Driver IEC ICCP Client/Server

Nome do Arquivo	ICCP.dll
Fabricante	International Electrotechnical Commission
Equipamentos	
Protocolo	ICCP - TASE.2 (MMS) sobre TCP/IP Versão 2000-08
Versão	2.0.36
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 ou superior
Leitura com Superblocos	Não
Nível	31300

Introdução

O Driver IEC ICCP Client/Server comunica com outros **Clientes** ou **Servidores** ICCP sobre Ethernet TCP/IP. Cada instância deste Driver permite Perfis de comunicação de **Clientes**, **Servidores** e **Cliente/Servidor** com outros Centros.

Este Driver, conforme definido pela norma 60870-6, exige a definição de uma Tabela Bilateral que especifica variáveis, parâmetros e permissões para o intercâmbio de dados entre Centros que utilizam o protocolo TASE.2. O formato, a configuração ou a manutenção desta tabela está fora do escopo deste padrão.

Para facilitar o intercâmbio desta tabela entre diferentes fabricantes de software, o Driver IEC ICCP Client/Server usa o formato de arquivo **CSV** (*Comma-Separated Values*) para a definição da Tabela Bilateral para **Clientes** ou **Servidores**. Estes arquivos, chamados ICCPClient.blt do lado **Cliente** e ICCPServer.blt do lado **Servidor**, têm uma estrutura interna que deve ser seguida pelos usuários.

Para criar estes arquivos, existem as possibilidades descritas a seguir:

- Se este Driver funciona como um **Cliente** ICCP, pode-se criar o arquivo de declaração do ponto de leitura ICCPClient.blt a partir de um **Servidor** ICCP *Online*
- Usar um arquivo no formato do Excel, fornecido com este Driver, exportá-lo para o formato **CSV** e renomeá-lo para ICCPClient.blt ou ICCPServer.blt
- Importar um arquivo XML no formato **Sisco AXS4**
- Editar manualmente um arquivo com um editor ASCII, como por exemplo o Notepad
- Editar, em tempo de execução, os Tags de operação deste Driver

Tags **Cliente** ou **Servidor** no **Elipse E3** ou **Elipse Power** podem ser criados automaticamente em modo **Offline** usando a janela **Tag Browser**, se os arquivos já foram criados.

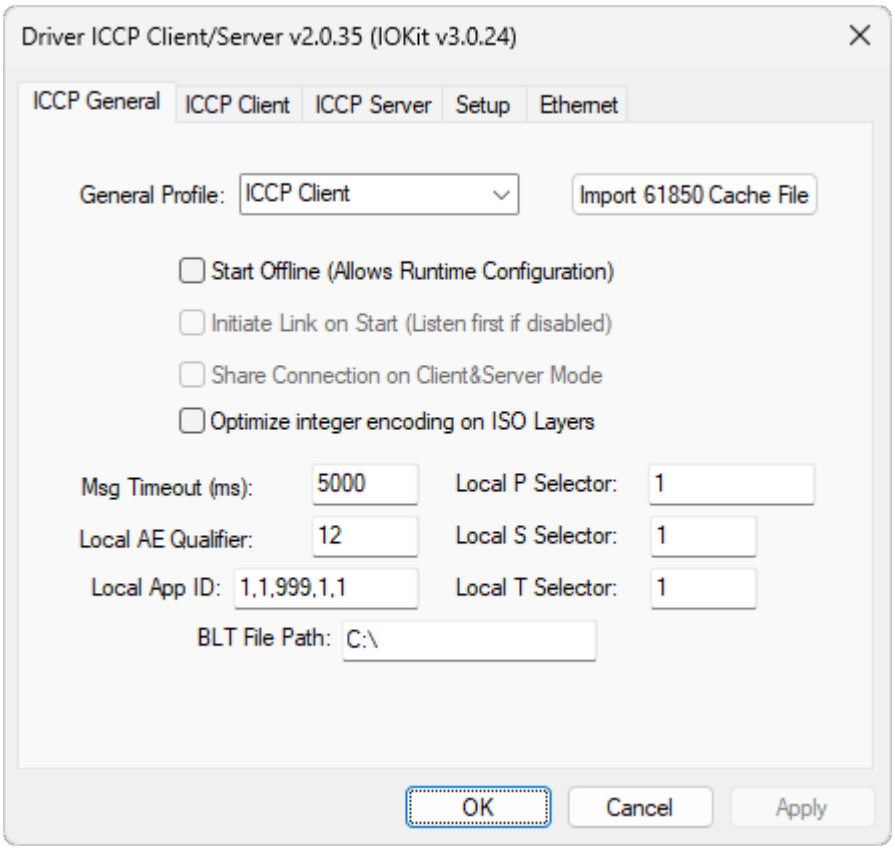
Este Driver implementa os Blocos ICCP 1 (um, *Basic Services*), 2 (dois, *Extended Conditions*), 3 (três, *Blocked Transfers*) e 5 (cinco, *SBO Device Control*).

Configuração

Esta seção contém informações sobre a configuração do Driver IEC ICCP Client/Server. Os parâmetros **[P]** não são usados e todas as configurações são realizadas na janela de propriedades deste Driver, nas abas **ICCP General**, **ICCP Client** e **ICCP Server**, descritas nos tópicos a seguir.

Aba ICCP General

Esta aba contém configurações de parâmetros locais e gerais.



