

# Driver Gerente SNMP (50 Conexões)

Nome do Arquivo	SNMPManager.dll
Fabricante	Simple Network Management Protocol
Equipamentos	
Protocolo	SNMP V1 e V2C
Versão	2.0.28
Última Atualização	28/01/2026
Plataforma	Win32 e Elipse E3 versão 3.2 ou superior
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31301

## Introdução

O Driver Gerente SNMP (50 Conexões) 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 Gerente SNMP (50 Conexões).

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

Device:   
 IP:   
 Port:   
 Max Requests per Packet:   
 Version:

Read Community:   
 Write Community:   
 MIB Parser:

**SNMP v3**  
 UserName:   
 Context Name:   
 Security Level:   
 Auth Passphrase:   
 Auth Style:   
 Priv Passphrase:   
 Priv Encrypt Style:

Extra Modules:   
 Disable

**Aba SNMP Device Config**

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

**Opções disponíveis na aba SNMP Device Config**

OPÇÃO	DESCRIÇÃO
<b>Add</b>	Adiciona um dispositivo ou Agente
<b>Update</b>	Atualiza as configurações para o dispositivo ou Agente selecionado. Alterações realizadas sem a confirmação através desta opção são descartadas
<b>Delete</b>	Remove o dispositivo ou Agente selecionado
<b>Device</b>	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 <b>IP</b> e <b>Port</b> , respectivamente, além de outras configurações
<b>IP</b>	Informa o endereço IP de um Agente
<b>Port</b>	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
<b>Max Requests per Packet</b>	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. <b>NOTA:</b> 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

OPÇÃO	DESCRIÇÃO
	a cada pedido. Consulte o tópico <b>Referência de Tags</b> para mais informações
<b>Version</b>	Versão do protocolo SNMP usada por um Agente. Atualmente este Driver suporta as versões <b>V1, V2C e V3</b>
<b>Read Community</b>	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
<b>Write Community</b>	Informe a comunidade padrão que é usada para escrita. Só é aceito um valor de comunidade para escrita
<b>MIB Parser</b>	Até a versão <b>2.0.21</b> , este Driver utilizava apenas o interpretador <b>smidump</b> para arquivos no formato <b>MIB</b> , gerando a visualização dos Tags que podem ser importados para uma aplicação pelo Tag Browser. A partir da versão <b>2.0.22</b> , também é possível usar o interpretador <b>snmptranslate</b> , que pode gerar resultados diferentes dependendo do arquivo no formato <b>MIB</b>
<b>Browse MIBs</b>	Abre uma caixa de diálogo para selecionar arquivos no formato <b>MIB</b> para importação e que fornecem padrões de Tags
<b>Extra Modules</b>	Lista dos módulos importados usando a opção <b>Browse MIBs</b>
<b>Disable</b>	Desabilita este dispositivo, não estabelecendo comunicação com este Driver ao iniciar, e também esconde um Agente na lista de dispositivos da janela <b>Tag Browser</b>
<b>SNMP v3 - UserName</b>	Nome de usuário a ser utilizado quando a versão selecionada é <b>V3</b> . Deve ser compatível com o nome de usuário definido em um Agente
<b>SNMP v3 - Context Name</b>	Nome de contexto a ser utilizado quando a versão selecionada é <b>V3</b> . Deve ser compatível com o nome de contexto definido em um Agente
<b>SNMP v3 - Security Level</b>	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 <b>NoAuthNoPriv</b> : Não há nenhuma funcionalidade de segurança e o protocolo se comporta

OPÇÃO	DESCRIÇÃO
	de forma similar à versão <b>V2C</b> , <b>AuthNoPriv</b> : Utiliza autenticação sem privacidade ou <b>AuthPriv</b> : Utiliza autenticação e privacidade
<b>SNMP v3 - Auth Passphrase</b>	Informa a senha utilizada para a autenticação. Esta senha deve coincidir com a senha definida em um Agente
<b>SNMP v3 - Auth Style</b>	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 <b>HMAC-MD5</b> , <b>HMAC-SHA1-96</b> ou <b>HMAC-SHA2-256</b>
<b>SNMP v3 - Priv Passphrase</b>	Informa a senha utilizada para a privacidade. Esta senha deve coincidir com a senha definida em um Agente
<b>SNMP v3 - Priv Encrypt Style</b>	Algoritmo para criptografia. As opções disponíveis são <b>DES</b> , <b>3DES</b> , <b>AES128</b> , <b>AES196</b> ou <b>AES256</b>

### 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
<b>MD5</b>	16 bytes
<b>SHA1-96</b>	20 bytes
<b>SHA2-256</b>	32 bytes

#### Tamanhos de chaves de criptografia

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

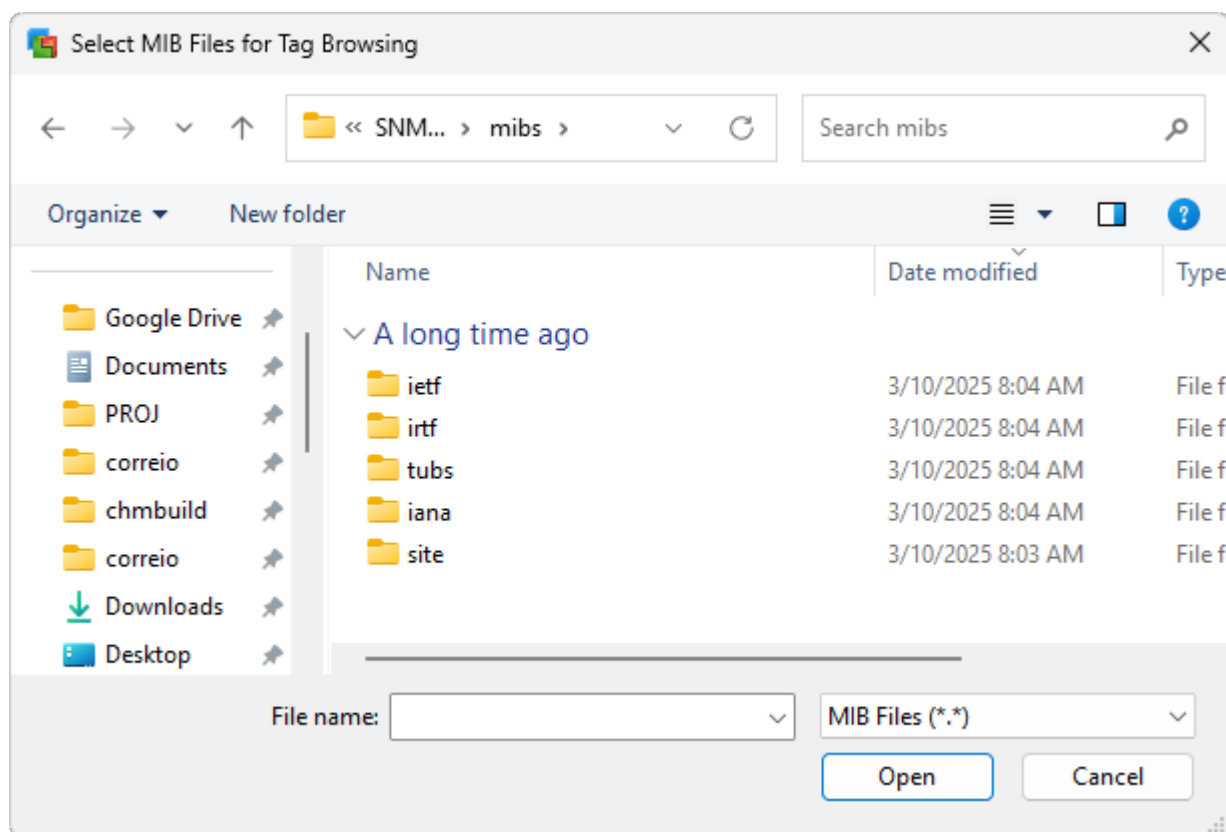
Portanto, ao selecionar a opção de autenticação **MD5** e a opção de criptografia **3DES**, o algoritmo estende a chave automaticamente de 16 para 32 bytes.

## 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.



Janela de seleção de arquivos no formato MIB

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 **Eclipse E3**, **Eclipse Power** ou **Eclipse 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
<b>Name</b>	Nome simbólico padrão de um escalar ou coluna de tabela, conforme obtido de um arquivo no formato <b>MIB</b>
<b>ParamDevice</b>	Nome do Agente definido pelo usuário na lista de equipamentos ou Agentes
<b>ParamItem</b>	Valor numérico padrão do endereço OID ( <i>Object Identifier</i> ) já no formato <b>ASN.1</b> ( <i>Abstract Syntax Notation</i> ), usado pelo protocolo SNMP e obtido de um arquivo no formato <b>MIB</b>

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.

