Driver JCI Metasys

Nome do Arquivo	Metasys.dll
Fabricante	Johnson Controls International
Equipamentos	Servidores Metasys versão 10 ou superior
Protocolo	API Metasys versões 3 ou 4
Versão	1.0.15
Última Atualização	24/08/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31314

Introdução

Este Driver implementa a comunicação com Servidores Metasys versão 10 ou superior da Johnson Controls International através da API Metasys versões 3 (três) ou 4 (quatro).

Configuração do Driver

A configuração deste Driver deve ser realizada na janela de propriedades. Adicionalmente, devem ser definidos manualmente os arquivos MetasysState0.txt e MetasysState1.txt contendo diversos nomes de estados que correspondem aos valores 0 (zero) e 1 (um), respectivamente, para que este Driver possa traduzir estes estados para os valores numéricos 0 (zero) e 1 (um).

Estes arquivos devem conter uma lista de palavras com os diversos estados, uma palavra em cada linha. Uma palavra que esteja no arquivo para o estado 0 (zero) não pode estar no arquivo de estado 1 (um) e vice-versa.

Todas as palavras são convertidas para minúsculas e as linhas que começam com o símbolo # são consideradas comentários, conforme os exemplos a seguir.

```
#Arquivo MetasysState0.txt
Parado
Desligado
Local
Fechar
Inativo
Inactive
FALSO
Desativar
Deactivate
Normal
```

```
#Arquivo MetasysState1.txt
Início
Start
Reset
Ligado
Remoto
Abrir
Active
Ativo
VERDADEIRO
Ativar
Activate
Alarme
```

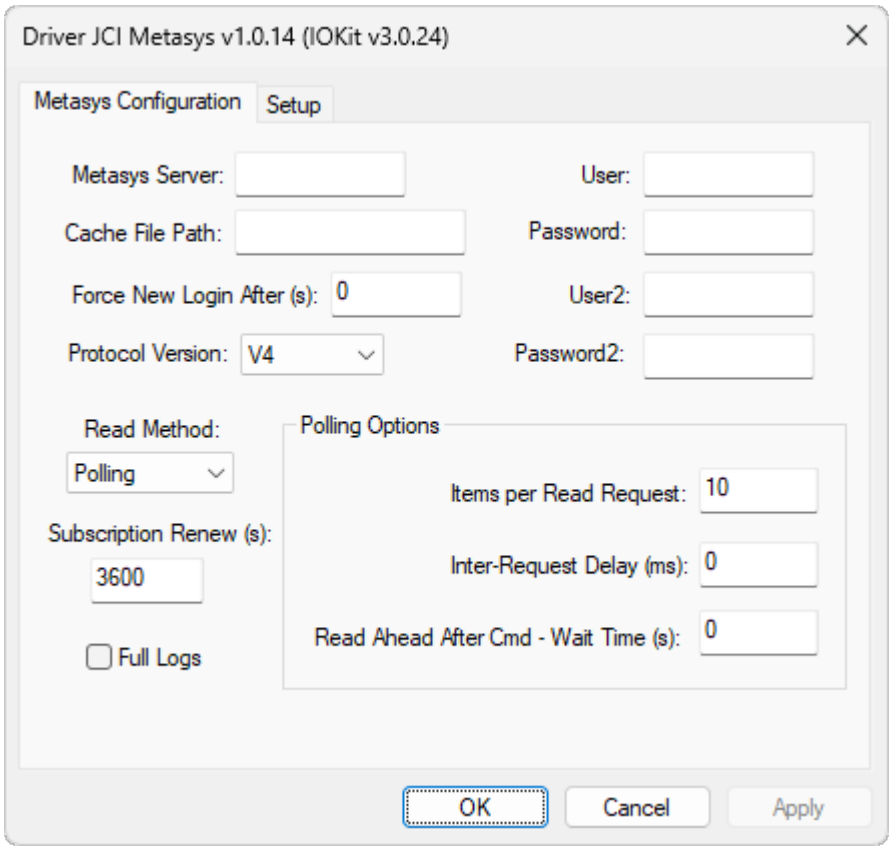
Também é possível criar um arquivo MetasysReadAhead.txt, que informa uma parte do nome de um comando associado a um Tag de estado para ser lido em sequência após a espera de um tempo configurável na opção **Read Ahead after Cmd - Wait Time**. Este arquivo é lido ao iniciar este Driver, com o conteúdo indicado no exemplo a seguir.

```
# Coloque neste arquivo as palavras-chave para associar
# um comando a um estado que deve ser lido o mais rápido possível
# no formato keyword_comando;keyword_estado
# Os objetos devem pertencer ao mesmo concentrador NAE
QDL-S;QDL-C
SF-COM;SF-STS
SYSENA;SF-STS
SF-CVF;SF-SVF
```