Aba ICCP General

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

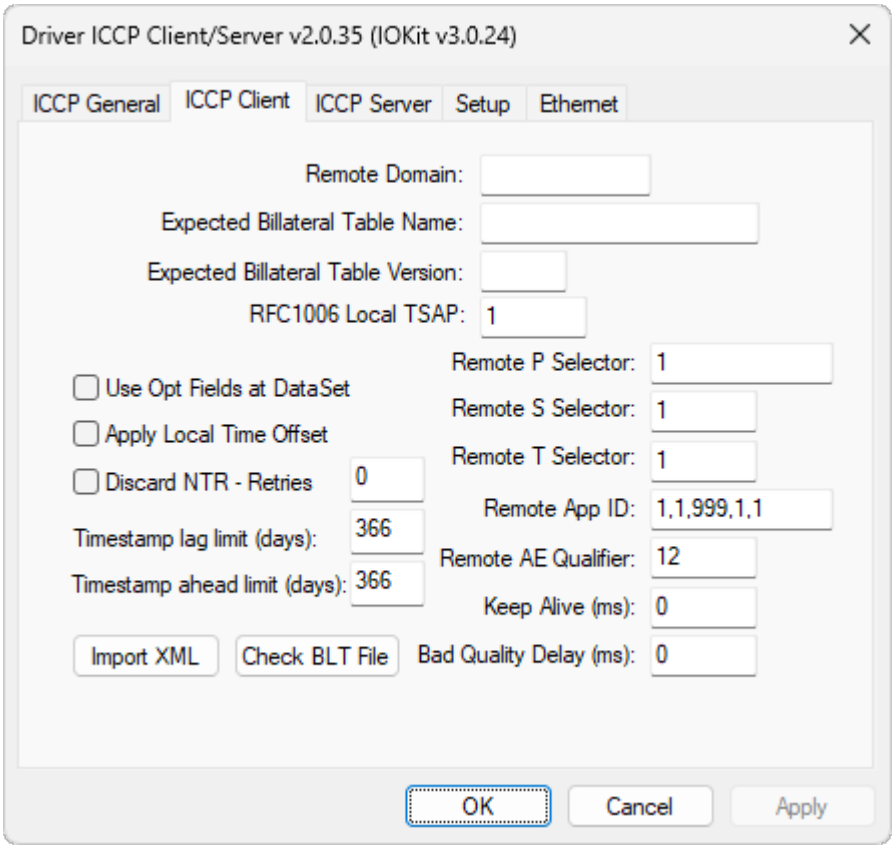
Opções disponíveis na aba ICCP General

OPÇÃO	DESCRIÇÃO
General Profile	Informa o modo de operação deste Driver. As opções disponíveis são **ICCP Client**: Este Driver opera como um **Cliente** ICCP, coletando dados de um **Servidor** ICCP, **ICCP Server**: Este Driver opera como um **Servidor** ICCP, enviando dados para um **Cliente** ICCP ou **ICCP Client/Server**: Este Driver opera como **Servidor** ICCP e **Cliente** ICCP ao mesmo tempo. Para esta última opção, selecione a opção **Initiate Link on Start (Listen first if disabled)** para que este Driver inicie a comunicação, ou seja, funcione inicialmente como um **Cliente** ICCP
Import 61850 Cache File	Gera um arquivo BLT básico a partir de arquivos **Logical Device Cache**, com a extensão .ld, gerados pelo Driver IEC 61850 da **Elipse Software**. Use esta opção caso uma aplicação **Elipse E3** ou **Elipse Power** local leia dados de

OPÇÃO	DESCRIÇÃO
	um equipamento IEC 61850 para fornecer dados como **Servidor** ICCP para outros Centros
Start Offline (Allows Runtime Configuration)	Configura este Driver para esperar pelos parâmetros de configuração antes de ser executado. Ao utilizar esta opção, depois de carregado em memória, este Driver permanece no status **Idle**, em que uma aplicação pode configurar qualquer parâmetro definido na janela de configuração. Esta operação é realizada utilizando os Tags **Set Parameter** da biblioteca **IOKit**. Para mais informações, consulte o tópico **Documentação das Interfaces de Comunicação**. Cada parâmetro corresponde a um identificador textual, listado no tópico **Parâmetros deste Driver**, e um valor. Depois de configurados todos os parâmetros, este Driver pode ser executado modificando esta opção para Falso (0, zero)
Initiate Link on Start (Listen first if disabled)	Em uma conexão dual **Cliente/Servidor**, em que ambas as estações são **Clientes** e **Servidores** ao mesmo tempo, é possível definir nesta opção se uma aplicação tenta conectar a um **Servidor** remoto imediatamente após ser executada ou se uma aplicação deve aguardar pela conexão remota para iniciar a operação no lado **Cliente**
Share Connection on Client&Server Mode	Quando em modo **Cliente/Servidor**, deixe esta opção selecionada para permitir que a comunicação bidirecional ocorra sobre a mesma conexão TCP/IP (*socket*). Se a opção **Initiate Link on Start** também está selecionada, então este Driver executa a conexão TCP/IP e inicia o protocolo ICCP como **Cliente**. Caso contrário, recomenda-se configurar este Driver para escutar em uma porta TCP/IP, na aba **Ethernet**, de forma que o parceiro remoto possa se conectar e iniciar como **Cliente** ICCP
Optimize integer encoding on ISO Layers	Informa se este Driver deve compactar números inteiros no menor número de bytes possíveis ou se mantém o tamanho máximo
Msg Timeout (ms)	Tempo de espera para a resposta de uma aplicação a comandos **Select** e **Operate**
Local P Selector	Valor do seletor local usado pela camada *OSI Presentation Layer* (ISO/IEC 8823). O valor padrão desta opção é 1 (um)
Local AE Qualifier	Identificador usado pela camada *OSI Association Layer* como formato. O valor padrão desta opção é 12
Local S Selector	Valor do seletor local usado pela camada *OSI Session Layer* (ISO/IEC 8327). O valor padrão desta opção é 1 (um)
Local App ID	Identificador usado pela camada *OSI Association Layer* (ISO/OSI 8650) no formato **ASN.1** (*Abstract Syntax Notation 1*). Indica o formato de dados adotado pela função **AARQ** (*Association Request*), cujo valor padrão é "1,1,999,1,1" (iso.1.999.1.1)
Local T Selector	Valor do seletor local usado pela camada *OSI Transport Layer* (ISO/IEC 8073). O valor padrão desta opção é 1 (um)
BLT File Path	Caminho de um arquivo de tabela bilateral, com a extensão .blt

Aba ICCP Client

Esta aba contém configurações de parâmetros no lado **Cliente**.