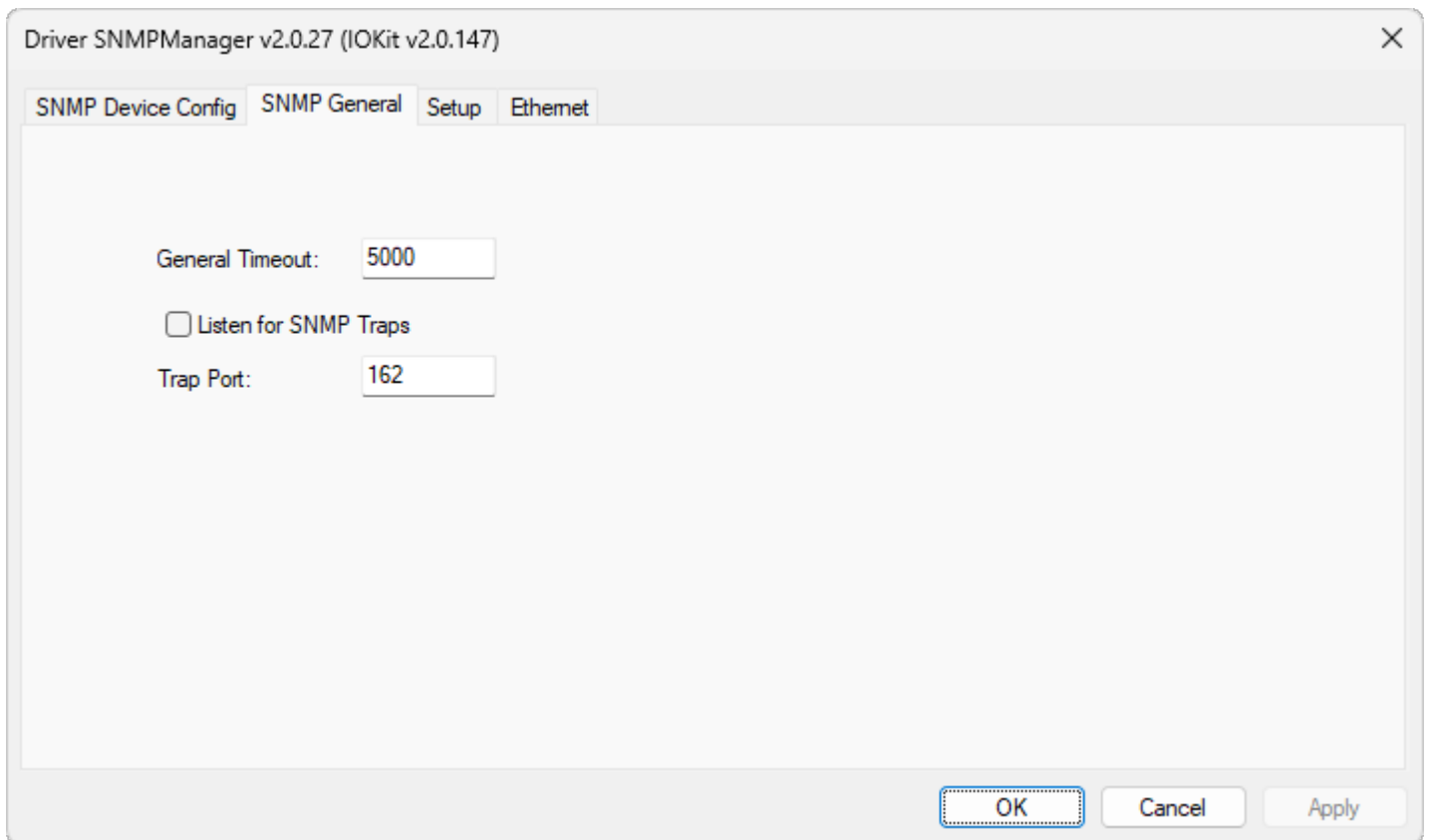
```
'Tag padrão para uma coluna chamada lldpPortNum:
lldpPortNum.1 = 1.0.8802.1.1.2.1.3.7.1.1[1]

'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.



**Aba SNMP General**


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

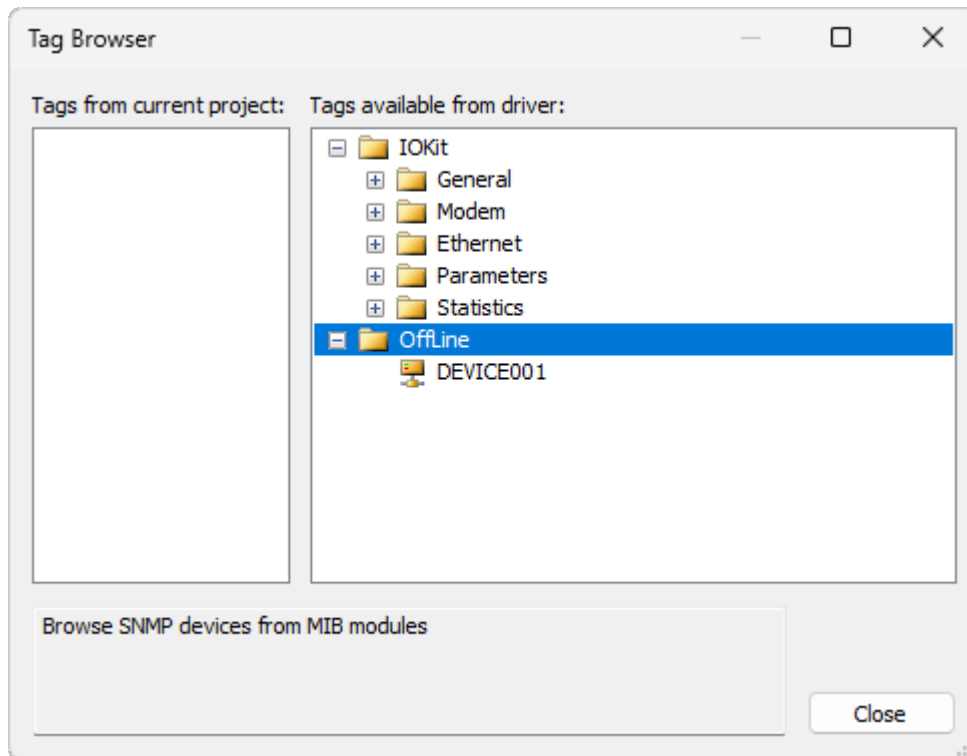
**Opções disponíveis na aba SNMP General**

OPÇÃO	DESCRIÇÃO
<b>General Timeout</b>	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 <b>Setup</b> da biblioteca <b>IOKit</b> . 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 <b>IOKit</b> . Esta opção pode ser modificada em tempo de execução através de um Tag de configuração chamado <b>SNMP.Timeout</b> . Para mais informações, consulte o tópico <b>Documentação das Interfaces de Comunicação</b>
<b>Listen for SNMP Traps</b>	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 <b>Device</b> e <b>Item</b> descrito nesta mensagem. Esta opção pode ser modificada em tempo de execução através de um Tag de configuração chamado <b>SNMP.UseTraps</b> . Para mais informações, consulte o tópico <b>Documentação das Interfaces de Comunicação</b>
<b>Trap Port</b>	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 <b>SNMP.TrapPort</b> . Para

OPÇÃO	DESCRIÇÃO
	mais informações, consulte o tópico <b>Documentação das Interfaces de Comunicação</b>

## Janela Tag Browser

Clique em **Tag Browser**  no **Elipse E3**, **Elipse Power** ou **Elipse 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**.



Janela Tag Browser

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

## 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 com 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 Trap, existem 4 sufixos possíveis para extrair metadados específicos do Trap recebido:

- EnterpriseOID.Source: Retorna o endereço IP do agente que enviou o Trap
  - EnterpriseOID.Generic: Retorna o código Generic-Trap (tipo genérico do trap SNMP v1: 0-6)
  - EnterpriseOID.Specific: Retorna o código Specific-Trap (tipo específico definido pelo fabricante)
  - EnterpriseOID.Timestamp: Retorna o timestamp (sysUpTime) em segundos do momento em que o Trap foi gerado
- **N1:** Se este parâmetro é diferente de 0 (zero), indica que um Tag só é atualizado através de uma mensagem *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 <i>N1</i> seja diferente de 0 (zero), este Tag só é atualizado quando receber 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, retornando o resultado da mesma forma que na opção <b>WalkTable</b> 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 <b>Tabela</b> do protocolo SNMP como Tags Bloco, em que cada Elemento corresponde a uma coluna de uma tabela.

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			O parâmetro <b>Item</b> 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 <b>OnRead</b> deste Tag Bloco. Portanto, se esta tabela tem 5 (cinco) linhas, o evento <b>OnRead</b> deste Tag Bloco é disparado 5 (cinco) vezes
<b>Idem ao anterior</b>	TableOID[Row]	Somente leitura	Da mesma forma que no item anterior, ao adicionar no parâmetro <b>Item</b> de um Tag Bloco associado a uma tabela um índice entre colchetes, a leitura retorna apenas a linha selecionada, caso exista nesta tabela
<b>Idem ao anterior</b>	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 <b>Item</b> , usando o comando SNMP <b>GetNextRequest</b> . O resultado é retornado em um Tag como uma <b>String</b> composta de uma sequência de combinações "OID = valor" separadas por ponto e vírgula, 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'"
<b>Idem ao anterior</b>	EnterpriseOID.Source	Somente leitura	Uso exclusivo com Traps (N1 ≠ 0) Retorna o endereço IP do agente que enviou a mensagem Trap. O EnterpriseOID refere-se ao identificador da organização/fabricante configurado no dispositivo. Este sufixo extrai metadados

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			do Trap recebido, não o valor da variável em si.
<b>Idem ao anterior</b>	EnterpriseOID.Generic	Somente leitura	<p>Uso exclusivo com Traps (N1 ≠ 0)</p> <p>Retorna o código Generic-Trap da mensagem SNMP v1. Valores possíveis:</p> <ul style="list-style-type: none"> <li>0=coldStart,</li> <li>1=warmStart,</li> <li>2=linkDown,</li> <li>3=linkUp,</li> <li>4=authenticationFailure,</li> <li>5=egpNeighborLoss,</li> <li>6=enterpriseSpecific</li> </ul>
<b>Idem ao anterior</b>	EnterpriseOID.Specific	Somente leitura	<p>Uso exclusivo com Traps (N1 ≠ 0)</p> <p>Retorna o código Specific-Trap definido pelo fabricante. Este valor identifica tipos de eventos específicos do equipamento quando Generic-Trap = 6 (enterpriseSpecific).</p>
<b>Idem ao anterior</b>	EnterpriseOID.Timestamp	Somente leitura	<p>Uso exclusivo com Traps (N1 ≠ 0)</p> <p>Retorna o timestamp (sysUpTime) em segundos do momento em que o evento Trap foi gerado no agente. Representa o tempo decorrido desde a última reinicialização do equipamento.</p>
<b>Idem ao anterior</b>	ServerStatus	Somente leitura	<p>Retorna o status interno da conexão com um Agente. Os valores possíveis são <b>0</b>: Sem comunicação ou <b>1</b>: Comunicando normalmente</p>
<b>Nome qualquer, que não pode ser uma String vazia</b>	UpdateConfig	Somente escrita	<p>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 <b>Atualização de Parâmetros em Tempo de Execução</b></p>

# 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 **Elipse E3**, **Elipse Power** ou **Elipse 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
<b>SNMP.DeviceCount</b>	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
<b>SNMP.Device[Index].Name</b>	Nome de um equipamento conforme o índice, como por exemplo SNMP.Device[0].Name = "MainSwitch"
<b>SNMP.Device[Index].IP</b>	Endereço IP
<b>SNMP.Device[Index].Port</b>	Porta UDP/IP
<b>SNMP.Device[Index].Community</b>	Comunidade de leitura
<b>SNMP.Device[Index].WCommunity</b>	Comunidade de escrita
<b>SNMP.Device[Index].Version</b>	Versão do protocolo SNMP
<b>SNMP.Device[Index].MaxReq</b>	Número máximo de pedidos simultâneos
<b>SNMP.Device[Index].Modules</b>	Lista de módulos MIB
<b>SNMP.Device[Index].Disable</b>	Os valores possíveis são <b>1</b> : Desabilita a comunicação com este equipamento ou <b>0</b> : 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**, conforme a figura a seguir.



