

Driver Eclipse SuperDriver

Nome do Arquivo	SuperDriver.dll
Fabricante	Eclipse Software
Equipamentos	Qualquer equipamento que comunique com Drivers da Eclipse Software desenvolvidos com a biblioteca IOKit e que possam comunicar em Ethernet TCP/IP
Protocolo	Qualquer protocolo sobre TCP/IP implementado em Drivers da Eclipse Software desenvolvidos com a biblioteca IOKit versão 2.0 e que possam comunicar em Ethernet TCP/IP
Versão	2.1.10
Última Atualização	06/08/2026
Plataforma	Win32
Dependências	N/A
Leitura com Superblocos	Não
Nível	31302

Introdução

O objetivo primário do Driver Eclipse SuperDriver é comunicar através de outros Drivers da **Eclipse Software**, podendo cada um destes Drivers ser especializado em algum protocolo. Múltiplos Drivers são associados ao Driver Eclipse SuperDriver, gerando itens chamados **Escravos**, que têm definições de configurações isoladas por **Perfis de Drivers**. Assim, os Tags de Comunicação existentes em uma aplicação, referenciados a estes **Escravos** por algum campo de configuração, têm os parâmetros repassados à respectiva instância de Driver apontada.

Os Drivers da **Eclipse Software** que podem ser subordinados ao Driver Eclipse SuperDriver devem ser aqueles desenvolvidos para a biblioteca **IOKit**, e a topologia deve proporcionar obrigatoriamente como camada física de comunicação imediata à aplicação a interface **Ethernet**, e como protocolo de transporte e rede o **TCP/IP**.

O Driver Eclipse SuperDriver é incompatível com o **Eclipse SCADA**.

Funcionamento do Driver

Para que o Driver Eclipse SuperDriver seja utilizado adequadamente em uma aplicação, é realmente importante entender como este Driver funciona.

Definições

É necessário familiarizar-se antes com alguns conceitos que são utilizados mais adiante neste documento, para que o funcionamento do Driver Eclipse SuperDriver seja bem entendido.

Perfil de Driver

Perfil de Driver é um objeto de uma estrutura de dados que contém um nome e um caminho para um arquivo da biblioteca de ligação dinâmica (DLL) de um Driver da **Eclipse Software**, além de uma declaração de configurações para este Driver. Vários Perfis de Drivers podem compartilhar o mesmo arquivo de Driver, cada um com uma configuração e um nome diferente.

Na aba de propriedades **SuperDriver** há uma lista de declarações de Perfis, que é explicada em capítulo posterior. Para que o Driver Eclipse SuperDriver funcione, uma lista de Perfis de Drivers deve ser configurada com pelo menos 1 (um) item.

Status de um Tag em Advise

O status **Em Advise** de um Tag é definido para que este Tag se mantenha em atualização constante, em intervalos de tempo definidos pela propriedade **Scan**, ou períodos de varredura.

A forma de mudar o status de um Tag para **Em Advise** é configurando a propriedade **AllowRead** para Verdadeiro e por uma das seguintes condições:

- Um Tag está com o valor da propriedade **AdviseType** igual a 0 (zero), ou seja, igual a **AlwaysInAdvise**
- Um Tag está com o valor da propriedade **AdviseType** igual a 1 (um), ou seja, igual a **AdviseWhenLinked**, e está associado a algum objeto ativo em uma Tela aberta

Caso alguma destas condições não esteja presente, um Tag de Comunicação não tem o status **Em Advise**, ou seja, não é lido ou escrito.

No caso de um serviço de escrita através do Driver Eclipse SuperDriver, não basta apenas habilitar a propriedade **AllowWrite** de um Tag, é preciso também que haja pelo menos um Tag de Comunicação com o status **Em Advise**, conforme a definição anterior, ativo no respectivo Perfil de Driver.

É somente nesta condição que um Tag de escrita pode ser adicionado na lista de *cache* de um **Escravo** para posterior processamento dentro dos períodos de varredura.

Portanto, não é possível realizar a escrita de um Tag de forma direta ou individualizada. Uma escrita sempre é processada via lista de *cache* de um **Escravo**, em conjunto com os demais Tags **Em Advise**.

Escravo

Entende-se por **Escravo** um objeto criado dinamicamente pelo Driver Eclipse SuperDriver, e que é portador de um *cache* (repositório) de Tags e de recursos de memória independentes, e relacionado a um Perfil de Driver de Comunicação. Os **Escravos** são criados pelo Driver Eclipse SuperDriver a partir do fornecimento de pares de Perfis e Portas pelos Tags declarados.

O par **Perfil** e **Porta** é fornecido pelo conteúdo do campo **Dispositivo**. O campo **Dispositivo** de cada Tag deve conter o nome de um dos Perfis declarados e um número de porta TCP/IP, separados pelo caractere de dois pontos. Esta sintaxe é apresentada mais adiante.

