de uma nova página
ptxoy		Especifica os tempos de uma página
scx /sc		Especifica o espaçamento entre caracteres
trx,y,w,h		Especifica o posicionamento de um texto em um display

Documentação das Interfaces de Comunicação

Esta seção contém a documentação das Interfaces de Comunicação referentes ao Driver **SNMPManager**.

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.29	10/08/2026	A. Fetzner M. Ludwig	- Implementado suporte à leitura em bloco de mensagens de <i>trap</i> SNMP versão 1 (<i>Case 39394</i>). - Implementado suporte à leitura de metadados de mensagens de <i>trap</i> SNMP versão 1 (<i>Case 39340</i>). - Corrigido um problema de <i>buffer overflow</i> ao decodificar um <i>Object ID</i> para String (<i>Case 39319</i>). - Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 38020</i>).
2.0.27	10/03/2025	M. Salvador	Corrigidas as escritas ou comandos (<i>Case 37437</i>).
2.0.26	21/03/2024	M. Salvador	Corrigida a leitura de tabelas (<i>Case 35438</i>).
2.0.24	26/06/2023	M. Salvador	Implementada uma segunda opção de MIB Parser por equipamento. Agora é possível escolher entre smidump e snmptranslate (<i>Case 33723</i>).
2.0.23	26/06/2023	M. Salvador	Implementado o Perfil NTCIP DMS (<i>Dynamic Message Sign</i>), utilizado em rodovias (<i>Case 33727</i>).
2.0.19	11/11/2021	M. Salvador	Adicionado suporte ao protocolo SNMP versão 3 (<i>Case 18316</i>).
2.0.16	05/08/2019	M. Ludwig	Driver portado para o Visual Studio 2017 (<i>Case 27093</i>).
2.0.15	03/04/2019	M. Ludwig F. Englert M. Salvador	- Melhorado o diagnóstico de erros de carga de arquivos MIB (<i>Case 23007</i>). - Implementadas diversas melhorias na importação de Tags através de arquivos MIB (<i>Case 21919</i>). - A busca de Tags dos arquivos MIB selecionados pelo usuário não mostra mais pastas que estejam vazias (<i>Case 22832</i>). - Corrigida uma situação que impedia a configuração deste Driver em tempo de execução usando o Tag UpdateConfig (<i>Case 20336</i>).
2.0.4	22/12/2015	M. Salvador	- Incluído suporte à nova implementação de UDP Sockets da biblioteca IOKit (<i>Case 18185</i>). - Driver portado para a biblioteca IOKit versão 2 (<i>Case 16423</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.1	28/01/2014	M. Salvador	• Adicionado um limite de 50 dispositivos por Driver.
1.0.1	20/01/2010	M. Salvador M. Bihre	• Versão inicial deste Driver e da documentação.