## 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 <b>Elemento 1</b> : Texto exibido no formato <b>MULTI String</b> , <b>Elemento 2</b> : Tempo restante, em minutos (o valor 65535 indica que não há limite de tempo), <b>Elemento 3</b> : Prioridade, entre 1 (um) e 255, <b>Elemento 4</b> : Status. Os valores possíveis são <b>1</b> : Not used, <b>2</b> : Modifying, <b>3</b> : Validating, <b>4</b> : Valid, <b>5</b> : Error, <b>6</b> : ModifyReq, <b>7</b> : ValidateReq ou <b>8</b> : NotUsedReq, <b>Elemento 5</b> : Nome do usuário que inseriu uma mensagem e <b>Elemento 6</b> : Origem de uma mensagem. Os valores possíveis são <b>1</b> : SourceModeOther, <b>2</b> : SourceModeLocal, <b>3</b> : SourceModeExternal, <b>8</b> :

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			SourceModeCentral, <b>9</b> : SourceModeTimeBasedScheduler, <b>10</b> : SourceModePowerRecovery, <b>11</b> : SourceModeReset, <b>12</b> : SourceModeCommLoss, <b>13</b> : SourceModePowerLoss ou <b>14</b> : 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 <b>2</b> : MemoryTypePermanent, <b>3</b> : MemoryTypeChangeable, <b>4</b> : MemoryTypeVolatile, <b>5</b> : MemoryTypeCurrentBuffer, <b>6</b> : MemoryTypeSchedule ou <b>7</b> : MemoryTypeBlank. Este Tag Bloco contém 4 (quatro) Elementos. Os Elementos são <b>Elemento 1</b> : Texto exibido no formato <b>MULTI String</b> , <b>Elemento 2</b> : Prioridade, entre 1 (um) e 255, <b>Elemento 3</b> : Status. Os valores possíveis são <b>1</b> : Not Used, <b>2</b> : Modifying, <b>3</b> : Validating, <b>4</b> : Valid, <b>5</b> : Error, <b>6</b> : ModifyReq, <b>7</b> : ValidateReq ou <b>8</b> : NotUsedReq e <b>Elemento 4</b> : Nome do usuário que inseriu uma mensagem
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 <b>Elemento 1</b> : Tipo de memória. Os valores possíveis são <b>2</b> : MemoryTypePermanent, <b>3</b> : MemoryTypeChangeable ou <b>4</b> : MemoryTypeVolatile, <b>Elemento 2</b> : Índice de uma mensagem, entre 1 (um) e 255, <b>Elemento 3</b> : Texto exibido no formato <b>MULTI String</b> , <b>Elemento 4</b> : Prioridade, entre 1 (um) e 255, <b>Elemento 5</b> : Nome do usuário que inseriu uma mensagem, <b>Elemento 6</b> :

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
			Indica se os <i>beacons</i> são acionados ou não. Os valores possíveis são 0 (zero, não aciona) ou 1 (um, aciona), <b>Elemento 7</b> : 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 <b>Elemento 1</b> : Tipo de memória. Os valores possíveis são <b>2</b> : MemoryTypePermanent, <b>3</b> : MemoryTypeChangeable ou <b>4</b> : MemoryTypeVolatile, <b>Elemento 2</b> : Índice de uma mensagem, entre 1 (um) e 255, <b>Elemento 3</b> : Prioridade, entre 1 (um) e 255 e <b>Elemento 4</b> : 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 <b>2</b> : MemoryTypePermanent, <b>3</b> : MemoryTypeChangeable ou <b>4</b> : MemoryTypeVolatile. Na escrita este Tag aceita qualquer valor para iniciar a operação de escrita. Na leitura este Tag retorna um valor no formato <b>MMXXXCCCC</b> , em que <b>MM</b> é um tipo de memória com 8 (oito) bits, <b>XXXX</b> é o índice de memória com 16 bits e <b>CCCC</b> é o código CRC ( <i>Cyclic Redundancy Code</i> ou Verificação de Redundância Cíclica) de uma mensagem

PARAMDEVICE	PARAMITEM	OPERAÇÃO	DESCRIÇÃO
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 <b>2</b> : MemoryTypePermanent, <b>3</b> : MemoryTypeChangeable ou <b>4</b> : MemoryTypeVolatile. Na escrita este Tag aceita qualquer valor para iniciar a operação de escrita. Na leitura este Tag retorna um valor no formato <b>MMXXXCCCC</b> , em que <b>MM</b> é um tipo de memória com 8 (oito) bits, <b>XXXX</b> é o índice de memória com 16 bits e <b>CCCC</b> 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
<b>cbx</b>		Cor de fundo de uma mensagem
<b>pbz ou pbr,g,b</b>		Cor de fundo de uma página
<b>cfx ou cfr,g,b</b>		Cor do texto de uma mensagem

TAG DE ABERTURA	TAG DE FECHAMENTO	DESCRIÇÃO
<b>crx,y,w,h,r,g, b</b> ou <b>crx,y,w,h,z</b>		Cor de uma área retangular da página atual de uma mensagem
<b>fx,y</b>		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
<b>fltxoy</b> ou <b>floytx</b>	<b>/fl</b>	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)
<b>fox</b> ou <b>fox,cccc</b>		Permite selecionar o código de uma fonte para visualizar uma mensagem. O parâmetro opcional <b>cccc</b> indica a versão da fonte
<b>gn</b> ou <b>gn,x,y</b> ou <b>gn,x,y,cccc</b>		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 <b>cccc</b> indica o identificador de uma imagem
<b>hcx</b>		Valor hexadecimal de um caractere para ser mostrado
<b>jlx</b>		Especifica o alinhamento do texto. Os valores possíveis são <b>left</b> : Alinhamento à esquerda, <b>center</b> : Texto centralizado, <b>right</b> : Alinhamento à direita ou <b>full</b> : Texto justificado
<b>jpx</b>		Especifica o alinhamento da página. Os valores possíveis são <b>top</b> : Alinhamento pelo topo, <b>middle</b> : Alinhamento pelo centro ou <b>bottom</b> : Alinhamento pela base
<b>msx,y</b>	<b>/msx,y</b>	Especifica um Tag específico de um fabricante
<b>mvtdw,s,r,text</b>		Especifica os parâmetros para um texto que pode ser rolado
<b>nlx</b>		Especifica o início de uma nova linha
<b>np</b>		Especifica o início de uma nova página
<b>ptxoy</b>		Especifica os tempos de uma página
<b>scx /sc</b>		Especifica o espaçamento entre caracteres
<b>trx,y,w,h</b>		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**.

## Configurações de um Driver

A configuração das Interfaces de Comunicação é realizada na caixa de diálogo de configuração de um Driver. Para acessar a configuração da caixa de diálogo no **Eclipse E3** na versão 1.0, siga estes passos:

1. Clique com o botão direito do mouse em um objeto Driver (IODriver).
2. Selecione o item **Propriedades** no menu contextual.
3. Selecione a aba **Driver**.
4. Clique em **Outros parâmetros**.

No **Eclipse E3** versão 2.0 ou posterior, clique em **Configurar o driver**  na barra de ferramentas de um Driver. No **Eclipse SCADA**, siga estes passos:

1. Abra o Organizer.
2. Selecione um Driver na árvore do Organizer.
3. Clique em **Extras** na aba **Driver**.

Atualmente, as Interfaces de Comunicação permitem que apenas uma conexão seja aberta para cada Driver. Isto significa que, no caso de acesso a duas portas seriais, é preciso adicionar dois Drivers em um aplicação e configurar cada um destes Drivers para cada porta serial.

## Caixa de Diálogo de Configuração

A caixa de diálogo das Interfaces de Configuração permite configurar a conexão de I/O que é utilizada por um Driver. Esta caixa de diálogo contém as abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS** descritas nos tópicos a seguir. Se um Driver não implementa uma conexão de I/O específica, a respectiva aba não está disponível para configuração. Alguns Drivers podem conter abas adicionais, específicas para aquele Driver, na caixa de diálogo de configuração.

## Aba Setup

A aba **Setup** contém a configuração geral de um Driver. Esta aba é dividida nos seguintes grupos:

- **Configurações gerais:** Configurações da camada física de um Driver, *time-out* e modo de inicialização