Existe um **Escravo** para cada par destes diferente que tenha sido declarado no conjunto de Tags subordinados ao objeto do Driver Eclipse SuperDriver em uma aplicação. Dez mil **Escravos** é o valor máximo suportado pelo Driver Eclipse SuperDriver.

Modos de Operação

O Driver Eclipse SuperDriver pode operar nos modos **Varredura Geral** (*General Scan*) ou **Varredura Individual** (*Individual Scan*). O modo **Varredura Geral** é executado sempre, para todos Escravos que possuem Tags **Em Advise**, e o modo **Varredura Individual** de um **Escravo** somente é executado se está habilitado.

Varredura Geral

O Driver Eclipse SuperDriver usa o modo **Varredura Geral**, o tempo de varredura geral existente na aba de configurações, para realizar a leitura e escrita de todos os Tags **Em Advise** de todos os **Escravos**. Cada vez que o modo **Varredura Geral** de cada **Escravo** expira, este inicia a leitura e escrita de todos os Tags, obedecendo à seguinte estratégia de conexão:

1. O **Escravo** inicia o respectivo objeto de Driver, sem habilitar a conexão automática a um endereço IP de *backup*.
2. O **Escravo** tenta processar todos os Tags. À medida que os Tags vão tendo sucesso, estes são retirados da lista de *cache*.
3. Ao final de uma iteração de serviços, o objeto de Driver é parado e deixa-se o recurso disponível, ou seja, a porta TCP/IP, caso haja um outro **Escravo** requerendo este recurso.
4. Existindo Tags na lista de *cache* ao final de uma primeira iteração de serviços, a leitura ou escrita é retentada até *n* vezes, conforme o número de retentativas configurado na aba de propriedades do Driver Elipse SuperDriver.
5. Se ainda há Tags na lista de *cache* ao final de todas as retentativas, e o número de Tags de Comunicação permanece igual ao número inicialmente requisitado, o Driver Elipse SuperDriver toma o endereço IP de *backup* como endereço IP principal e retenta novamente o ciclo de serviços com retentativas por este endereço IP. Isto evidentemente está condicionado ao que foi configurado na aba de propriedades. Se o usuário não deseja um endereço IP de *backup*, não há esta retentativa.
6. Se o número de Tags de Comunicação é inferior ao número inicialmente requisitado, isto é, se obteve sucesso em pelo menos um Tag, o processamento termina sem tentar o endereço IP de *backup*.

NOTA

Se há uma recusa de conexão por parte do endereço IP de destino, tanto para o endereço IP principal quanto para o endereço IP de *backup*, considera-se que um equipamento está em estado desligado, e portanto não são realizadas as retentativas programadas.

Varredura Individual

Assim como há o modo **Varredura Geral**, todos os Tags dos **Escravos** têm os *scans* individuais respeitados se existe **Em Advise** de um Tag **Varredura Individual** para o respectivo Perfil. As leituras e escritas realizadas por estes tempos não influenciam nos tempos de leitura e escrita do modo **Varredura Geral**, isto é, o serviço por **Varredura Individual** não reinicia o tempo que falta para o serviço por **Varredura Geral**. Este *scan* individual sem influência sobre o modo **Varredura Geral** é desejado para leituras e escritas provocadas ao longo do intervalo entre *scans* gerais, que dessincronizam a expiração de todos os Tags ao mesmo tempo. O modo **Varredura Geral** cuida, portanto, da atualização de todos os Tags **Em Advise** de todos os Perfis em ocorrências entre intervalos iguais para todos. Se há um ou mais Tags com *scan* individual expirado, estes são agrupados e colocados em uma lista de *cache*, obedecendo à seguinte estratégia de conexão:

1. O **Escravo** inicia o respectivo objeto de Driver, habilitando a conexão automática a um endereço IP de *backup*.
2. O módulo **IOKit** do respectivo objeto de Driver do **Escravo** se encarrega de automaticamente tentar um endereço IP principal ou um endereço IP de *backup*, de acordo com a configuração do gerenciamento de conexão automática.
3. O **Escravo** realiza o processamento de todos os Tags da lista de *cache*. À medida que os Tags vão tendo sucesso, estes são retirados desta lista.
4. Esta lista de Tags em **Varredura Individual** também retenta *n* vezes, conforme configurado na aba de propriedades, enquanto há Tags na lista de *cache*. A cada retentativa, o Perfil libera o recurso caso exista outro Perfil concorrente a este.
5. O processamento dos Tags individualizados termina com o sucesso de todo o grupo ou ao atingir o final das retentativas.

NOTA

Se há uma recusa de conexão por parte do endereço IP de destino, tanto para o endereço IP principal quanto para o endereço IP de *backup*, considera-se que um equipamento está em estado desligado, e portanto não são realizadas as retentativas programadas.