Aba ICCP Client

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

Opções disponíveis na aba ICCP Client

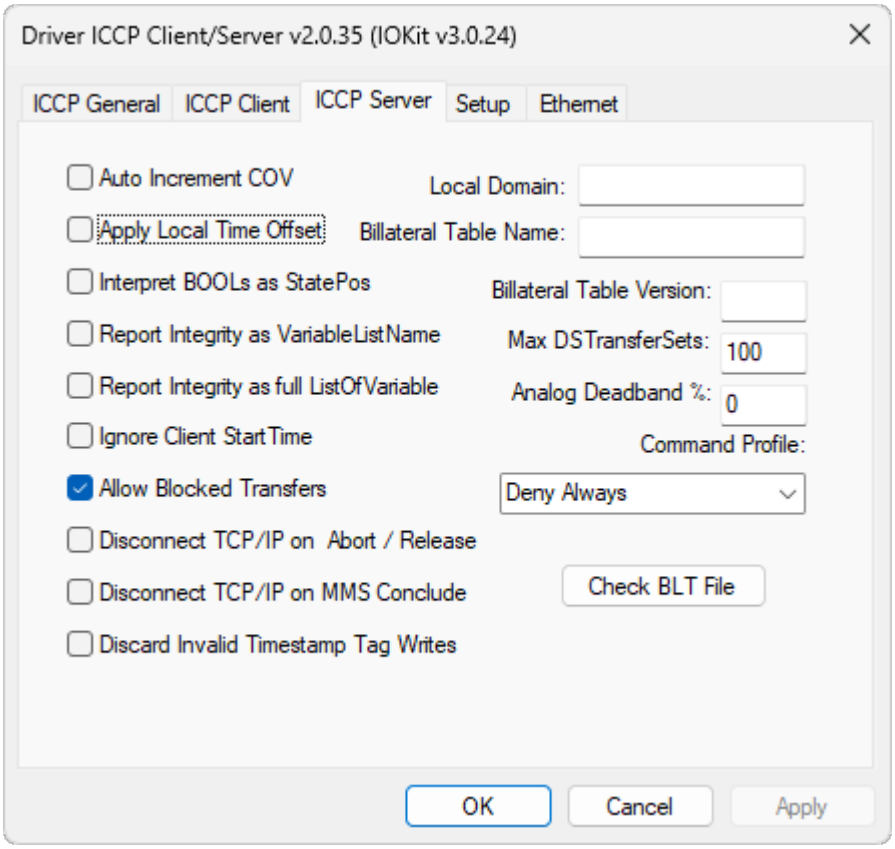
OPÇÃO	DESCRIÇÃO
Remote Domain	Informa o nome do Domínio de um **Servidor** ICCP remoto, ou seja, o nome da instância deste **Servidor**. Este nome é verificado em ambos os lados, **Cliente** e **Servidor**, para diversas operações e deve coincidir exatamente
Expected Billateral Table Name	Informa o nome da Tabela Bilateral de um **Servidor** ICCP. Se no lado do **Servidor** existe uma aplicação **Elipse E3** ou **Elipse Power** usando este Driver, este nome é um *alias* para as definições de um arquivo ICCPServer.blt e deve ser verificado por um **Cliente** para garantir a segurança desta aplicação
Expected Billateral Table Version	Informa a versão da Tabela Bilateral de um **Servidor** ICCP. Esta versão deve ser verificada por um **Cliente** para garantir a segurança desta aplicação
RFC1006 Remote TSAP	Este Driver usa a especificação RFC 1006 como camada de transporte de pacotes ISO sobre TCP. Portanto, é necessário informar o **TSAP** (*Transport Service Access Point*) usado por este Driver no lado **Servidor** para estabelecer a conexão usando este protocolo. O valor padrão desta opção é 1 (um)

OPÇÃO	DESCRIÇÃO
Use Opt Fields at DataSet	A norma IEC 60870-6 versão 2000-08 define campos opcionais que podem ser declarados e enviados junto com uma mensagem **DataSet TransferSet**. Estes campos ajudam a identificar os parâmetros de uma mensagem, como por exemplo a estampa de tempo e o propósito. O valor padrão desta opção é selecionada
Apply Local Time Offset	Considera a estampa de tempo fornecida por um **Servidor** ICCP no formato **UTC** (*Coordinated Universal Time*), aplicando o *offset* do horário local
Discard NTR - Retries	Informa se este Driver deve descartar leituras recebidas de um **Servidor** contendo estampas de tempo nulas (*Null Timestamp Responses*). Se na leitura de um **DataSet** há tipos de dados com estampa de tempo e todas estas estampas de tempo são nulas, então além de descartar as leituras destes pontos, podem ser solicitadas novas leituras, até o valor máximo de retentativas definidas nesta opção
Timestamp lag limit (days)	Informa o número máximo de dias que uma estampa de tempo pode estar atrasada em relação ao horário atual. Se este número de dias é ultrapassado, a estampa de tempo é substituída pelo horário atual e a qualidade de um Tag é configurada para o valor 64 (*Uncertain*)
Timestamp ahead limit (days)	Informa o número máximo de dias que uma estampa de tempo pode estar adiantada em relação ao horário atual. Se este número de dias é ultrapassado, a estampa de tempo é substituída pelo horário atual e a qualidade de um Tag é configurada para o valor 64 (*Uncertain*)
Import XML	Cria um arquivo ICCPClient.blt importando um arquivo XML no formato **Sisco AXS4**
Check BLT File	Verifica a consistência do arquivo ICCPClient.blt e mostra uma caixa de diálogo com os resultados desta verificação
Remote P Selector	Valor do seletor remoto usado pela camada *OSI Presentation Layer* (ISO/IEC 8823). O valor padrão desta opção é 1 (um)
Remote S Selector	Valor do seletor remoto usado pela camada *OSI Session Layer* (ISO/IEC 8327). O valor padrão desta opção é 1 (um)
Remote T Selector	Valor do seletor remoto usado pela camada *OSI Transport Layer* (ISO/IEC 8073). O valor padrão desta opção é 1 (um)
Remote App ID	Identificador usado pela camada *OSI Association Layer* (ISO/OSI 8650) no formato **ASN.1** (*Abstract Syntax Notation 1*). Indica o formato de dados adotado pela função **AARQ** (*Association Request*), cujo valor padrão é "1,1,999,1,1" (iso.1.999.1.1)
Remote AE Qualifier	Identificador usado pela camada *OSI Association Layer* como formato. O valor padrão desta opção é 12
Keep Alive (ms)	Intervalo de tempo usado por um **Cliente** para verificar se a conexão está ativa, através do envio de uma mensagem **MMS Identify**, caso nenhuma outra mensagem tenha sido trocada no período especificado

OPÇÃO	DESCRIÇÃO
Bad Quality Delay (ms)	Atraso a ser considerado para retornar uma qualidade ruim para os pontos após a recepção de um valor com qualidade ruim vindo de um **Servidor**. Se este valor é maior que 0 (zero), então o valor recebido é reportado imediatamente, ainda com qualidade boa. Se, após expirado o tempo informado nesta opção, um ponto ainda está com qualidade ruim, então este ponto é atualizado de forma correspondente. Se um valor com qualidade boa é recebido antes deste intervalo expirar, então este valor é reportado imediatamente e a contagem é interrompida

Aba ICCP Server

Esta aba contém configurações de parâmetros no lado **Servidor**.