- **Connection management:** Configurações de como a Interface de Comunicação mantém a conexão e qual a política de recuperação em caso de falha
- **Logging options:** Controla a geração dos arquivos de log

Setup

Physical Layer: Ethernet  Start driver OFFLINE

Timeout: 1000 ms    Communication check time: 5000 ms

Connection management

Mode: Automatic (managed by the driver)

Retry failed connection every 20 seconds

Give up after 1 failed retries

Disconnect if non-responsive for 0 seconds

Logging Options

Log to File: C:\eeLogs\MicrolokII\_%DATE%.log

File size limit (MB): 0 ('0' is unlimited)

**Aba Setup**

**Opções gerais da aba Setup**

OPÇÃO	DESCRIÇÃO
<b>Physical Layer</b>	Selecione a interface física em uma lista. As opções disponíveis são <b>Serial</b> , <b>Ethernet</b> , <b>Modem</b> e <b>RAS</b> . A interface selecionada deve ser configurada na aba específica
<b>Timeout</b>	Configure o <i>time-out</i> , em milissegundos, para a camada física. Esta é a medida de tempo que a interface de I/O aguarda para a recepção de um byte qualquer do <i>buffer</i> de recepção
<b>Communication check time</b>	Configure o tempo, em milissegundos, para definir o intervalo em que a comunicação é considerada em estado inativo. Enquanto um Driver de Comunicação receber dados válidos, o estado de comunicação é considerado ativo. Porém, se durante o funcionamento um Driver de Comunicação não receber dados válidos neste período de tempo, o estado é considerado inativo. O estado de comunicação é mostrado no Tag <b>IO.CommunicationStatus</b>
<b>Start driver OFFLINE</b>	Selecione esta opção para que um Driver inicie em modo <b>Offline</b> ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo <b>Online</b> utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface de I/O em tempo de execução

### Opções para o grupo Connection management

OPÇÃO	DESCRIÇÃO
<b>Mode</b>	Selecione o modo de gerenciamento de conexão. Selecionar a opção <b>Automatic</b> permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção <b>Manual</b> permite que uma aplicação gerencie a conexão completamente
<b>Retry failed connection every ... seconds</b>	Selecione esta opção para habilitar a retentativa de conexão de um Driver em um determinado intervalo, em segundos. Se a opção <b>Give up after failed retries</b> não está selecionada, este Driver continua retentando até que a conexão seja efetuada, ou que a aplicação seja parada
<b>Give up after ... failed retries</b>	Habilite esta opção para definir um número máximo de retentativas de conexão. Quando o número especificado de tentativas consecutivas de reconexão é atingido, um Driver vai para o modo <b>Offline</b> , assumindo que um problema de hardware foi detectado. Se um Driver estabelece uma conexão com sucesso, o número de retentativas sem sucesso é zerado. Se esta nova conexão é perdida, então o contador de retentativas inicia do zero
<b>Disconnect if non-responsive for ... seconds</b>	Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> configurado na opção <b>Timeout</b>

**Opções para o grupo Logging Options**

OPÇÃO	DESCRIÇÃO
<b>Log to File</b>	<p>Habilite esta opção e configure o nome do arquivo onde o log é escrito. Arquivos de log podem ser bem extensos, portanto utilize esta opção por curtos períodos de tempo, apenas para o propósito de testes e depurações. Caso se utilize a macro <b>%PROCESS%</b> no nome do arquivo de log, esta é substituída pelo identificador do processo atual. Esta opção é particularmente útil ao se utilizar várias instâncias de um mesmo Driver no <b>Elipse E3</b>, permitindo assim que cada instância gere um arquivo separado de log. Por exemplo, ao configurar esta opção com o valor "c:\e3logs\drivers\sim_%PROCESS%.log", gera-se um arquivo c:\e3logs\drivers\sim_00000FDA.log para o processo <b>0FDAh</b>. Pode-se também utilizar a macro <b>%DATE%</b> no nome do arquivo. Neste caso é gerado um arquivo de log por dia, no formato <b>aaaa_mm_dd</b>. Por exemplo, ao configurar esta opção com o valor "c:\e3logs\drivers\sim_%DATE%.log", gera-se o arquivo c:\e3logs\drivers\sim_2005_12_31.log em 31/12/2005 e o arquivo c:\e3logs\drivers\sim_2006_01_01.log em 01/01/2006. De forma semelhante, a macro <b>%DATE_HOUR%</b> gera um arquivo de log por hora, no formato <b>aaaa_mm_dd_hh</b></p>
<b>File size limit (MB)</b>	<p>Configure o limite de tamanho do arquivo de log, em megabytes. Um valor igual a 0 (zero) significa que não há limite de tamanho para o arquivo de log</p>

## Aba Ethernet

Utilize esta aba para configurar os parâmetros da Interface **Ethernet**. Estes parâmetros, exceto as configurações de porta, devem ser também configurados para uso na Interface **RAS**.

Ethernet

Transport: TCP/IP ▼

PING before connecting  
 Timeout: 4000 ms  
 Retries: 1

Listen for connections on port: 0  
 Share listen port with other processes  
 Interface: (All Interfaces) ▼  
 Use IPv6  Use SSL SSL Settings  
 Enable 'ECHO' suppression  
 IP Filter:

Connect to

<input type="checkbox"/> Main IP:	<span style="border: 1px solid gray; padding: 2px;"> </span>	Port:	<span style="border: 1px solid gray; padding: 2px;">502</span>	<input type="checkbox"/> Local port:	<span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 1:	<span style="border: 1px solid gray; padding: 2px;"> </span>	Port:	<span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port:	<span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 2:	<span style="border: 1px solid gray; padding: 2px;"> </span>	Port:	<span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port:	<span style="border: 1px solid gray; padding: 2px;">0</span>
<input type="checkbox"/> Backup IP 3:	<span style="border: 1px solid gray; padding: 2px;"> </span>	Port:	<span style="border: 1px solid gray; padding: 2px;">0</span>	<input type="checkbox"/> Local port:	<span style="border: 1px solid gray; padding: 2px;">0</span>

**Aba Ethernet**

**Opções disponíveis na aba Ethernet**

OPÇÃO	DESCRIÇÃO
<b>Transport</b>	Selecione o valor <b>TCP/IP</b> para um <i>socket</i> TCP ( <i>stream</i> ) ou selecione o valor <b>UDP/IP</b> para utilizar um <i>socket</i> UDP ( <i>connectionless datagram</i> )
<b>Listen for connections on port</b>	Utilize esta opção para aguardar por novas conexões em uma porta IP específica, comum em Drivers Escravos. Caso esta opção permaneça desmarcada, um Driver se conecta ao endereço e porta especificados no grupo <b>Connect to</b>
<b>Share listen port with other processes</b>	Selecione esta opção para compartilhar a porta de escuta com outros Drivers e processos