Execução

Quando colocado em execução, o Driver Elipse SuperDriver imediatamente cria os objetos **Escravos**. No momento em que um **Escravo** é criado pelo Driver Elipse SuperDriver, este **Escravo** entra em um estado de repouso até o momento em que um Tag entre no estado **Em Advise** e é colocado no *cache*. O Driver Elipse SuperDriver realiza um gerenciamento de todos os Tags declarados e que estejam no estado **Em Advise**. O algoritmo de gerenciamento dos Tags do Driver Elipse SuperDriver separa e adiciona Tags aos *caches* dos respectivos **Escravos**, à medida que estes Tags entram no estado **Em Advise**. A saída do estado **Em Advise** leva à retirada de um Tag do *cache*. Portanto, estar com o status **Em Advise** é condição para um Tag entrar em um *cache* de leitura e escrita de um **Escravo**.

A partir do momento que existe um ou mais Tags no *cache* de um **Escravo**, este **Escravo** imediatamente passa para um estado de leitura ou escrita e processa uma lista de Tags que precisam ser lidos ou escritos. Esta lista de Tags é processada conforme os passos a seguir:

1. Se há Tags de Comunicação:
 1. O Driver do objeto **Escravo** é iniciado e conecta-se ao dispositivo remoto.
 2. O Driver processa, um por um, os Tags de Comunicação.
 3. O Driver processa, um por um, os Tags especiais que são guardados na memória do **Escravo**.
 4. O Driver se desconecta e para.
2. Se há Tags especiais, estes Tags são atualizados com o conteúdo guardado na memória do **Escravo**.
3. Se há retentativas de serviços, retorna ao passo 1 (um).

Os Tags no estado **Em Advise** são sempre lidos ou escritos a cada expiração do tempo do modo **Varredura Geral**, e o período de varredura é definido pelo valor geral configurado na aba **SuperDriver**, ou seja, o gerenciamento de serviços do Driver Elipse SuperDriver não leva em consideração os valores de períodos de varredura, ou seja, a propriedade **Scan** de cada um dos Tags, a menos que o modo de varredura individual esteja habilitado para o respectivo Perfil destes Tags. Os Tags no estado **Em Advise** apenas são lidos ou escritos respeitando os *scans* individuais quando o modo **Varredura Individual** está habilitado.

Um **Escravo** volta ao estado de repouso enquanto aguarda até que o próximo Tag expire o tempo de *scan*. Acontecendo uma expiração de tempo, este **Escravo** monta uma nova lista de Tags a serem lidos ou escritos.

Este processo de execução descrito anteriormente é efetuado para qualquer **Escravo** subordinado ao Driver Elipse SuperDriver. Não é permitido que **Escravos** com portas TCP/IP iguais tenham as conexões realizadas ao mesmo tempo. Um **Escravo** que necessite de uma porta TCP/IP ocupada permanece em modo de espera até que este recurso seja desocupado.

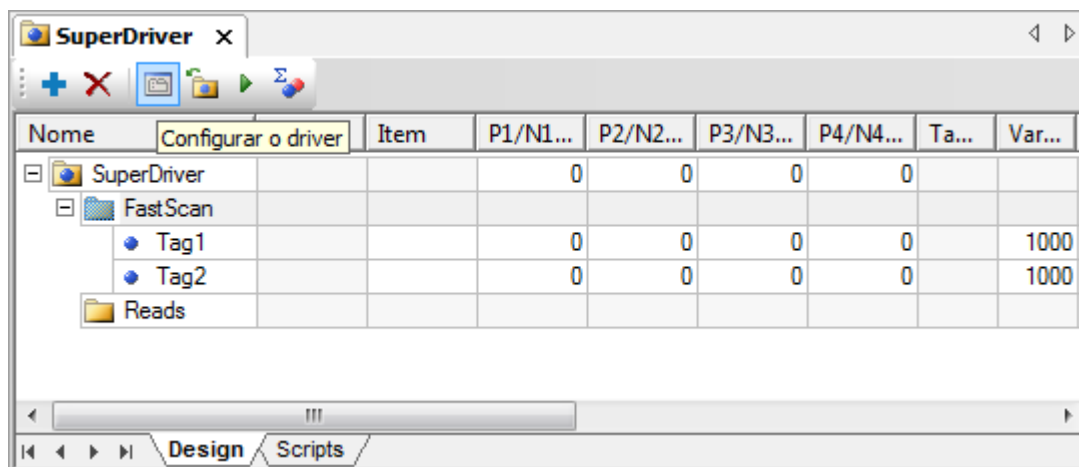
Comunicações Simultâneas

O Driver Elipse SuperDriver, quando em execução, gerencia a expiração dos tempos dos Tags **Em Advise** de todos os **Escravos**. Ao detectar a expiração dos tempos de um ou mais Tags, o Driver Elipse SuperDriver salva uma lista de **Escravos** com Tags de tempos expirados para terem leituras executadas tão logo seja possível. Esta lista pode ter o tamanho máximo de **Escravos** possíveis. No entanto, as tarefas de comunicação de todos os **Escravos** nem sempre podem ocorrer simultaneamente.