Aba ICCP Server

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

Opções disponíveis na aba ICCP Server

OPÇÃO	DESCRIÇÃO
Auto Increment COV	Se esta opção está selecionada, para tipos de dados com a propriedade **COV** (*Counter of Variations*), cada novo valor que ultrapassa a verificação de banda morta força o auto-incremento da propriedade **COV**
Local Domain	Informa o nome do Domínio de um **Servidor** ICCP local, ou seja, o nome da instância. Este nome é verificado em

OPÇÃO	DESCRIÇÃO
	ambos os lados, **Cliente** e **Servidor**, para diversas operações e deve coincidir exatamente
Apply Local Time Offset	Considera a estampa de tempo fornecida por um **Servidor** ICCP no formato **UTC**, aplicando o *offset* do horário local
Billateral Table Name	Informa o nome da Tabela Bilateral de um **Servidor** ICCP. Este nome é um *alias* para as definições de um arquivo ICCPServer.blt e deve ser verificado por um **Cliente** para garantir a segurança de uma aplicação
Interpret BOOLs as StatePos	Os tipos de dados **StateXXX** (**StateQ** e **StateQTimeTag**, entre outros) utilizam 2 (dois) bits para identificar a posição de disjuntores ou chaves (**Aberto**, **Fechado** ou **Inválido**). Com esta opção permite-se o mapeamento de um tipo de dados Booleano, 0 (zero) ou 1 (um), para os estados **Aberto** e **Fechado**. O valor 0 (zero) mapeia para 1 (um, **StateOff**) e o valor 1 (um) mapeia para 2 (dois, **StateOn**)
Billateral Table Version	Informa a versão da Tabela Bilateral de um **Servidor** ICCP. Esta versão deve ser verificada por um **Cliente** para garantir a segurança de uma aplicação
Report Integrity VariableListName	Ao enviar relatórios, esta opção informa se os nomes de variáveis devem ser informados como **VariableListName** se selecionada e como **ListOfVariable** caso contrário. O padrão desta opção em versões anteriores é desmarcada, ou seja, **ListOfVariable**
Max DS TransferSets	Para cada **DataSet** a ser transferido, um **Servidor** deve criar uma instância de um **TransferSet**, por requisição de um **Cliente**, que é associado a este **DataSet**. Esta opção indica o número máximo de **TransferSets** que podem ser criados por este Driver. O valor padrão desta opção é 100
Report Integrity as full ListOfVariable	Ao enviar relatórios como **ListOfVariable**, esta opção indica se deve ser informado o nome de todos os itens. Esta opção torna um relatório menos eficiente por diminuir o espaço para os dados, e deve ser usada apenas por compatibilidade com **Clientes** não corretamente implementados
Analog Deadband %	Informa a banda morta analógica geral aplicada aos dados quando escritos por uma aplicação. Somente os valores que ultrapassem a verificação desta banda morta são copiados para valores atuais considerados como eventos
Ignore Client StartTime	Quando um **Cliente** envia um pedido de **Start DataSet TransferSet** e o intervalo de tempo está errado, utilize esta opção para ignorar o tempo informado, assim um **Servidor** começa o processamento e transmissão dos dados imediatamente
Allow Blocked Transfers	Indica se a resposta da leitura do item reservado **Supported_Features** indica que é suportado o Bloco 3 (*Blocked Transfers*)
Disconnect TCP/IP on Abort/Release	Quando um **Cliente** envia um comando **Abort** da Sessão ou **Release** da Associação sem fechar o canal TCP/IP, deixe

OPÇÃO	DESCRIÇÃO
	esta opção selecionada para que um **Servidor** force a desconexão
Disconnect TCP/IP on MMS Conclude	Quando um **Cliente** envia um comando **Conclude** do MMS sem fechar o canal TCP/IP, deixe esta opção selecionada para que um **Servidor** force a desconexão
Check BLT File	Verifica a consistência do arquivo ICCPServer.blt e mostra uma caixa de diálogo com os resultados desta verificação
Discard Invalid Timestamp Tag Writes	Ao receber uma escrita informando um valor para um Tag, este Driver como **Servidor** utiliza os mesmos parâmetros informados nas opções **Timestamp Lag Limit Days** e **Timestamp Ahead Limit Days** da aba **ICCP Client** para definir se uma estampa de tempo é válida. Se esta opção está selecionada, as escritas com estampa de tempo inválida são descartadas. Caso contrário, este valor é aceito, mas a estampa de tempo é substituída pelo valor atual e a qualidade é configurada com o valor 64 (*Uncertain*)

Tabela Bilateral

As Tabelas Bilaterais ICCPClient.blt e ICCPServer.blt devem seguir algumas regras para processamento por um Driver IEC ICCP Client/Server. Os arquivos devem estar no formato **ASCII**, gerados por **uma das formas descritas no início deste documento** e conter as seções descritas nos tópicos a seguir.