<b>Interface</b>	Selecione a interface de rede local, identificada pelo endereço IP, que um Driver utiliza para efetuar e receber conexões, ou selecione o valor <b>(All Interfaces)</b> para permitir conexões em qualquer interface de rede
<b>Use IPv6</b>	Selecione esta opção para forçar um Driver a utilizar endereços no formato <b>IPv6</b> em todas as conexões Ethernet. Deixe esta opção desmarcada para utilizar o formato <b>IPv4</b>
<b>Enable 'ECHO' suppression</b>	Habilite esta opção para eliminar o <i>eco</i> dos dados recebidos. O <i>eco</i> é uma cópia dos dados enviados, que pode ser retornada antes da mensagem de resposta
<b>IP Filter</b>	Lista de endereços IP restringidos ou permitidos de onde um Driver aceita conexões ( <i>Firewall</i> ). Consulte a propriedade <b>IO.Ethernet.IPFilter</b> para mais informações
<b>PING before connecting</b>	Habilite esta opção para executar um comando <b>ping</b> , ou seja, para verificar se um dispositivo pode ser encontrado na rede, em um dispositivo antes de tentar uma conexão

OPÇÃO	DESCRIÇÃO
	<p>com o <i>socket</i>. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um <i>socket</i> com um dispositivo. O <i>time-out</i> de uma conexão com um <i>socket</i> pode ser bem alto. As opções disponíveis são:</p> <ul style="list-style-type: none"> <li>• <b>Timeout:</b> Especifique o número de milissegundos de espera por uma resposta de um comando <b>ping</b>. Deve-se usar um comando <b>ping</b> para verificar o tempo normal de resposta, configurando esta opção para um valor acima desta média. Normalmente pode-se configurar um valor entre 1000 e 4000 milissegundos, ou seja entre 1 (um) e 4 (quatro) segundos</li> <li>• <b>Retries:</b> Número de retentativas de um comando <b>ping</b>, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada</li> </ul>

#### Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
<b>Main IP</b>	<p>Digite o endereço IP de um dispositivo remoto. Pode-se usar tanto o endereço IP separado por pontos quanto uma URL. No caso de uma URL, um Driver usa o serviço de DNS disponível para mapear a URL para um endereço IP, como por exemplo "192.168.0.13" ou "Server1"</p>
<b>Port</b>	<p>Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535</p>
<b>Local port</b>	<p>Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto</p>
<b>Backup IP 1, 2 e 3</b>	<p>Indique o endereço IP, a porta IP e a porta IP local fixa de até 3 (três) endereços de <i>backup</i> de um dispositivo remoto</p>

## Configurações Gerais

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** gerais das Interfaces de Comunicação.

### Tags de Comunicação

#### Tags Gerais das Interfaces de Comunicação (N2/B2 = 0)

Os Tags descritos a seguir são fornecidos para todas as Interfaces de I/O suportadas.

## IO.CommunicationStatus

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	6 (seis)
<b>Configuração por String</b>	IO.CommunicationStatus

Este Tag informa o estado da comunicação de um Driver. Indica o funcionamento da comunicação em função do recebimento de dados válidos dentro de um período de tempo arbitrado na configuração. Para mais informações, consulte o tópico **Aba Setup**. Os valores possíveis são **0 - Comunicação inativa**: O Driver não recebeu dados válidos ou deixou de receber dados depois de  $n$  milissegundos, conforme configurado na janela de propriedades, ou **1 - Comunicação ativa**: O Driver está recebendo dados válidos.

## IO.IOKitEvent

<b>Tipo de Tag</b>	Tag Bloco
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro B1</b>	-1 (menos um)
<b>Parâmetro B2</b>	0 (zero)
<b>Parâmetro B3</b>	0 (zero)
<b>Parâmetro B4</b>	1 (um)
<b>Propriedade Size</b>	4 (quatro)
<b>Propriedade ParamItem</b>	IO.IOKitEvent

Este Bloco retorna eventos de Driver gerados por várias fontes nas Interfaces de Comunicação. A propriedade **TimeStamp** de um Bloco representa o momento em que um evento ocorre. Os Elementos de Bloco são os seguintes:

- **Elemento 0**: Tipo de evento. Os valores possíveis são **0**: Informação, **1**: Advertência ou **2**: Erro
- **Elemento 1**: Fonte de um evento. Os valores possíveis são **0**: Driver (específico de um Driver), **-1**: IOKit (eventos genéricos da Interface de Comunicação), **-2**: Interface **Serial**, **-3**: Interface **Modem**, **-4**: Interface **Ethernet** ou **-5**: Interface **RAS**
- **Elemento 2**: Número do erro, específico de cada fonte de evento
- **Elemento 3**: Mensagem de um evento, uma **String** específica de cada evento

### NOTA

Um Driver mantém um número máximo de 100 eventos internamente. Se eventos adicionais são reportados, os eventos mais antigos são descartados.

## IO.PhysicalLayerStatus

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	IO.PhysicalLayerStatus

Este Tag indica o estado da camada física. Os valores possíveis são os seguintes:

- **0:** Camada física parada, ou seja, um Driver está em modo **Offline**, a camada física falhou ao inicializar ou excedeu o número máximo de tentativas de reconexão
- **1:** Camada física iniciada mas não conectada, ou seja, um Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** está configurada com o valor **Automatic**, a camada física pode estar conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** está configurada com o valor **Manual**, então a camada física permanece neste estado até ser forçada a conectar
- **2:** Camada física conectada, ou seja, a camada física está pronta para ser usada. Isto **NÃO** significa que um equipamento esteja conectado, apenas que a camada de acesso está funcionando

## IO.SetConfigurationParameters

<b>Tipo de Tag</b>	Tag Bloco
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro B1</b>	-1 (menos um)
<b>Parâmetro B2</b>	0 (zero)
<b>Parâmetro B3</b>	0 (zero)
<b>Parâmetro B4</b>	3 (três)
<b>Propriedade Size</b>	2 (dois)
<b>Propriedade ParamItem</b>	IO.SetConfigurationParameters

Use este Tag para modificar qualquer propriedade da caixa de diálogo de configuração de um Driver em tempo de execução.

Este Tag funciona somente enquanto um Driver está em modo **Offline**. Para iniciar um Driver em modo **Offline**, selecione a opção **Start driver OFFLINE** na caixa de diálogo de configuração deste Driver. Pode-se tanto escrever em um Tag PLC ou em um Tag Bloco contendo os parâmetros a serem modificados. As escritas de Elementos de Bloco individuais não são suportadas, um Bloco inteiro precisa ser escrito de uma vez só.

No **Elipse SCADA** é necessário usar um Tag Bloco. Cada parâmetro a ser configurado utiliza dois Elementos de Bloco. Por exemplo, caso seja necessário configurar 3 (três) parâmetros, então o tamanho do Bloco deve ser 6 (seis,  $3 \times 2$ ). O primeiro Elemento é o nome da propriedade, como uma **String**, e o segundo Elemento é o valor desta propriedade, conforme o exemplo a seguir.