Toda tarefa de comunicação que ocorre em um **Escravo** consome recursos de processamento, ou uso de CPU, e ao menos um encadeamento de execução próprio (*thread*) é criado para cada comunicação. O sistema operacional limita o número máximo de *threads* por processo. Por esta razão, o Driver Elipse SuperDriver possui uma limitação máxima de comunicações ocorrendo ao mesmo tempo. Quando este limite é atingido, todos os **Escravos** que precisam comunicar permanecem em uma lista de espera, e entram em comunicação à medida que as comunicações atuais sejam terminadas.

Configuração do Driver

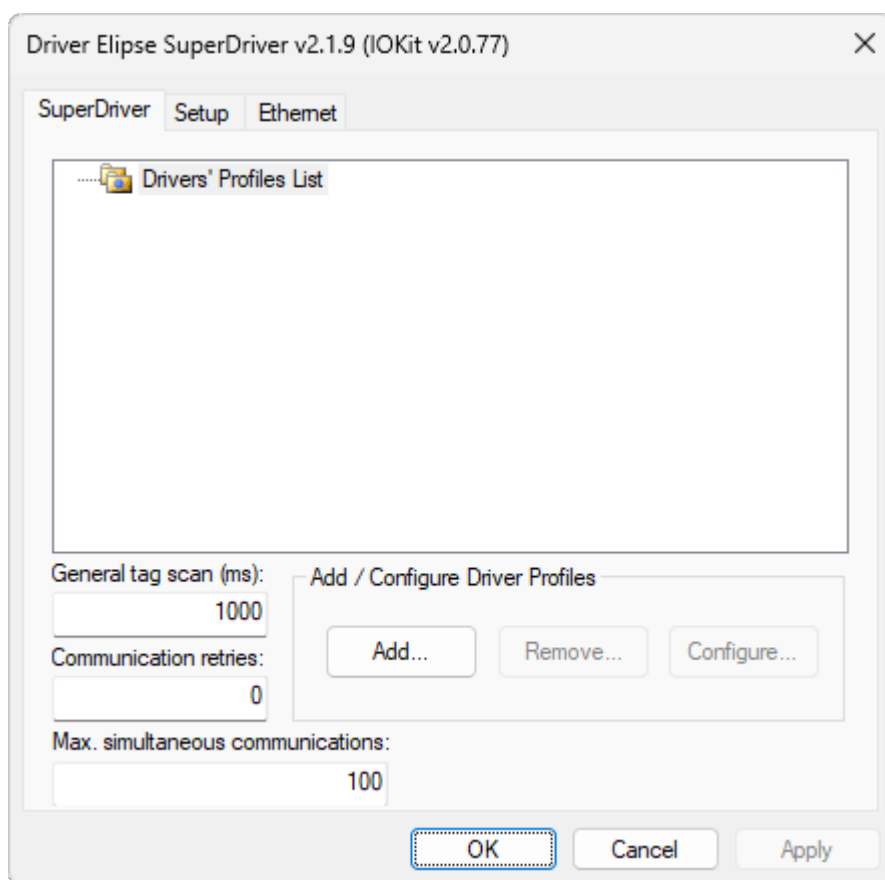
O Driver Elipse SuperDriver não usa parâmetros [P] para configuração. Todas as configurações são realizadas na janela de propriedades, disponível na aba **Design** de um objeto Driver de Comunicação. Para isto, clique em **Configurar o Driver** , conforme a figura a seguir.



Opção Configurar o driver

Aba SuperDriver

Ao clicar em **Configurar o Driver** , a janela da figura a seguir é aberta e a aba **SuperDriver** é selecionada.