O procedimento de busca é realizado verificando se um comando que foi executado com sucesso contém a primeira palavra-chave, substituindo esta palavra-chave pela segunda palavra-chave e verificando se existe algum Tag com este nome. Se existir, então este Tag é colocado em uma fila especial, esperando o tempo indicado na opção **Read Ahead After Cmd - Wait Time** para ser lido.

Configuração das Propriedades

Os parâmetros de configuração para efetuar a comunicação com um sistema Metasys estão disponíveis na aba **Metasys Configuration** da janela de propriedades deste Driver, conforme a figura a seguir.



Aba Metasys Configuration

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

Opções disponíveis na aba Metasys Configuration

OPÇÃO	DESCRIÇÃO
Metasys Server	DNS (*Domain Name Server*) ou endereço IP de um servidor Metasys ou concentrador NAE
User e Password	Nome de usuário e senha para login
User2 e Password2	Nome de usuário e senha para login alternativo em caso de falha do usuário indicado nas opções **User** e **Password**. Este login alternativo acontece após 2 (duas) retentativas de login
Cache File Path	Nome de um diretório onde são salvos e lidos os arquivos de *cache* e dos significados das enumerações. Os arquivos disponíveis são **Arquivos de GUIDs (MetasysGUID.txt)**: Contêm os códigos internos de cada objeto Metasys, armazenados em arquivos de *cache* para acelerar o processo de inicialização, **Arquivos de Cache de Comandos e Estados (DeviceName_MetasysStateCache.txt)**: Contêm o significado de estados e comandos dos Tags com o parâmetro *N1* maior que 0 (zero), armazenados em um arquivo de *cache* por concentrador NAE para acelerar o processo de inicialização e tradução dos significados para valores 0 (zero) e 1 (um) e **Arquivos de Nomes de Estados (MetasysState0.txt e MetasysState1.txt)**: Contém os nomes dos estados que correspondem aos valores 0 (zero) e 1 (um), respectivamente. Se nenhum diretório é

OPÇÃO	DESCRIÇÃO
	informado, estes arquivos são salvos na mesma pasta do arquivo .dll deste Driver
Force New Login After (s)	Força um novo login independente da data de expiração de um *token*. Configurar esta opção em 0 (zero) desabilita um novo login
Protocol Version	Versão do protocolo usada por este Driver. Os valores disponíveis para esta opção são **V3**: Utiliza apenas leitura por *polling* ou **V4**: Permite a leitura por *polling* e por eventos ou assinaturas
Read Method	Método de leitura. Os valores disponíveis são **ByTag(N2)**: Permite selecionar, no parâmetro *N2* de um Tag, se a leitura é por *polling* ou por evento (este valor só pode ser usada se a opção **Protocol Version** está configurada com o valor **V4**), **Events**: Realiza a leitura de Tags por assinatura (este valor só pode ser usada se a opção **Protocol Version** está configurada com o valor **V4**) ou **Polling**: Realiza a leitura de Tags por *polling* (esta opção pode ser usada tanto com o valor **V3** quanto com o valor **V4** na opção **Protocol Version**)
Subscription Renew (s)	Ao selecionar o método de leitura de um Tag por eventos, uma nova assinatura é realizada se o valor deste Tag não muda dentro deste intervalo de tempo, em segundos, o que força a atualização deste Tag. Se o valor recebido é o mesmo do valor já existente, este Tag não é atualizado, exceto se a propriedade **EnableDeadBand** está configurada em Falso
Full Logs	Informa se os logs deste Driver, quando habilitados, são gerados com informações detalhadas da comunicação
Polling Options - Items per Read Request	Indica o número máximo de itens que podem ser solicitados a cada pedido de comunicação. O valor mínimo desta opção é 1 (um). Esta opção também é usada nas assinaturas de eventos para definir o número de itens que podem ser assinados em uma mesma mensagem
Polling Options - Inter-Request Delay (ms)	Informa um tempo padrão, em milissegundos, de atraso entre pedidos de leitura para todos os concentradores NAE. Este valor é usado se um concentrador NAE não tem um tempo de atraso informado no Tag **InterReqDelayMS**
Polling Options - Read Ahead After Cmd - Wait Time (s)	Informa um tempo de espera, em segundos, para realizar a leitura de um Tag de status associado a um comando. Esta associação é realizada no arquivo MetasysReadAhead.txt