```
// 'Block' deve ser um Tag Bloco com leitura automática,
// leitura por varredura e escrita automática desabilitadas.
// Configura os parâmetros
Block.element001 = "IO.Type" // Parâmetro 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parâmetro 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parâmetro 3
Block.element006 = 19200
// Escreve o Bloco inteiro
Block.Write()
```

Ao usar o **Elipse E3**, a habilidade de criar *arrays* em tempo de execução permite o uso tanto de um Tag de Comunicação quanto de um Tag Bloco. Pode-se utilizar o método **Write** de um Driver para enviar os parâmetros diretamente para este Driver, sem a necessidade de criar um Tag, conforme o exemplo a seguir.

```
Dim arr(6)
' Configura os elementos do array
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' Há dois métodos de enviar os parâmetros
' Método 1: Usando um Tag de Comunicação
tag.WriteEx arr
' Método 2: Sem utilizar um Tag
Driver.Write -1, 0, 0, 3, arr
```

Uma variação do exemplo anterior usa um *array* bidimensional.

```
Dim arr(10)
' Configura os elementos do array. Note que o array foi redimensionado
' para 10 elementos. Elementos vazios são ignorados pelo Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

Um Driver não valida nomes de parâmetros ou valores passados, por isto tenha cuidado ao escrever parâmetros e valores. O método **Write** falha se o *array* de configuração é criado incorretamente. Pode-se consultar o log de um Driver ou usar o parâmetro *writeStatus* do método **WriteEx** para descobrir a causa exata de um erro.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , , strError Then
    MsgBox "Falha ao configurar os parâmetros do Driver: " + strError
End If
```

## IO.WorkOnline

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	4 (quatro)
<b>Configuração por String</b>	IO.WorkOnline

Este Tag informa o estado atual de um Driver e permite iniciar ou parar a camada física. Os valores possíveis são os seguintes:

- **0 - Driver Offline:** A camada física está fechada ou parada. Este modo permite uma configuração dinâmica dos parâmetros de um Driver através do Tag **IO.SetConfigurationParameters**
- **1 - Driver Online:** A camada física está aberta ou em execução. Enquanto está em modo **Online**, a camada física pode ser conectada ou desconectada e o estado atual pode ser conferido no Tag **IO.PhysicalLayerStatus**

No exemplo a seguir, utilizando o **Elipse E3**, um Driver é colocado em modo **Offline**, a porta COM é modificada e então é colocado em modo **Online** novamente.