Formato

NOTA
Para declarar uma linha como um comentário, utilize a sequência de caracteres "\\\\" no início desta linha.

```
ICCPVersionYear-ICCPVersionMonth;BLTFileFormatVersion;BLTName
**DataSets**
DataSetCount
DataSetDeclaration (repete-se o número de vezes indicado em DataSetCount)
**DataValues**
DataValueCount
DataValueDeclaration (repete-se o número de vezes indicado em DataValueCount)
**Devices**
DeviceCount
DeviceDeclaration (repete-se o número de vezes indicado em DeviceCount)
**Messages**
MessageCount
MessageDeclaration (repete-se o número de vezes indicado em MessageCount)
```

Opções disponíveis para o formato de Tabela Bilateral

OPÇÃO	DESCRIÇÃO
ICCPVersionYear	Deve ser igual a "2000"
ICCPVersionMonth	Deve ser igual a "08"
BLTFileFormatVersion	Deve ser igual a "2"
BLTName	Nome textual de uma Tabela Bilateral
****DataSets****	Identificador dos **DataSets**

OPÇÃO	DESCRIÇÃO
DataSetCount	Número total de **DataSets** que seguem
DataSetDeclaration	Série de parâmetros separados por ponto e vírgula. Consulte a tabela **Parâmetros da opção DataSetDeclaration** para mais informações
****DataValues****	Identificador de **DataValues**
DataValueCount	Número total de **DataValues** que seguem
DataValueDeclaration	Série de parâmetros separados por ponto e vírgula. Consulte a tabela **Parâmetros da opção DataValueDeclaration** para mais informações
****Devices****	Identificador de equipamentos
DeviceCount	Número total de equipamentos que seguem
DeviceDeclaration	Série de parâmetros separados por ponto e vírgula. Consulte a tabela **Parâmetros da opção DeviceDeclaration** para mais informações
****Messages****	Identificador de mensagens
MessageCount	Número total de mensagens que seguem
MessageDeclaration	Série de parâmetros separados por ponto e vírgula. Os valores disponíveis são **InfoReference**: Nome do *slot* de uma mensagem, **Size**: Tamanho máximo e **Description**: Descrição opcional

Parâmetros da opção DataSetDeclaration

PARÂMETRO	DESCRIÇÃO
DataSetName	Nome único de um **DataSet**
Delay	Atraso, em segundos, para que um **Servidor** inicie o monitoramento de pontos de um **DataSet** depois de iniciar a execução
Interval	Intervalo de tempo entre relatórios de um **Servidor**, em segundos, usado pela opção **IntervalTimeout** no parâmetro **DSConditionsRequested**
TLE	Limite de tempo para execução, em segundos
BufferTime	Intervalo de tempo para manter em memória os eventos de mudanças em objetos antes de enviar um relatório, em segundos
IntegrityCheck	Intervalo de tempo para verificação de integridade quando a condição **IntegrityTimeout** é usada no parâmetro **DSConditionsRequested**, em segundos. Este parâmetro só é significativo quando o parâmetro **RBE** é igual a 1 (um)
DSConditionsRequested	Um byte cujos 5 (cinco) bits mais significativos representam as instruções de monitoramento para um **Servidor** ICCP. Os valores dos bits são **7**: **IntervalTimeout** (envio periódico), **6**: **IntegrityTimeout** (válido somente se a opção **RBE** está selecionada), **5**: **ObjectChange**, **4**:

PARÂMETRO	DESCRIÇÃO
	OperatorRequest, **3**: **OtherExternalEvent**, **2**: Não usado, **1**: Não usado e **0**: Não usado
BlockData	Os valores possíveis são **1**: Habilita a compressão de uma mensagem ou **0**: Desabilita a compressão de uma mensagem
Critical	Os valores possíveis são **1**: Solicita a confirmação de **Clientes** ICCP para cada relatório ou **0**: Não requer a confirmação de **Clientes** ICCP
RBE	Os valores possíveis são **1**: Habilita um relatório pelo mecanismo de exceções ou **0**: Desabilita um relatório pelo mecanismo de exceções
ReportAllException	Os valores possíveis são **1**: Informa a um **Servidor** para reportar todas as mudanças nos pontos monitorados ou **0**: Informa a um **Servidor** para reportar apenas o valor do último ponto monitorado no momento da geração deste relatório

Parâmetros da opção DataValueDeclaration

PARÂMETRO	DESCRIÇÃO
DataSetIndex	Índice de um **DataSet** na seção **DataSetDeclaration** que reporta este **DataValue**, iniciando em 1 (um), ou configure em 0 (zero) se não há **DataValues** para reportar
TagName	Nome único de um **DataValue**
DataType	Tipo de dados ICCP pré-definido. Os valores possíveis são **Discrete**, **DiscreteQ**, **DiscreteQTimeTag**, **DiscreteQTimeTagExtended**, **DiscreteExtended**, **Real**, **RealQ**, **RealQTimeTag**, **RealQTimeTagExtended**, **RealExtended**, **State**, **StateQ**, **StateQTimeTag**, **StateQTimeTagExtended** ou **StateExtended**
Scope	Os valores possíveis são **1**: Local (ICC) ou **0**: Global (VCC)
SourceType	Os valores possíveis são **0**: Telemetria, **1**: Calculado, **2**: Digitado ou **3**: Estimado
AllowWrite	Os valores possíveis são **1**: Sim ou **0**: Não
Description	Descrição opcional

Parâmetros da opção DeviceDeclaration

PARÂMETRO	DESCRIÇÃO
DataSetIndex	Índice de um **DataSet**
Name	Nome de um equipamento
CmdType	Os valores possíveis são **C**: CMD, **D**: SP_DISCRETE ou **R**: SP_REAL
Scope	Os valores possíveis são **1**: Local (ICC) ou **0**: Global (VCC)

PARÂMETRO	DESCRIÇÃO
TAGGABLE	Os valores possíveis são **1**: Sim ou **0**: Não
SBO	Os valores possíveis são **1**: Sim ou **0**: Não
CheckBackID	Número retornado por um **Servidor** quando um equipamento é selecionado e somente é utilizado quando o parâmetro **SBO** é igual a 1 (um)
Description	Descrição opcional