Além da janela de propriedades, estas configurações também podem ser definidas em tempo de execução em aplicações **Elipse E3**, **Elipse Power** ou **Elipse Water**. Para isto, inicialize este Driver em modo **Offline**, ou seja, execute uma aplicação com a opção **Start driver OFFLINE** habilitada, configurável na aba **Setup** da janela de propriedades. As opções de configuração deste Driver estão descritas na tabela a seguir.

Opções de configuração do Driver JCI Metasys

PARÂMETRO	STRING OFFLINE	TIPO DE DADOS	DESCRIÇÃO
Metasys Server	Metasys.UseChecksum	Texto	Nome de um servidor Metasys
User	Metasys.User	Texto	Usuário para login
Password	Metasys.Password	Texto	Senha para login
Cache File Path	Metasys.GUIDPath	Texto	Diretório de *cache*
Items per Read Request	Metasys.MaxItemsRead	Texto	Número máximo de items por pedido de leitura
Protocol Version	Metasys.APIVersion	DWord	Versão do protocolo. Os valores possíveis são **3** (três) ou **4** (quatro)
Read Method	Metasys.ReadConfig	DWord	Método de leitura. Os valores possíveis são **0**: Por *polling*, **1**: Por eventos ou **2**: Por Tag
Inter Request Delay	Metasys.InterReqDelay	Dword	Tempo de espera entre leituras
Force New Login after (s)	Metasys.ForceLoginTime	Dword	Força um novo login após *n* segundos
Force New Login after consec errors	Metasys.ForceLoginConsecErrors	Dword	Força um novo login após *n* erros consecutivos
Force New Login with read error rate	Metasys.ForceLoginReadErrorRate	Dword	Força um novo login se a taxa de falha é superior a *n*
Read Ahead After Cmd - Wait Time	Metasys.ReadAheadWaitTime	Dword	Tempo de espera para leitura de status após um comando
Full Logs	Metasys.FullLogs	Int	Habilita logs completos

Estas propriedades podem ser escritas usando um Tag com o parâmetro *N1* igual a -1 (menos um), o parâmetro *N2* igual a 0 (zero), o parâmetro *N3* igual a 0 (zero) e o parâmetro *N4* igual a 3 (três).

Referência de Tags

Devem ser utilizados Tags individuais para a comunicação. Estes Tags são endereçados através dos parâmetros mostrados na tabela a seguir.

Device	Nome de um servidor Metasys ou concentrador NAE
Item	ObjetoMetasys;propriedade{Extra}. Para mais informações, consulte a seção **Endereçamento** a seguir
N1 ou B1	Tipo de objeto. Os valores possíveis são **0**: Padrão, **1**: Enumeração (convertida para número) ou **2**: Analógico com limites de alarme
N2 ou B2	Tipo de leitura, usado caso a opção **Read Method** seja configurada com o valor **ByTag(N2)**. Os valores possíveis são **0**: Por *polling* ou **1**: Por evento
N3 ou B3	Retentativas para leitura de limites de alarme caso o parâmetro *N1* seja configurado com o valor 2 (dois)
N4 ou B4	Não utilizado, deixe em 0 (zero)