```
'Configura o Driver em modo Offline
Driver.Write -1, 0, 0, 4, 0
'Muda a porta para COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configura o Driver em modo Online
Driver.Write -1, 0, 0, 4, 1
```

O método **Write** pode falhar ao configurar um Driver em modo **Online**, ou seja, escrevendo o valor 1 (um). Neste caso, este Driver permanece em modo **Offline**. A causa da falha pode ser:

- Tipo de camada física configurada incorretamente, provavelmente um valor inválido foi configurado para a propriedade **IO.Type**
- Este Driver pode ter ficado sem memória
- A camada física pode ter deixado de criar a *thread* de trabalho. Procure no arquivo de log pela mensagem "Failed to create physical layer thread!"
- A camada física não conseguiu inicializar. A causa da falha depende do tipo de camada física. Pode ser um número de porta serial inválida, falha ao inicializar o Windows Sockets ou falha ao inicializar o TAPI (modem), entre outras. A causa é gravada no arquivo de log

## IMPORTANTE

Mesmo que a configuração de um Driver para o modo **Online** seja bem-sucedida, isto não significa necessariamente que a camada física esteja pronta para uso, ou seja, pronta para executar operações de entrada e saída com um equipamento externo. O Tag **IO.PhysicalLayerStatus** deve ser verificado para assegurar que a camada física esteja conectada e preparada para a comunicação.

## Propriedades

Estas são as propriedades gerais de todas as Interfaces de I/O suportadas.

### IO.ConnectionMode

9 Controla o modo de gerenciamento da Conexão. Os valores possíveis são **0**: Modo automático, em que um Driver gerencia a conexão ou **1**: Modo manual, em que uma aplicação gerencia a conexão.

### IO.GiveUpEnable

■ Quando configurada para Verdadeiro, define um número máximo de tentativas de reconexão. Se todas as reconexões falharem, um Driver entra em modo **Offline**. Se configurada para Falso, um Driver tenta até que uma reconexão seja bem-sucedida.

## IO.GiveUpTries

9 Número de tentativas de reconexão antes que esta seja abortada. Por exemplo, se o valor desta propriedade é igual a 1 (um), um Driver tenta apenas uma reconexão quando a conexão é perdida. Se esta falhar, este Driver entra em modo **Offline**.

## IO.InactivityEnable

☑ Configure em Verdadeiro para habilitar e em Falso para desabilitar a detecção de inatividade. A camada física é desconectada se está inativa por um certo período de tempo. A camada física é considerada inativa apenas se é capaz de enviar dados mas não de recebê-los de volta.

## IO.InactivityPeriodSec

9 Número de segundos para a verificação de inatividade. Se a camada física está inativa por este período de tempo, então é desconectada.

## IO.RecoverEnable

☑ Configure em Verdadeiro para habilitar um Driver a recuperar conexões perdidas e em Falso para deixar um Driver em modo **Offline** quando uma conexão é perdida.

## IO.RecoverPeriodSec

9 Tempo de espera entre duas tentativas de conexão, em segundos.

### NOTA

A primeira reconexão é executada imediatamente após a conexão ser perdida.

## IO.StartOffline

☑ Configure em Verdadeiro para iniciar um Driver em modo **Offline** e em Falso para iniciar um Driver em modo **Online**.

### NOTA

Não faz sentido modificar esta propriedade em tempo de execução, já que esta só pode ser modificada quando um Driver já está em modo **Offline**. Para configurar um Driver em modo **Online** em tempo de execução, escreva o valor 1 (um) no Tag **IO.WorkOnline**.

## IO.TimeoutMs

9 Define o *time-out* da camada física, em milissegundos. Um segundo equivale a 1000 milissegundos.

## IO.Type

A Define o tipo de interface física utilizada por um Driver. Os valores possíveis são os seguintes:

- **N ou None:** Não utiliza uma interface física, ou seja, um Driver deve fornecer uma interface personalizada
- **S ou Serial:** Utiliza uma porta serial local (COM $n$ )
- **M ou Modem:** Utiliza um modem local, interno ou externo, acessado via TAPI (*Telephony Application Programming Interface*)

- **E ou Ethernet:** Utiliza um *socket* TCP/IP ou UDP/IP
- **R ou RAS:** Utiliza uma Interface **RAS** (*Remote Access Server*). Um Driver conecta-se a um equipamento RAS através da Interface **Ethernet** e então emite um comando **AT** (*dial*)

## Configuração de Estatísticas

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** das estatísticas das Interfaces de Comunicação.

### Tags de Comunicação

#### Tags de Estatísticas das Interfaces de Comunicação (N2/B2 = 0)

Os Tags descritos a seguir mostram estatísticas para todas as Interfaces de Comunicação.

#### IO.Stats.Partial.BytesRecv

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1101
<b>Configuração por String</b>	IO.Stats.Partial.BytesRecv

Este Tag retorna a quantidade de bytes recebidos na conexão atual.

#### IO.Stats.Partial.BytesSent

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1100
<b>Configuração por String</b>	IO.Stats.Partial.BytesSent

Este Tag retorna a quantidade de bytes enviados na conexão atual.

## IO.Stats.Partial.TimeConnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1102
<b>Configuração por String</b>	IO.Stats.Partial.TimeConnectedSeconds

Este Tag retorna o número de segundos que um Driver está conectado na conexão atual ou 0 (zero) se um Driver está desconectado.

## IO.Stats.Partial.TimeDisconnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1103
<b>Configuração por String</b>	IO.Stats.Partial.TimeDisconnectedSeconds

Este Tag retorna o número de segundos que um Driver está desconectado desde o término da última conexão ou 0 (zero) se um Driver está conectado.

## IO.Stats.Total.BytesRecv

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1001
<b>Configuração por String</b>	IO.Stats.Total.BytesRecv

Este Tag retorna a quantidade de bytes recebidos desde que um Driver foi carregado.

## IO.Stats.Total.BytesSent

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1000
<b>Configuração por String</b>	IO.Stats.Total.BytesSent

Este Tag retorna a quantidade de bytes enviados desde que um Driver foi carregado.

## IO.Stats.Total.ConnectionCount

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1004
<b>Configuração por String</b>	IO.Stats.Total.ConnectionCount

Este Tag retorna a quantidade de conexões que um Driver já estabeleceu, com sucesso, desde que foi carregado.

## IO.Stats.Total.TimeConnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1002
<b>Configuração por String</b>	IO.Stats.Total.TimeConnectedSeconds

Este Tag retorna o número de segundos que um Driver permaneceu conectado desde que foi carregado.

## IO.Stats.Total.TimeDisconnectedSeconds

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	0 (zero)
<b>Parâmetro N4</b>	1003
<b>Configuração por String</b>	IO.Stats.Total.TimeDisconnectedSeconds

Este Tag retorna o número de segundos que um Driver permaneceu desconectado desde que foi carregado.

## Propriedades

Atualmente, não existem propriedades definidas especificamente para mostrar as estatísticas das Interfaces de Comunicação em tempo de execução.

## Configuração da Interface Ethernet

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e das **Propriedades** da Interface **Ethernet**.

## Tags de Comunicação

### Tags da Interface Ethernet (N2/B2 = 4)

Os Tags descritos a seguir permitem controlar e identificar a Interface **Ethernet** em tempo de execução e também são válidos quando a Interface **RAS** está selecionada.

#### IMPORTANTE

Estes Tags estão disponíveis **SOMENTE** enquanto um Driver está em modo **Online**.

## IO.Ethernet.IPSelect

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Leitura ou Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	0 (zero)
<b>Configuração por String</b>	IO.Ethernet.IPSelect

Indica o endereço IP ativo. Os valores possíveis são **0**: O endereço principal de IP está selecionado, **1**: O primeiro endereço IP alternativo ou de *backup* está selecionado, **2**: O segundo endereço IP alternativo ou de *backup* está selecionado ou **3**: O terceiro endereço IP alternativo ou de *backup* está selecionado.

Se a Interface **Ethernet** ou **RAS** está conectada, este Tag indica qual dos quatro endereços IP configurados está em uso. Se a Interface está desconectada, este Tag indica qual endereço IP é usado primeiro na próxima tentativa de conexão.

Durante o processo de conexão, se o endereço IP ativo não está disponível, a Interface de Comunicação tenta conectar-se usando o outro endereço IP. Se a conexão com o endereço IP alternativo funcionar, este é configurado como o endereço IP ativo (*switchover* automático).

Para forçar um *switchover* manual, escreva valores de 0 (zero) a três (3) neste Tag. Isto força a reconexão com o endereço IP especificado (**0**: Endereço principal, **1, 2, 3**: Endereços alternativos) se um Driver está atualmente conectado. Se um Driver está desconectado, este Tag configura o endereço IP ativo para a próxima tentativa de conexão.

## IO.Ethernet.IPSwitch

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Escrita
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	1 (um)
<b>Configuração por String</b>	IO.Ethernet.IPSwitch

Qualquer valor escrito neste Tag força um *switchover* manual. Se o endereço principal de IP está ativo, então o primeiro endereço IP alternativo ou de *backup* é ativado, e assim por diante passando por todos os endereços IP alternativos e voltando para o endereço principal até estabelecer uma conexão.

Se um Driver está desconectado, este Tag configura o endereço IP ativo para a próxima tentativa de conexão.

## IO.Ethernet.SocketState

<b>Tipo de Tag</b>	Tag de Comunicação
<b>Tipo de Acesso</b>	Somente Leitura
<b>Parâmetro N1</b>	-1 (menos um)
<b>Parâmetro N2</b>	0 (zero)
<b>Parâmetro N3</b>	4 (quatro)
<b>Parâmetro N4</b>	2 (dois)
<b>Configuração por String</b>	IO.Ethernet.SocketState

A propriedade **Value** deste Tag corresponde a estados do *socket* em um mapa de bits:

- **Bit 0**: 0 (zero, não está em escuta) ou 1 (um, em escuta)

- **Bit 1:** 0 (zero, desconectado) ou 1 (um, conectado)

## Propriedades

Estas propriedades controlam a configuração da Interface **Ethernet**.

### NOTA

A Interface **Ethernet** também é usada pela Interface **RAS**.

## IO.Ethernet.AcceptConnection

☑ Configure em Falso se um Driver não deve aceitar conexões externas, ou seja, se um Driver se comporta como mestre, ou configure em Verdadeiro para habilitar a recepção de conexões, ou seja, se um Driver se comporta como escravo.

## IO.Ethernet.BackupEnable[2,3]

☑ Configure em Verdadeiro para habilitar o endereço IP alternativo ou de *backup*. Se a tentativa de reconectar com o endereço IP principal falhar, um Driver tenta utilizar um endereço IP alternativo ou de *backup*. Configure em Falso para desabilitar a utilização.