Exemplo de Arquivo

```
2000-08;2;Example
\\\\ DataSet Declaration
\\\\ **Identifier**
\\\\ Number of Items
\\\\ DataSetName, Delay, Interval,
\\\\ TLE, BufferTime, IntegrityCheck,
\\\\ DSConditionsRequested, BlockData, Critical,
\\\\ RBE, ReportAllException
**DataSets**
1
DS1;0;0;0;0;0;0;1;1;1;1
\\\\ DataValue Declaration
\\\\ **Identifier**
\\\\ Number of Items
\\\\ DataSetIndex, TagName, DataType,
\\\\ Scope:ICC=1 VCC=0,
\\\\ SourceType:TEL=0 CAL=1 ENT=2 EST=3,
\\\\ AllowWrite:YES=1 NO=0, Description
**DataValues**
10
1;CMMXU1$MX$A$phsA$cVal$mag$f;RealQTimeTagExtended;0;0;0;
1;CMMXU1$MX$A$phsA$range;RealQTimeTagExtended;0;0;0;
1;CMMXU1$MX$A$phsB$cVal$mag$f;RealQTimeTagExtended;0;0;0;
1;CMMXU1$MX$A$phsB$range;RealQTimeTagExtended;0;0;0;
1;CMMXU1$MX$A$phsC$cVal$mag$f;RealQTimeTagExtended;0;0;0;
1;CMMXU1$MX$A$phsC$range;RealQTimeTagExtended;0;0;0;
1;CVMMXU1$MX$Amp$mag$f;RealQTimeTagExtended;0;0;0;
1;CVMMXU1$MX$Amp$range;RealQTimeTagExtended;0;0;0;
1;CVMMXU1$MX$Hz$mag$f;RealQTimeTagExtended;0;0;0;
1;CVMMXU1$MX$Hz$range;RealQTimeTagExtended;0;0;0;
\\\\ Command Declaration
\\\\ **Identifier**
\\\\ Number of Items
\\\\ DataSetIndex, Name,
\\\\ CmdType: NONE=N CMD=C SP_DISCRETE=D SP_REAL=R,
\\\\ Scope:ICC=1 VCC=0, TAGGABLE: YES=1 NO=0,
\\\\ SBO: YES=1 NO=0, CheckBackID, Description
**Devices**
2
0;CMMXU1$CO;C;0;0;0;0;
0;CMSQI1$CO;C;0;0;0;0;
\\\\ Message Declaration
\\\\ **Identifier**
\\\\ Number of Items
\\\\ InfoReference, Size, Description
**Messages**
0
```

Notas Importantes

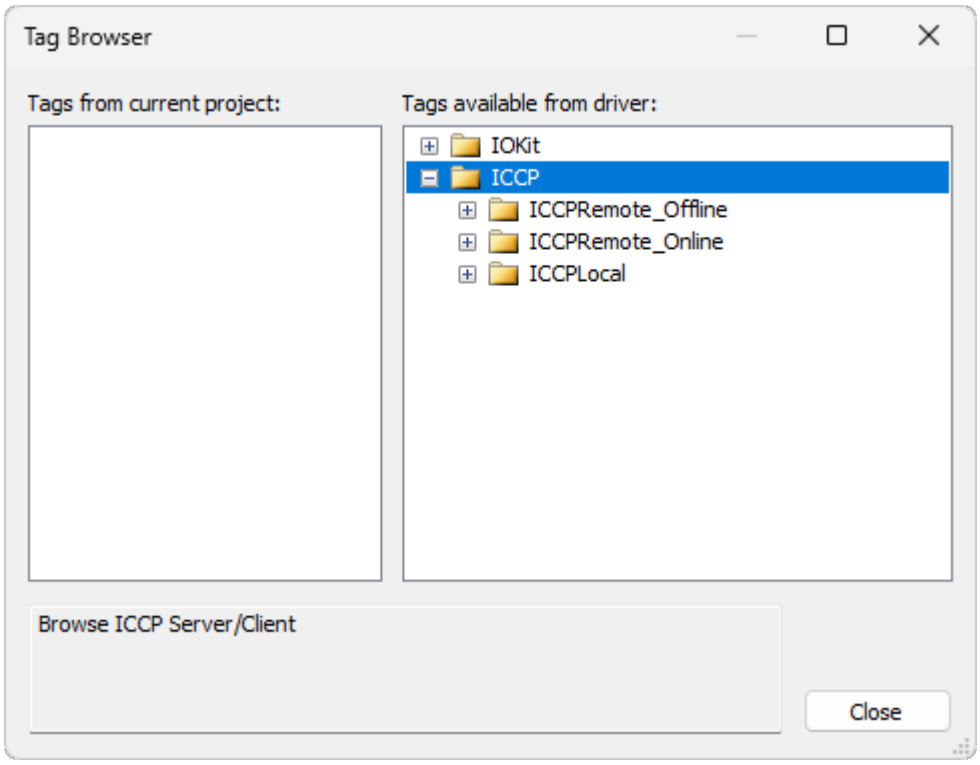
- Um **Servidor** ICCP usa um tamanho de **MaxPDU** (*Maximum Number of Bytes in a Single MMS Data Unit*) igual a 32000 bytes. É responsabilidade de um **Cliente** restringir o tamanho de cada **DataSet** para caber em um único PDU. Se este tamanho é excedido, um novo **DataSet** deve ser criado.

- Um **Servidor** ICCP somente verifica a quantidade total de mensagens com eventos, ou **ReportAllChanges**, em modo **Block**. **Clientes** devem preferencialmente usar este modo **Block** para evitar mensagens com tamanho maior do que o esperado por um **Cliente**.

- Cada Tag deve aparecer em um único **DataSet**, mas vários **DataSets** são permitidos.

Tag Browser

Clique em **Tag Browser** na barra de ferramentas de um Driver no **Elipse E3** ou **Elipse Power** para abrir a janela Tag Browser, mostrada na figura a seguir. Esta janela permite arrastar e soltar Tags encontrados nas configurações de **Clientes** e **Servidores**.



Janela Tag Browser

A listagem **Tags available from driver** (*Tags disponibilizados pelo driver*) contém os itens descritos a seguir.

- **IOKit**: Contém os Tags padrão da biblioteca **IOKit** para leitura e escrita de parâmetros gerais e status, entre outros. Estes Tags **NÃO PODEM** ser usados por este Driver porque a conexão padrão de um *host* não é usada. Este Driver usa 2 (duas) instâncias internas de *hosts*, uma para **Cliente** e outra para **Servidor**

- **ICCP**: Contém os itens **ICCPRemote_Offline**, que mostra Tags lidos de um arquivo ICCPClient.blt representando um **Servidor** ICCP remoto, **ICCPRemote_Online**, que lê declarações de pontos de um **Servidor** ICCP remoto, cria um arquivo ICCPClient.blt e mostra estes Tags (uma conexão TCP/IP corretamente configurada deve estar disponível para que estes pontos sejam lidos) e **ICCPLocal_Offline**, que mostra Tags lidos de um arquivo ICCPServer.blt representando um **Servidor** ICCP local

Referência de Tags

Este Driver somente utiliza os parâmetros *Item* e *N1* dos Tags. Se um Tag referencia o lado **Servidor**, o parâmetro *N1* deve ser igual a 0 (zero). Se um Tag referencia o lado **Cliente**, o parâmetro *N1* deve ser igual a 1 (um).