Endereçamento

No parâmetro **Item**, o valor **ObjetoMetasys** é o nome de algum objeto em um sistema, e **propriedade** é um parâmetro opcional que informa a propriedade lida de um objeto. Se esta propriedade não é informada, a propriedade lida é **presentValue**. Caso seja informada a propriedade especial **PVALM**, são lidas as propriedades **presentValue** e **alarmState**, em que a primeira é mapeada para a propriedade **Value** de um Tag e a segunda é mapeada para a propriedade **Quality** de um Tag. Se o valor da propriedade **alarmState** é igual a **HighAlarm** ou **Alarm**, isto é mapeado para o campo **OPC_LIMIT_HIGH** (bit um da propriedade **Quality**) e se o valor é igual a **LowAlarm**, isto é mapeado para o campo **OPC_LIMIT_LOW** (bit zero da propriedade **Quality**), resultando respectivamente nos valores 194 (**OPC_GOOD** + **OPC_LIMIT_HIGH**) e 193 (**OPC_GOOD** + **OPC_LIMIT_LOW**).

O valor **Extra** é um texto no formato **JSON** que pode ser uma lista de estados discretos (*States*) ou o nome de um Alarme (*Alarm*). *States* é um texto no formato **JSON** em que se declara o significado dos estados para um Tag em específico e este Driver dá preferência à tradução de estados declarados em um Tag. Se as traduções não são encontradas, utiliza os textos declarados nos arquivos de estados 0 (zero) e 1 (um). O formato esperado para o valor *States* é o seguinte:

```
{
  "states": [
    {
      "Nome1": Valor1
    },
    {
      "Nome2": Valor2
    },
    {
      "NomeN": ValorN
    }
  ]
}
```

O valor **Alarm** é um texto no formato **JSON** com o nome de um Alarme, somente para Tags analógicos, conforme o exemplo a seguir.

```
{
  "alarmName": "Name"
}
```

Os limites **Low**, **High**, **Warning** e **Alarm** deste objeto de Alarme são buscados e salvos em um arquivo usando o Tag especial **GetAlarmLimits**. O código a seguir contém exemplos de Tags.

```
#Lê a propriedade presentValue
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1
#Lê a propriedade presentValue
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1;presentValue
#Lê a propriedade presentValue e a propriedade alarmState
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1;PVALM
#Lê a propriedade presentValue com o nome de um objeto de Alarme igual a "Alarm1"
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1{"alarmName":"Alarm1"}
#Lê a propriedade status de um objeto
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: FieldBus1.4oPav.FX34.BinaryInputs;status
#Lê a propriedade presentValue, interpretando o conteúdo conforme os estados Local, Remoto e
Automático
Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1{"states":[{"Local":0},{"Remoto":1},
{"Automático":2}]}
```

NOTA

Caso o valor obtido de uma propriedade seja uma enumeração e a propriedade *N1* é igual a 1 (um), este Driver tenta traduzir o texto desta enumeração para um valor correspondente, mapeando comandos e declarações de nomes de estados. Para a escrita, este Driver obtém o nome de um comando na propriedade e envia o que é recebido em um Tag, traduzindo para a enumeração correspondente, se necessário.

Tags Especiais

Esta seção contém informações sobre Tags especiais deste Driver.

CallStats

Somente Leitura

Device	Se este parâmetro é vazio, obtém todos os totais deste Driver. Para obter totais de um concentrador NAE específico, configure este parâmetro com o valor "Nome_Servidor_Metasys:Concentrador_NAE"
Item	Os valores possíveis são **CallStats5** (cinco minutos) ou **CallStats60** (uma hora)

Este Tag Bloco contém 12 Elementos, descritos a seguir:

- **Elemento 1**: **Total** (Total de chamadas da API)
- **Elemento 2**: **TImeout** (Operações com *time-out* ou com falha na chamada)
- **Elemento 3**: **Error** (Operações que retornam erro)
- **Elemento 4**: **ErrorCodes** (Lista com o código de erro retornado e as quantidades, separados por ponto e vírgula, como por exemplo "401:1234;403:57")
- **Elemento 5**: **OK** (Total de chamadas com sucesso)
- **Elemento 6**: **TotalReads** (Total somente de leituras de *polling*)
- **Elemento 7**: **TimeoutReads** (Operações de leitura com *time-out*)
- **Elemento 8**: **ErrorReads** (Operações de leitura que retornam códigos de erro diferentes de 200)