## IO.Ethernet.BackupIP[2,3]

📌 Endereço IP alternativo ou de *backup* de um equipamento remoto. Pode-se utilizar tanto o endereço numérico como o nome de *host* de um equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

## IO.Ethernet.BackupLocalPort[2,3]

📌 Número da porta local a ser utilizada na conexão ao endereço IP alternativo ou de *backup* de um equipamento remoto. Usado apenas se a propriedade **IO.Ethernet.BackupLocalPortEnable** está configurada para Verdadeiro.

## IO.Ethernet.BackupLocalPortEnable[2,3]

☑ Configure em Verdadeiro para forçar o uso de uma porta local específica ao conectar ao endereço IP alternativo ou de *backup* ou configure em Falso para utilizar qualquer porta local disponível.

## IO.Ethernet.BackupPort[2,3]

📌 Número da porta do endereço IP alternativo ou de *backup* de um equipamento remoto, usado juntamente com a propriedade **IO.Ethernet.BackupIP**.

## IO.Ethernet.IPFilter

📌 Lista de endereços IPv4 ou IPv6 separados por vírgula, que define de quais endereços um Driver aceita ou bloqueia conexões. Pode-se utilizar asteriscos, como por exemplo "192.168.\*.\*", ou intervalos, como por exemplo "192.168.0.41-50", em qualquer parte dos endereços IP. Para bloquear um endereço IP ou um intervalos de endereços IP, use o caractere til ("~") no início do endereço, conforme os exemplos a seguir:

- **192.168.0.24:** Aceita apenas conexões do endereço IPv4 192.168.0.24
- **192.168.0.41-50:** Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.41 e 192.168.0.50
- **192.168.0.\*:** Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255

- **fe80:3bf:877::\*:\*** (expande para **fe80:03bf:0877:0000:0000:0000:0000:0000**): Aceita conexões de endereços IPv6 no intervalo entre `fe80:03bf:0877:0000:0000:0000:0000:0000` e `fe80:03bf:0877:0000:0000:0000:0000:ffff:ffff`
- **192.168.0.10**, **192.168.0.15**, **192.168.0.20**: Aceita conexões dos endereços IPv4 192.168.0.10, 192.168.0.15 e 192.168.0.20
- **~192.168.0.95**, **192.168.0.\***: Aceita conexões dos endereços IPv4 no intervalo entre 192.168.0.0 e 192.168.0.255, exceto o endereço IPv4 192.168.0.95

Quando um Driver recebe uma tentativa de conexão, a lista de filtros é percorrida sequencialmente da esquerda para a direita, procurando por uma autorização ou bloqueio específico para o endereço IP de onde veio a conexão. Se nenhum elemento da lista corresponde ao endereço IP, a autorização ou bloqueio são ditados pelo último elemento da lista:

- Se o último elemento da lista é uma autorização, como por exemplo "192.168.0.24", então todos os endereços IP não encontrados na lista são bloqueados
- Se o último elemento da lista é um bloqueio, como por exemplo "~192.168.0.24", então todos os endereços IP não encontrados na lista são autorizados

Se um endereço IP aparece em mais de um filtro da lista, o filtro mais à esquerda tem precedência. Por exemplo, no caso de "~192.168.0.95, 192.168.0.\*", o endereço IP 192.168.0.95 se encaixa nas duas regras, mas a regra que vale é a mais à esquerda, "~192.168.0.95", e portanto o endereço IP é bloqueado.

Quando o **IOKit** bloqueia uma conexão, a mensagem "Blocked incoming socket connection from {IP}!" é logada.

No caso de conexões UDP em modo escuta em *broadcast*, em que um Driver pode receber pacotes de diferentes endereços IP, o bloqueio ou permissão é realizado a cada pacote recebido. Se um pacote é recebido de um endereço IP bloqueado, a mensagem "Blocked incoming packet from {IP} (discarding {N} bytes)!" é logada.

## IO.Ethernet.ListenIP

**A** Endereço IP da interface local de rede por onde um Driver efetua e aceita conexões. Deixe esta propriedade vazia para efetuar e aceitar conexões por qualquer interface local de rede.

## IO.Ethernet.ListenPort

**9** Número da porta IP utilizada por um Driver para escutar conexões.

## IO.Ethernet.MainIP

**A** Endereço IP de um equipamento remoto. Pode-se utilizar tanto o endereço numérico como o nome de *host* de um equipamento, como por exemplo "192.168.0.7" ou "SERVER2".

## IO.Ethernet.MainLocalPort

**9** Número da porta local a ser utilizada na conexão ao endereço IP principal de um equipamento remoto. Este valor é usado apenas se a propriedade **IO.Ethernet.MainLocalPortEnable** é igual a Verdadeiro.

## IO.Ethernet.MainLocalPortEnable

**☑** Configure em Verdadeiro para forçar o uso de uma porta local específica ao conectar ao endereço IP principal ou configure em Falso para utilizar qualquer porta local disponível.

## IO.Ethernet.MainPort

9 Número da porta IP em um equipamento remoto, usado em conjunto com a propriedade **IO.Ethernet.MainIP**.

## IO.Ethernet.PingEnable

▣ Configure em Verdadeiro para habilitar o envio de um comando **ping** para o endereço IP de um equipamento remoto, antes de tentar conectar-se ao *socket*. O *time-out* de conexão do *socket* não pode ser controlado, por isto o envio de um comando **ping** antes de conectar-se é uma maneira rápida de detectar se a conexão vai falhar. Configure em Falso para desabilitar o comando **ping**.

## IO.Ethernet.PingTimeoutMs

9 Tempo de espera por uma resposta de um comando **ping**, em milissegundos.

## IO.Ethernet.PingTries

9 Número máximo de tentativas de comandos **ping**. O valor mínimo é 1 (um), incluindo o primeiro comando **ping**.

## IO.Ethernet.ShareListenPort

▣ Configure em Verdadeiro para compartilhar a porta de escuta com outros Drivers e processos ou Falso para abrir a porta de escuta em modo exclusivo. Para compartilhar uma porta de escuta com sucesso, todos os Drivers e processos envolvidos devem abrir esta porta em modo compartilhado. Quando uma porta de escuta é compartilhada, cada nova conexão é distribuída para um dos processos que estão escutando. Desta forma, se um Driver Escravo só suporta uma conexão por vez, pode-se utilizar várias instâncias deste Driver escutando na mesma porta, portanto simulando um Driver com suporte a múltiplas conexões.

## IO.Ethernet.SupressEcho

▣ Configure em Verdadeiro para eliminar o eco presente em uma comunicação. O eco é a recepção indesejada de uma cópia exata de todos os pacotes de dados que um Driver enviou para um equipamento.

## IO.Ethernet.Transport

A Define o protocolo de transporte. Os valores possíveis são **T ou TCP**: Utiliza o protocolo TCP/IP ou **U ou UDP**: Utiliza o protocolo UDP/IP.

## IO.Ethernet.UseIPv6

▣ Configure em Verdadeiro para utilizar endereços IPv6 em todas as conexões Ethernet ou configure em Falso para utilizar endereços IPv4 (padrão).

## Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.28	28/01/2026	A. Fetzner M. Ludwig	<ul style="list-style-type: none"> <li>Implementado suporte à leitura de metadados de Trap SNMP v1 (Case 39340).</li> <li>Corrigido um problema de <i>buffer overflow</i> ao decodificar um <i>Object ID</i> para <b>String</b> (Case 39319).</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			<ul style="list-style-type: none"> <li>• Driver atualizado para a biblioteca <b>IOKit</b> versão <b>3.0</b> e Visual Studio 2022 (<i>Case 38020</i>).</li> </ul>
<b>2.0.27</b>	10/03/2025	M. Salvador	<ul style="list-style-type: none"> <li>• Corrigidas as escritas ou comandos (<i>Case 37437</i>).</li> </ul>
<b>2.0.26</b>	21/03/2024	M. Salvador	<ul style="list-style-type: none"> <li>• Corrigida a leitura de tabelas (<i>Case 35438</i>).</li> </ul>
<b>2.0.24</b>	26/06/2023	M. Salvador	<ul style="list-style-type: none"> <li>• Implementada uma segunda opção de MIB Parser por equipamento. Agora é possível escolher entre <b>smidump</b> e <b>snmptranslate</b> (<i>Case 33723</i>).</li> </ul>
<b>2.0.23</b>	26/06/2023	M. Salvador	<ul style="list-style-type: none"> <li>• Implementado o Perfil NTCIP DMS (<i>Dynamic Message Sign</i>), utilizado em rodovias (<i>Case 33727</i>).</li> </ul>
<b>2.0.19</b>	11/11/2021	M. Salvador	<ul style="list-style-type: none"> <li>• Adicionado suporte ao protocolo SNMP versão <b>3</b> (<i>Case 18316</i>).</li> </ul>
<b>2.0.16</b>	05/08/2019	M. Ludwig	<ul style="list-style-type: none"> <li>• Driver portado para o Visual Studio 2017 (<i>Case 27093</i>).</li> </ul>
<b>2.0.15</b>	03/04/2019	M. Ludwig F. Englert M. Salvador	<ul style="list-style-type: none"> <li>• Melhorado o diagnóstico de erros de carga de arquivos MIB (<i>Case 23007</i>).</li> <li>• Implementadas diversas melhorias na importação de Tags através de arquivos MIB (<i>Case 21919</i>).</li> <li>• A busca de Tags dos arquivos MIB selecionados pelo usuário não mostra mais pastas que estejam vazias (<i>Case 22832</i>).</li> <li>• Corrigida uma situação que impedia a configuração deste Driver em tempo de execução usando o Tag <b>UpdateConfig</b> (<i>Case 20336</i>).</li> </ul>
<b>2.0.4</b>	22/12/2015	M. Salvador	<ul style="list-style-type: none"> <li>• Incluído suporte à nova implementação de UDP Sockets da biblioteca <b>IOKit</b> (<i>Case 18185</i>).</li> <li>• Driver portado para a biblioteca <b>IOKit</b> versão <b>2.0</b> (<i>Case 16423</i>).</li> </ul>
<b>1.1.1</b>	28/01/2014	M. Salvador	<ul style="list-style-type: none"> <li>• Adicionado um limite de 50 dispositivos por Driver.</li> </ul>

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.0.1	20/01/2010	M. Salvador M. Bihre	<ul style="list-style-type: none"><li>• Versão inicial deste Driver e da documentação.</li></ul>

**Matriz**

Rua Mostardeiro, 322/Cj. 902, 1001 e  
1002

90430-000 — Porto Alegre — RS

Fone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: [elipse-rs@elipse.com.br](mailto:elipse-rs@elipse.com.br)

**Filial no Paraná**

Av. Sete de Setembro, 4698/1708

80240-000 — Curitiba — PR

Fone: (+55 41) 4062-5824

E-mail: [elipse-pr@elipse.com.br](mailto:elipse-pr@elipse.com.br)

**Filial no Rio de Janeiro**

Av. José Silva de A. Neto, 200/Bl. 4/Sl.  
109B

22250-044 — Rio de Janeiro — RJ

Fone: (+55 21) 2430-5912

Suporte Técnico: (+55 21) 2430-5963

E-mail: [elipse-rj@elipse.com.br](mailto:elipse-rj@elipse.com.br)

**Filial em São Paulo**

Rua dos Pinheiros, 870/Cj. 141 e 142

05422-001 — São Paulo — SP

Fone: (+55 11) 3061-2828

Fax: (+55 11) 3086-2338

E-mail: [elipse-sp@elipse.com.br](mailto:elipse-sp@elipse.com.br)

**Filial em Minas Gerais**

Rua Antônio de Albuquerque, 156/705

30112-010 — Belo Horizonte — MG

Fone: (+55 31) 4062-5824

E-mail: [elipse-mg@elipse.com.br](mailto:elipse-mg@elipse.com.br)

**Filial em Taiwan**

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Fone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: [evan@elipse.com.br](mailto:evan@elipse.com.br)

Consulte nosso website para informações sobre o representante do seu estado.

[www.elipse.com.br](http://www.elipse.com.br)

[kb.elipse.com.br](http://kb.elipse.com.br)

[forum.elipse.com.br](http://forum.elipse.com.br)

[www.youtube.com/elipsesoftware](http://www.youtube.com/elipsesoftware)

[elipse@elipse.com.br](mailto:elipse@elipse.com.br)



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

**Microsoft Partner**

Gold Independent Software Vendor (ISV)