Tags do Lado de um Servidor

A tabela a seguir contém os Tags do lado **Servidor** de um Driver IEC ICCP Client/Server.

Tags disponíveis no lado Servidor

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
PointName	Somente escrita	0 (zero)	Recebe o valor, a estampa de tempo e a qualidade atual de um ponto em uma aplicação. Este Tag deve ser declarado no arquivo ICCPServer.blt
PointName:Source	Somente escrita	0 (zero)	Muda a fonte padrão de um Ponto. Os valores possíveis são **0**: Telemetria, **1**: Calculado, **2**: Digitado ou **3**: Estimado
DeviceName:Select	Leitura ou escrita	0 (zero)	Indica se um equipamento está selecionado no lado **Cliente** utilizando o valor 1 (um). Uma aplicação deve escrever de volta este valor 1 (um) para confirmar a aceitação da operação selecionada ou 0 (zero) para não confirmar esta operação, que deve ser completada no tempo máximo informado na opção **Msg Timeout (ms)** da aba **ICCP General**
DeviceName:Operate	Leitura ou escrita	0 (zero)	Indica se um equipamento está em operação no lado **Cliente**. O valor deste Tag é enviado pelo lado **Cliente**. Uma aplicação deve enviar o mesmo valor de volta para confirmar o sucesso desta operação ou qualquer outro valor para não confirmar esta operação, que deve ser completada no tempo máximo informado na opção **Msg Timeout (ms)** da aba **ICCP General**
DeviceName:Tag	Leitura ou escrita	0 (zero)	Um Tag Bloco com 2 (dois) Elementos contendo as marcações de um equipamento. O Elemento 1 (um) indica o estado de um equipamento e é uma máscara de bits, em que o bit 0 (zero) pode ser **1**: Armed ou **0**: Not Armed e os bits 1 (um) e 2 (dois) podem ser **0**: No Tag, **1**: CloseOnlyInhibit ou **2**: OpenAndCloseInhibit. O

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
			Elemento 2 (dois) indica o motivo para as marcações, no formato **Texto**. **NOTA**: Se um equipamento é marcado com o estado **Armed**, este equipamento é colocado novamente no estado **Idle** após o *time-out* de um comando **Select**
ServerStatus	Somente leitura	0 (zero)	Retorna o status interno no lado de um **Servidor** ICCP. Os valores possíveis são **0**: Iniciando, **1**: Lendo arquivos .ld, **2**: Recuperando o diretório de um **Servidor** (**GetServerDirectory**) ou **3**: Processando um arquivo .ld (*Tag Browsing*)
ServerConnection	Somente leitura	0 (zero)	Indica o status de uma conexão. Os valores possíveis são **1**: Conectado ou **0**: Desconectado
UpdatePointDatabase	Somente escrita	0 (zero)	Insere ou atualiza um ponto no banco de dados de um **Servidor**. Recebe um Tag Bloco com 2 (dois), 3 (três) ou 4 (quatro) Elementos. Consulte a tabela **Elementos do Tag UpdatePointDatabase** para mais informações
UpdateDeviceDatabase	Somente escrita	0 (zero)	Insere um equipamento no banco de dados de um **Servidor**. Deve receber um Tag Bloco contendo entre 2 (dois) e 7 (sete) Elementos. Consulte a tabela **Elementos do Tag UpdateDeviceDatabase** para mais informações
DeleteDatabase	Somente escrita	0 (zero)	Apaga todas as definições de um arquivo de Tabela Bilateral (BLT). Deve receber uma **String** com o nome de um arquivo BLT. O nome e o caminho deste arquivo não são modificados. Deve-se escrever neste Tag somente quando este Driver inicia em modo **Offline**, permitindo apagar o banco de dados atual, redefinir o conteúdo e,

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
			por fim, utilizar o Tag **SaveDatabase**
SaveDatabase	Somente escrita	0 (zero)	Salva um arquivo de Tabela Bilateral (BLT). Deve receber uma **String** com o nome de um arquivo BLT. O nome e o caminho deste arquivo não são modificados
UpdatePoint	Somente escrita	0 (zero)	Atualiza um ponto na memória. Deve receber um Tag Bloco com 2 (dois) ou 3 (três) Elementos. Consulte a tabela **Elementos de um Tag UpdatePoint** para mais informações. **NOTA**: Este Tag representa a mesma operação de escrita de um Tag **PointName**, exceto que permite informações de uma fonte na mesma operação
SelectConfirm	Somente escrita	0 (zero)	Recebe a confirmação de uma aplicação de que um equipamento foi selecionado. Deve receber um Tag Bloco com 2 (dois) Elementos. Os valores para os Elementos são **1**: Nome de um equipamento e **2**: 1 (um, valor fixo). **NOTA**: Este Tag representa a mesma operação de escrita de um Tag **DeviceName:Select**
OperateConfirm	Somente escrita	0 (zero)	Recebe a confirmação de uma aplicação de que um equipamento foi operado. Deve receber um Tag Bloco com 2 (dois) Elementos. Os valores para os Elementos são **1**: Nome de um equipamento e **2**: O mesmo valor recebido para confirmar ou um valor diferente para não confirmar. **NOTA**: Este Tag representa a mesma operação de escrita de um Tag **DeviceName:Operate**
Tagging	Somente escrita	0 (zero)	Insere uma marcação em um equipamento, semelhante à operação de escrita de um Tag **DeviceName:Tag**. Deve receber um Tag Bloco com 3 (três) Elementos. Consulte a

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
			tabela **Elementos de um Tag Tagging** para mais informações. **NOTA**: Se um equipamento é marcado com o estado **Armed**, este equipamento é colocado novamente no estado **Idle**