- **Elemento 9**: **OKReads** (Operações de leitura por *polling* com sucesso)
- **Elemento 10**: **TimeOKReadp50** (Tempo de resposta, percentil 50)
- **Elemento 11**: **TimeOKReadp95**(Tempo de resposta, percentil 95)
- **Elemento 12**: **TimeOKReadp99** (Tempo de resposta, percentil 99)

ClearStateCache

Leitura ou Escrita

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	ClearStateCache

Este Tag apaga o *cache* dos comandos e estados para um determinado concentrador NAE sem parar a execução de um sistema. Os valores possíveis são **0**: **Idle**, livre para iniciar, **1**: Em processamento ou **2**: Arquivo salvo e processo finalizado. Para iniciar um processo, escreva o valor 1 (um) com este Tag com o valor 0 (zero, **Idle**). Este Tag vai automaticamente para o valor 2 (dois) quando um processo é finalizado e permanece neste valor até que uma aplicação escreva o valor 0 (zero) e entrar em um estado livre para iniciar novamente, caso seja necessário.

FlushGUID

Somente Escrita

Device	Endereço de um concentrador NAE válido
Item	FlushGUID

Salva um arquivo contendo os GUIDs (*Globally Unique Identifiers*) de todos os concentradores NAE deste Driver.

GetAlarmLimits

Leitura ou Escrita

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	GetAlarmLimits

Este Tag solicita a leitura dos limites de alarme de Tags com o parâmetro *N1* igual a 2 (dois). Os valores possíveis para este Tag são **0**: **Idle** (Pronto para começar a coleta), **1**: **Pending** (Solicitação pendente), **2**: **Executing** (Em tratamento), **3**: **Finished OK** (Todos os objetos lidos) ou **4**: **Finished Error** (Ao menos um dos objetos não retornou os alarmes). Ao indicar os valores 3 (três) ou 4 (quatro), escreva o valor 0 (zero) para que este Driver armazene um arquivo NomeNAE_MetasysAlarmInfo.txt contendo uma linha para cada Tag, no formato "Path;GUID;Message;LowAlarm;HighAlarm;LowWarning;HighWarning;Delay;DelayActive".

GetDeviceEnumCount

Somente Leitura

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	GetDeviceEnumCount

Este Tag Bloco deve conter 2 (dois) Elementos. O primeiro Elemento retorna a quantidade total de objetos que contém ao menos um Tag do tipo declarado de enumeração. O segundo Elemento retorna, a partir da quantidade de objetos do Elemento anterior, quantos já tiveram o significado dos estados e comandos obtidos de um servidor Metasys.

GetGUID

Somente Leitura

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	GetGUID:ObjectName

Este Tag retorna o GUID (*Globally Unique Identifier*) de *ObjectName* se já foi obtido. Retorna erro se *ObjectName* não existe ou se ainda não foi traduzido.

InterReqDelayMS

Somente Escrita

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	InterReqDelayMS

Este Tag informa um tempo de atraso, em milissegundos, entre pedidos de leitura por concentrador NAE.

LoginStatus

Somente Leitura

Device	Nome_Servidor_Metasys:Concentrador_NAE
Item	LoginStatus

Este Tag informa se este Driver está logado em um servidor Metasys. O concentrador NAE informado no campo **Device** pode ser qualquer um, visto que este Driver realiza apenas um login, independente do número de concentradores NAE.

MaxItemsRead

Somente Escrita

Device	Metasys_Server_Name:NAE_Device
Item	MaxItemsRead

Este Tag informa o número máximo de itens lidos a cada pedido de leitura para este concentrador NAE em particular. Se este valor não é informado, usa o valor informado na opção **Items per Read Request** da janela de propriedades deste Driver.