Aba SuperDriver

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

Opções disponíveis na aba SuperDriver

OPÇÃO	DESCRIÇÃO
Drivers' Profiles List	Estrutura em árvore destinada à exibição da lista de Perfis de Drivers
General tag scan (ms)	Tempo, em milissegundos, do período de varredura, ou <i>scan</i> , relativo a todos os Tags que estão na lista de leitura e universal a todos os objetos Escravos . Isto significa que todos os Escravos procuram obedecer uma leitura cíclica de cada um dos Tags por este valor. A expiração do tempo de <i>scan</i> geral sempre gera um pedido de leitura de todos os Tags em lista de cada Escravo
Communication retries	Configura o número de retentativas de comunicação. Um objeto Escravo retenta a comunicação enquanto há Tags com leituras mal-sucedidas. Em cada modo de operação, o Driver Eclipse SuperDriver age de maneira diferente com relação a este valor. Para mais informações, consulte o tópico Modos de Operação . O número máximo permitido de retentativas é 10
Max. simultaneous communications	Configura o limite máximo de comunicações ocorrendo ao mesmo tempo. Para mais informações, consulte o tópico Comunicações Simultâneas . Quando este valor é atingido, todos os Escravos que precisam comunicar permanecem em uma lista de espera e entram em comunicação à medida que as comunicações atuais são terminadas. Este valor deve ser levado em conta pelas limitações físicas do processador do computador que executa o sistema. O valor mínimo configurável é 1 (um) e o valor máximo é 750. O melhor valor a ser configurado varia conforme a capacidade do processador do sistema em realizar tarefas em paralelo, portanto nem sempre o maior valor possível é o valor ótimo para o sistema, tal como em uma relação diretamente proporcional. O valor ótimo pode ser obtido por meio de testes com leituras simultâneas de um número de Escravos superior ao valor configurado
Add/Configure Driver Profiles	Permite adicionar, remover e configurar Perfis de Drivers. Para mais informações, consulte o tópico Configuração de Perfil

Clique em **OK** para confirmar as configurações listadas e fechar esta janela.

Abas Setup e Ethernet

As abas **Setup** e **Ethernet** são abas herdadas da biblioteca **IOKit** e são as únicas que devem ser levadas em conta. Consulte o tópico **Documentação das Interfaces de Comunicação** para mais informações sobre estas abas.

Diversos parâmetros configurados nestas abas são repassados aos Drivers **Escravos**. O tópico **Parâmetros dos Perfis Herdados do SuperDriver** especifica cada um destes parâmetros e deve ser consultado para mais informações sobre as configurações necessárias em cada Perfil.

Configuração de Perfil

Clique em **Add** ou **Configure** na aba **SuperDriver** para abrir a janela da figura a seguir e adicionar ou configurar um Perfil.

Janela Add/Configure Profile

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

Opções disponíveis na janela Add/Configure Profile

OPÇÃO	DESCRIÇÃO
Profile name	Configura o nome de um Perfil. Selecione um nome que represente este Perfil de Driver ao referenciar Tags
Description	Configura a descrição de um Perfil com a finalidade de documentação. Apesar de não impactar no funcionamento de um Perfil, é uma opção obrigatória que pode ser preenchida com qualquer conteúdo com entrada por teclado
Driver path and filename	Caminho e nome de um arquivo de Driver selecionado para um Perfil. Clique em Browse para abrir a janela de seleção de arquivos padrão do Windows e selecionar um arquivo existente
Configure Driver	Abre a janela de configuração do Driver indicado na opção Driver path and filename . Como cada Driver tem particularidades, esta janela de configuração deve ser preenchida conforme o Manual do Usuário do respectivo Driver. Consulte o tópico Parâmetros dos Perfis Herdados do SuperDriver para evitar configurações desnecessárias

Parâmetros dos Perfis Herdados do SuperDriver

Os parâmetros a seguir são copiados da configuração do Driver Eclipse SuperDriver para todos os Perfis subordinados, portanto não precisam ser preenchidos em todos os Perfis.

- Aba **Setup**:
- **Physical layer (Camada física)**: Fixo em **Ethernet** e não prevê comunicação em outras interfaces
- **Connection management mode (Modo de gerenciamento de conexão)**: **Automático** ou **Manual**

- **Retry failed connection every n seconds**
- **Give up after n failed retries**
- **Disconnect if non-responsive for n seconds**
- **Ativação e nome do arquivo de Log:** Gera um arquivo de log para cada **Escravo**, cujo nome de arquivo é composto pelo nome configurado no Driver Elipse SuperDriver com um sufixo composto pelo nome do Perfil e da Porta TCP/IP
- **Aba Ethernet:**
 - **Transport (Protocolo de transporte):** Fixo em **TCP** e não prevê outro protocolo de transporte
 - **Endereço IP principal:** Endereço IP de um servidor
 - **Backup address:** Endereço IP de *backup*
 - **Especificação de porta local (tanto para o IP principal quanto para o backup)**
 - **Ping:** Habilitação, *time-out* e retentativas

NOTAS

- As demais configurações nas abas **Setup** e **Ethernet** do Driver Elipse SuperDriver são ignoradas e vale a configuração de cada Perfil.
- A porta TCP/IP da aba **Ethernet** é desconsiderada. Este parâmetro é especificado nos Tags.