SubscriptionStats

Somente Leitura

Device	Se este parâmetro é vazio, obtém todos os totais deste Driver. Para obter totais de um concentrador NAE específico, configure este parâmetro com o valor "Nome_Servidor_Metasys:Concentrador_NAE"
Item	Os valores possíveis são **SubscriptionStats5** (cinco minutos) ou **SubscriptionStats60** (uma hora)

Este Tag Bloco contém 9 (nove) Elementos, descritos a seguir:

- **Elemento 1**: **Total Subscribed** (Total de propriedades assinadas desde a inicialização deste Driver)
- **Elemento 2**: **Requested** (Total de chamadas com pedidos de assinatura)
- **Elemento 3**: **Accepted** (Total de chamadas com respostas OK ou código 200)
- **Elemento 4**: **Failed** (Total de chamadas com respostas com falha na chamada ou *time-out*)
- **Elemento 5**: **Error** (Total de chamadas com respostas não OK ou código diferente de 200)
- **Elemento 6**: **Updated Messages** (Mensagens recebidas com dados)
- **Elemento 7**: **Error Messages** (Mensagens de erro recebidas)
- **Elemento 8**: **Updated Objects** (Número de objetos atualizados)
- **Elemento 9**: **Updated Properties** (Número de propriedades atualizadas)

Documentação das Interfaces de Comunicação

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

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 **Elipse 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 **Elipse E3** versão 2.0 ou posterior, clique em **Configurar o driver** na barra de ferramentas de um Driver. No **Elipse 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.

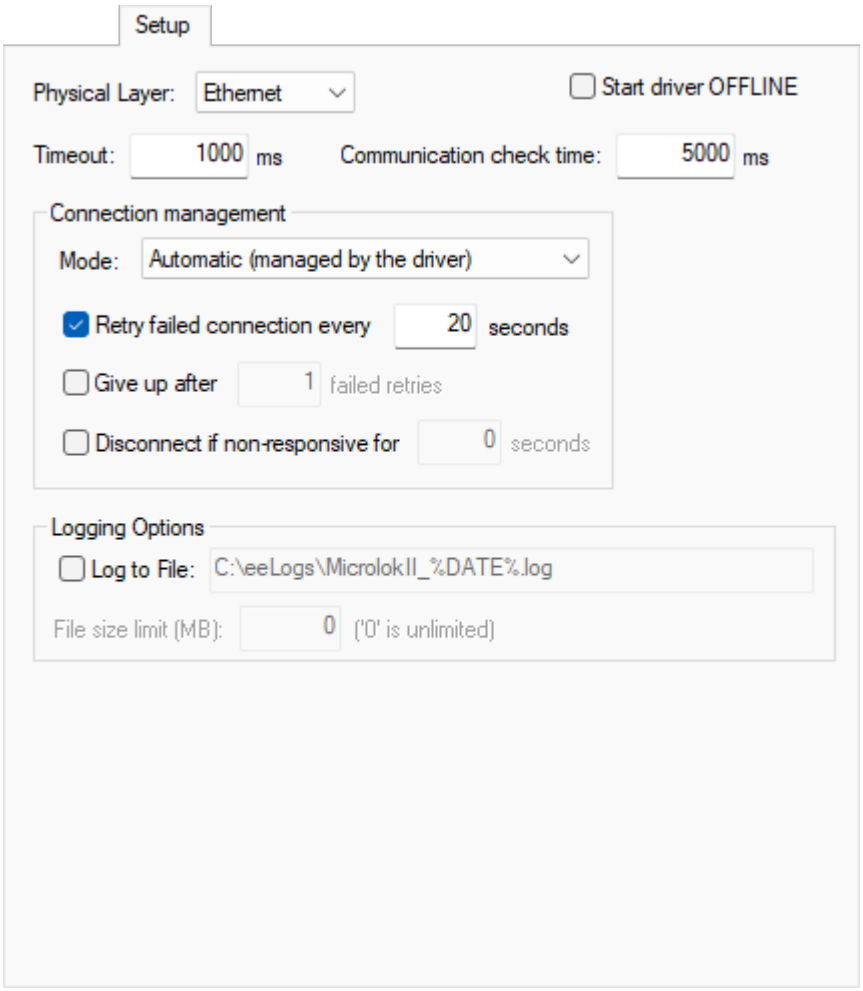
Caixa de Diálogo de Configuração

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



Aba Setup

Opções gerais da aba Setup

OPÇÃO	DESCRIÇÃO
Physical Layer	Selecione a interface física em uma lista. As opções disponíveis são **Serial**, **Ethernet**, **Modem** e **RAS**. A interface selecionada deve ser configurada na aba específica
Timeout	Configure o *time-out*, 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 *buffer* de recepção
Communication check time	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 **IO.CommunicationStatus**
Start driver OFFLINE	Selecione esta opção para que um Driver inicie em modo **Offline** ou parado. Isto significa que a interface de I/O não é criada até que se configure um Driver em modo **Online** 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
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection every ... seconds	Selecione esta opção para habilitar a retentativa de conexão de um Driver em um determinado intervalo, em segundos. Se a opção **Give up after failed retries** não está selecionada, este Driver continua retentando até que a conexão seja efetuada, ou que a aplicação seja parada
Give up after ... failed retries	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 **Offline**, 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
Disconnect if non-responsive for ... seconds	Habilite esta opção para forçar um Driver a se desconectar se nenhum byte chegou à interface de I/O no *time-out* especificado, em segundos. Este *time-out* deve ser maior que o *time-out* configurado na opção **Timeout**

Opções para o grupo Logging Options

OPÇÃO	DESCRIÇÃO
Log to File	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 **%PROCESS%** 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 **Elipse E3**, 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 **0FDAh**. Pode-se também utilizar a macro **%DATE%** no nome do arquivo. Neste caso é gerado um arquivo de log por dia, no formato **aaaa_mm_dd**. 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 **%DATE_HOUR%** gera um arquivo de log por hora, no formato **aaaa_mm_dd_hh**
File size limit (MB)	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

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

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	6 (seis)
Configuração por String	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

Tipo de Tag	Tag Bloco
Tipo de Acesso	Somente Leitura
Parâmetro B1	-1 (menos um)
Parâmetro B2	0 (zero)
Parâmetro B3	0 (zero)
Parâmetro B4	1 (um)
Propriedade Size	4 (quatro)
Propriedade ParamItem	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

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	2 (dois)
Configuração por String	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

Tipo de Tag	Tag Bloco
Tipo de Acesso	Somente Leitura
Parâmetro B1	-1 (menos um)
Parâmetro B2	0 (zero)
Parâmetro B3	0 (zero)
Parâmetro B4	3 (três)
Propriedade Size	2 (dois)
Propriedade ParamItem	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 × 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

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Leitura ou Escrita
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	4 (quatro)
Configuração por String	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

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

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

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

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

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

IO.Type

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

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1102
Configuração por String	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

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1103
Configuração por String	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

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1004
Configuração por String	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

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

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

IO.Stats.Total.TimeDisconnectedSeconds

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	0 (zero)
Parâmetro N4	1003
Configuração por String	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.

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.0.15	24/08/2026	M. Ludwig	• Atualização de documentação.
1.0.14	02/02/2026	M. Salvador	• Implementadas tags de estatísticas (*Case 38750*).
1.0.13	18/06/2025	M. Salvador M. Ludwig	• Corrigida uma situação que gerava logins excessivos em caso de falha de comunicação (*Case 36510*). • Implementado o bloqueio de Tags configurados com erros de sintaxe (*Case 36946*). • Agora novos equipamentos podem ser adicionados com este Driver em tempo de execução (*Case 38214*). • Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37987*).
1.0.9	21/03/2024	M. Salvador	• Adicionado suporte à versão 4 (quatro) da API.
1.0.8	20/09/2022	M. Salvador M. Ludwig	• Versão inicial deste Driver.

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

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

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

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

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

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

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

www.elipse.com.br
kb.elipse.com.br
forum.elipse.com.br
www.youtube.com/elipsesoftware
elipse@elipse.com.br

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

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)

Copyright © 2021-2026 Elipse Software Ltda. Todos os direitos reservados.