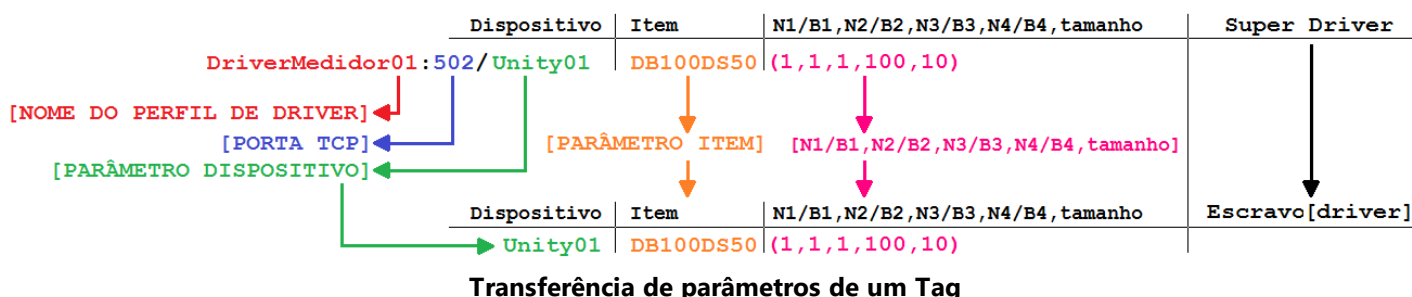
Referência de Tags

Esta seção contém informações sobre a configuração dos parâmetros *Dispositivo*, *Item* e *[N/B]* de cada Tag deste Driver. Os Tags são classificados nas categorias **Tags de Comunicação** e **Tags Especiais**.

Tags de Comunicação

Considera-se como **Tag de Comunicação** um Tag que é executado exclusivamente pela instância de Driver interna a um **Escravo**, através do repasse de parâmetros intermediado por este **Escravo**. O Driver Elipse SuperDriver cria um Tag virtual que é repassado para leitura ou escrita por um **Escravo**.

A figura a seguir esquematiza como é realizada a transferência de parâmetros de um Tag declarado no Driver Elipse SuperDriver para que um Tag virtual seja lido pelo Driver de um **Escravo**.



A documentação de ajuda de cada Driver utilizado como **Escravo** deve ser consultada para configuração dos parâmetros *N/B*, tamanho de Blocos e eventualmente os parâmetros *Dispositivo* e *Item*, caso estes sejam utilizados.

- **Dispositivo:** Para especificar um **Escravo** ao qual um Tag pertence, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]**. Caso o Driver **Escravo** utilize algum parâmetro *Dispositivo*, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/[PARÂMETRO DISPOSITIVO]**, obedecendo a documentação de ajuda
- **Item:** Caso o Driver **Escravo** utilize algum parâmetro *Item*, obedeça o que deve ser preenchido, segundo a documentação de ajuda
- **Parâmetros N/B e de tamanho:** Preencha conforme a documentação de ajuda

Os campos de valores de cada Elemento e de cada Tag PLC também têm as interpretações esclarecidas mediante a consulta da documentação de ajuda.

Tags Especiais

Considera-se um **Tag Especial** aquele Tag que é executado lendo apenas conteúdos memorizados por um **Escravo** e que podem realizar alguma função. Estes Tags, quando lidos, não dependem do carregamento, inicialização ou conexão do Driver de Comunicação de um **Escravo**, por isto têm um desempenho superior, e períodos de *scan* inferiores em relação à Tags de Comunicação podem ser exigidos. Os tipos de Tags especiais estão descritos nos tópicos a seguir.

IOKit

Para configurar um Tag da biblioteca **IOKit**, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/[TAG DO IOKIT]** no campo **Dispositivo** (*Device*), conforme o exemplo a seguir. Os demais campos são ignorados.

```
Profile02:502/IO.PhysicalLayerStatus [Tag PLC]
Profile02:502/IO.IOKitEvent [Tag Bloco com quatro Elementos]
```

Consulte o tópico **Documentação das Interfaces de Comunicação** para informações sobre os Tags da biblioteca **IOKit**.

IndividualScan

Este Tag, quando em modo **Advise**, indica a um **Escravo** subordinado ao Driver Elipse SuperDriver que deve realizar leituras dos Tags, respeitando os períodos de varredura individualmente configurados.

Para configurar um Tag **IndividualScan**, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/IndividualScan** no campo **Dispositivo**. Os parâmetros *[N]* são ignorados. A propriedade **Value** deste Tag mostra um valor numérico, incrementado a cada leitura realizada, conforme o exemplo a seguir.

```
Profile01:502/IndividualScan
```

CurrentMainIP

Este Tag indica o endereço IP ativo de uma conexão. Para configurá-lo, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/CurrentMainIP** no campo **Dispositivo** (*Device*). Os demais campos são ignorados. A propriedade **Value** deste Tag mostra uma **String** com o endereço IP ou *alias* do servidor ativo, conforme o exemplo a seguir.

```
Profile01:502/CurrentMainIP
```

ErrorHistory

Este Tag retorna uma lista de erros ocorridos durante a comunicação com o Driver de um **Escravo**. Quando não há erros, retorna uma lista vazia. Este Tag deve ser um Tag Bloco com até 2 (dois) Elementos. Para configurar este Tag, utilize a sintaxe **[NOME DO PERFIL DE DRIVER]:[PORTA TCP]/ErrorHistory** no campo **Dispositivo**. Os demais campos são ignorados, conforme o exemplo a seguir.

```
Profile01:502/ErrorHistory
```

A propriedade **Value** do Elemento 1 (um) contém um código numérico de erro. A interpretação do código obedece à **tabela** a seguir.

A propriedade **Value** do Elemento 2 (dois) contém uma **String** com o significado do erro, em inglês. Cada Driver possui um modo de exibir o significado de um erro. Portanto, para manter um padrão ou mesmo alterar o idioma, realize uma correspondência entre a tabela de códigos de erro e a aplicação.

Este Tag somente pode ser usado nos seguintes Drivers:

- Enron Modbus versão **1.2**
- Modicon Modbus versão **2.8**
- Elster-Instromet 999 versão **1.0**

Códigos de erro

CÓDIGO	CATEGORIA	DESCRIÇÃO
1	TIMEOUT	Um Driver não recebeu resposta de um equipamento no tempo de <i>time-out</i> configurado
2	RECEBIDO FRAME INVÁLIDO	Um Driver recebeu um <i>frame</i> inválido como resposta, como por exemplo um erro de checagem ou um campo de endereço diferente do solicitado
3	ARGUMENTO INVÁLIDO	Uma aplicação de usuário transferiu Tags com parâmetros inválidos para um Driver
4	ERRO DE CONEXÃO	Erro relacionado à camada física, como por exemplo uma falha na conexão TCP/IP ou porta serial inválida
5	SEM MEMÓRIA	O computador em que uma aplicação está em execução está sem memória, impedindo a alocação de recursos para as operações de um Driver
6	ERRO INTERNO	Este erro é retornado em casos inesperados, isto é, situações ou contextos detectados por um Driver que não podem ocorrer e que podem ser considerados como falhas. Esta categoria de erro é mais rara e precisa ser encaminhada ao suporte técnico da Elipse Software
7	RECEBIDO FRAME DE EXCEÇÃO DO PROTOCOLO	Um <i>frame</i> de resposta válido foi recebido de um equipamento, porém informando uma condição de erro prevista por protocolo

Tags de Estatísticas e Bloqueio de Varreduras

Durante a execução, o Driver Elipse SuperDriver acumula contagens para estatísticas que podem ser acessadas por uma aplicação através de Tags específicos de Estatísticas de Coleta. Todos estes Tags são somente de leitura, com exceção do Tag **\$BlockScans**, que bloqueia todas as varreduras de todos os **Escravos** do Driver Elipse SuperDriver.

Para que um Tag seja reconhecido, o nome deve estar no campo **Dispositivo** (*Device*) e o campo **Item** deve estar em branco.

- **\$ScansActive**: Retorna a quantidade de varreduras em execução neste instante
- **\$ScansCompleted**: Retorna a quantidade de varreduras que um Driver já completou
- **\$IndividualScansActive**: Retorna a quantidade de varreduras individuais em execução neste instante
- **\$IndividualScansCompleted**: Retorna a quantidade de varreduras individuais que um Driver já completou
- **\$GeneralScansActive**: Retorna a quantidade de varreduras gerais em execução neste instante
- **\$GeneralScansCompleted**: Retorna a quantidade de varreduras gerais que um Driver já completou
- **\$BlockScans**: Permite bloquear a execução de scans. Os valores possíveis para este Tag são **0**: Bloqueia todas as varreduras ou **1**: Libera todas as varreduras. Este Tag também permite escritas

Parâmetros de Configuração

Somente Escrita

B1	-1 (menos um)
B2	0 (zero)
B3	0 (zero)
B4	3 (três)

Existe a possibilidade de alterar os parâmetros configurados na aba **SuperDriver** em tempo de execução, com este Driver em modo **Offline**. Todas as configurações *offline* utilizam a escrita de um Tag Bloco com 2 (dois) Elementos. Os tópicos a seguir mostram cada um dos Tags configuráveis desta forma. Para mais informações sobre o uso deste Tag, consulte o tópico **Documentação das Interfaces de Comunicação**.

GeneralTagScan

Utilize os Elementos a seguir para alterar o tempo do período de varredura geral de todos os Tags em modo **Advise**:

- **Elemento 1**: "SUPERDRIVER.GeneralTagScan"
- **Elemento 2**: Tempo do período de varredura, ou *scan*, em milissegundos

CommunicationRetries

Utilize os Elementos a seguir para alterar o número de retentativas de comunicação:

- **Elemento 1**: "SUPERDRIVER.CommunicationRetries"
- **Elemento 2**: Número de retentativas. O valor máximo permitido é 10

Documentação das Interfaces de Comunicação

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

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

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
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 <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
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 <i>time-out</i> especificado, em segundos. Este <i>time-out</i> deve ser maior que o <i>time-out</i> 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

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

Main IP:		Port:	502	☐ Local port:	0
☐ Backup IP 1:		Port:	0	☐ Local port:	0
☐ Backup IP 2:		Port:	0	☐ Local port:	0
☐ Backup IP 3:		Port:	0	☐ Local port:	0

Aba Ethernet

Opções disponíveis na aba Ethernet

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

OPÇÃO	DESCRIÇÃO
	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: • Timeout: Especifique o número de milissegundos de espera por uma resposta de um comando ping. Deve-se usar um comando ping 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 • Retries: Número de retentativas de um comando ping, sem contar a tentativa inicial. Se todas as tentativas falharem, então a conexão com o <i>socket</i> é abortada

Opções disponíveis no grupo Connect to

OPÇÃO	DESCRIÇÃO
Main IP	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"
Port	Digite a porta IP de um dispositivo remoto, entre 0 (zero) e 65535
Local port	Selecione esta opção para utilizar uma porta IP local fixa ao conectar a um dispositivo remoto
Backup IP 1, 2 e 3	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

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

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

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.

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

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

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

Tipo de Tag	Tag de Comunicação
Tipo de Acesso	Somente Leitura
Parâmetro N1	-1 (menos um)
Parâmetro N2	0 (zero)
Parâmetro N3	4 (quatro)
Parâmetro N4	2 (dois)
Configuração por String	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]

A 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]

9 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]

9 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

A 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
2.1.10	06/08/2026	M. Ludwig	Driver atualizado para a biblioteca IOKit versão 3.0 e Visual Studio 2022 (<i>Case 38047</i>).
2.1.7	13/02/2017	M. Ludwig	Corrigido um travamento na finalização em condições de estresse (<i>Case 21663</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Corrigido um problema que disparava mais de uma coleta de forma simultânea no mesmo Perfil ou porta TCP/IP e poderia causar um GPF (Case 21751).
2.1.4	17/10/2016	M. Ludwig	- Corrigida a falta de carregamento das configurações em Perfis de Drivers para Escravos (Case 21664). - Driver portado para a biblioteca IOKit versão 2.0.72 (Case 21634).
2.1.2	17/03/2015	M. Ludwig	- Implementada a desistência das retentativas de comunicação caso se detecte um equipamento em modo Offline (Case 18356). - Implementada a sobrescrita das configurações Ethernet para evitar acionamento de redundância automática inadvertida em modo Varredura Geral (Case 18389).
2.1.1	14/11/2014	M. Ludwig	- Implementado um limite máximo de comunicações simultâneas (Case 15645). - Alterado o limite de Escravos para 1000. Ao atingir este limite, este Driver não cria mais Escravos (Case 17564).
2.0.5	21/03/2014	M. Ludwig	Adicionada uma limitação nas conexões de Escravos (Case 15976).
2.0.4	21/11/2013	F. Englert M. Ludwig	- Corrigida a possibilidade de travamento ao fechar este Driver (Case 14855). - Implementada uma segunda chamada de escritas após a execução de um processo de leitura (Case 15176).
2.0.3	27/09/2013	F. Englert	Adicionados Tags de estatísticas de coleta (Case 14954).
2.0.2	23/09/2013	F. Englert M. Ludwig	Corrigidos vazamentos de memória (<i>heaps</i> do

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			Windows) durante a execução (Case 14918). • Driver portado para a biblioteca IOKit versão 2.0 (Case 13888). • Implementada a geração de logs dos sub-Divers em arquivos diferentes (Case 13890). • O Driver Elipse SuperDriver agora só carrega Drivers que suportem múltiplas instâncias no mesmo processo (Case 13995).
1.2.2	05/03/2013	M. Ludwig	• Alterado o comportamento de espera no encerramento (Case 13139). • Corrigida a atualização de parâmetros em tempo de execução (Case 13358). • Corrigida a falta de liberação de recursos nas escritas de Tags de comunicação (Case 13484). • Corrigido o crescimento indefinido de uso de recursos quando o Tag ErrorHistory não é lido (Case 13477). • Corrigido o carregamento incorreto de endereços IP configurados na janela de propriedades (Case 13476).
1.1.1	31/05/2012	C. Mello	• Adicionado um serviço de escrita de Tags para Perfis de Driver com Tags em modo Advise (Case 12521).
1.0.1	22/09/2011	M. Ludwig	• Versão inicial deste Driver (Case 11581).

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 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 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 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, 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)