Elementos de um Tag UpdatePointDatabase

ELEMENTO	DESCRIÇÃO
1	Nome de um ponto
2	Tipo de um ponto. Os valores possíveis são **0**: Discrete, **1**: DiscreteQ, **2**: DiscreteQTimeTag, **3**: DiscreteQTimeTagExtended, **4**: DiscreteExtended, **5**: Real, **6**: RealQ, **7**: RealQTimeTag, **8**: RealQTimeTagExtended, **9**: RealExtended, **10**: State, **11**: StateQ, **12**: StateQTimeTag, **13**: StateQTimeTagExtended, **14**: StateExtended ou **15**: Tag_Value
3	Escopo. Os valores possíveis são **0**: VCC ou **1**: ICC
4	Fonte. Os valores possíveis são **0**: Telemetria (padrão), **1**: Calculado, **2**: Digitado ou **3**: Estimado

Elementos de um Tag UpdateDeviceDatabase

ELEMENTO	DESCRIÇÃO
1	Uma **String** com o nome de um equipamento
2	Uma **String** com o tipo de um equipamento. Os valores possíveis são **N**: None, **C**: CMD, **D**: SP_DISCRETE ou **R**: SP_REAL
3	Escopo. Um Elemento opcional que pode conter os valores **0**: VCC ou **1**: ICC
4	Marcação. Um Elemento opcional que pode conter os valores **0**: Não ou **1**: Sim
5	SBO. Um Elemento opcional que pode conter os valores **0**: Não ou **1**: Sim
6	*CheckBackID*. Elemento opcional
7	Descrição. Uma **String** opcional

Elementos de um Tag UpdatePoint

ELEMENTO	DESCRIÇÃO
1	Nome de um ponto
2	Novo valor

ELEMENTO	DESCRIÇÃO
3	Fonte. Os valores possíveis são **0**: Telemetria (padrão), **1**: Calculado, **2**: Digitado ou **3**: Estimado

Elementos de um Tag Tagging

ELEMENTO	DESCRIÇÃO
1	Nome de um equipamento
2	Estado de um equipamento como uma máscara de bits cujos bits 0 (zero) e 1 (um) indicam um Tag (**No Tag**, **OpenCloseInhibit** ou **CloseOnlyInhibit**) e o bit 2 (dois) indica o estado, **Armed** ou **Idle**
3	Motivo para esta marcação no formato **Texto**

Tags do Lado de um Cliente

A tabela a seguir contém os Tags do lado **Cliente** de um Driver IEC ICCP Client/Server.

Tags disponíveis no lado Cliente

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
PointName	Leitura ou escrita	1 (um)	Lê ou escreve o valor atual, a estampa de tempo e a qualidade de um ponto. Este Tag deve estar declarado no arquivo ICCPClient.blt
PointName:COV	Somente leitura	1 (um)	Lê a propriedade **COV** (*Counter of Variations*) de um ponto, se disponível
PointName:Source	Somente leitura	1 (um)	Lê a fonte atual de um ponto. Os valores possíveis são **0**: Telemetria, **1**: Calculado, **2**: Digitado ou **3**: Estimado
GetDataValue:PointName	Somente leitura	1 (um)	Operação que serve para validar se um determinado Tag existe na base de um **Servidor**, retornando este valor. Não deve ser usado como forma principal de leitura, pois está baseada em *polling* e pode perder eventos entre cada leitura
DeviceName	Somente leitura	1 (um)	Lê o estado de um equipamento. Uma máscara de bits em que o bit 0 (zero) pode ser **1**: Armed ou **0**: Not Armed e os bits 1 (um) e 2 (dois) podem ser **0**: No Tag, **1**: CloseOnlyInhibit ou **2**: OpenAndCloseInhibit

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
DeviceName:Select	Somente escrita	1 (um)	Requisita um comando **Select** em um equipamento. Esta requisição pode falhar se um ponto não é do tipo **SBO** ou se alguma das verificações de segurança realizadas por um **Servidor** ICCP falha. Se um equipamento está declarado em um **DataSet**, o Tag **DeviceName** mostra o bit 0 (zero) configurado, indicando que um equipamento está no estado **Armed** se esta operação é bem-sucedida. Este Tag pode conter qualquer valor
DeviceName:Operate	Somente escrita	1 (um)	Requisita um comando **Operate** em um equipamento. Esta requisição pode falhar se um ponto é do tipo **SBO** e não está no estado **Armed** ou se alguma das verificações de segurança realizadas por um **Servidor** ICCP falha. Este Tag deve conter o valor a ser enviado a um **Servidor** ICCP. A norma IEC 60870-6-802 define os valores 0 (zero) e 1 (um) para operações com comandos digitais, em que **0**: Trip, Open, Off ou Lower (para Transformadores) e **1**: Close, On ou Raise (para Transformadores)
DeviceName:Tag	Leitura ou escrita	1 (um)	Um Tag Bloco com 2 (dois) Elementos contendo os Tags de um equipamento. O Elemento 1 (um) indica o estado de um equipamento como uma máscara de bits em que o bit 0 (zero) pode ser **1**: Armed ou **0**: Not Armed e os bits 1 (um) e 2 (dois) podem ser **0**: No Tag, **1**: CloseOnlyInhibit ou **2**: OpenAndCloseInhibit. O Elemento 2 (dois) indica o motivo para as marcações, no formato **Texto**
DataSetName:GI	Somente escrita	1 (um)	Força uma requisição de integridade do **DataSet**

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
			informado, com o parâmetro **DSConditions** configurado para **OperatorRequest**
ClientStatus	Somente leitura	1 (um)	Retorna o estado interno do lado **Cliente** ICCP. Os valores possíveis são **0**: Iniciando, **1**: Lendo arquivos .ld, **2**: Recuperando o diretório (**GetServerDirectory**) ou **3**: Processando um arquivo .ld (*Tag Browsing*)
ClientConnection	Somente leitura	1 (um)	Indica o status de uma conexão. Os valores possíveis são **1**: Conectado ou **0**: Não conectado
StartBatchDatabaseUpdate	Somente escrita	1 (um)	Inicia um processo de atualização em lote para modificar a Tabela Bilateral de um **Cliente**. Este passo é obrigatório antes de utilizar outros Tags de edição em tempo de execução, tais como **UpdateDatasetDatabase**, **UpdatePointDatabase** ou **UpdateDeviceDatabase**
UpdateDatasetDatabase	Somente escrita	1 (um)	Insere ou atualiza um **DataSet** no banco de dados de um **Cliente**. Deve receber um Tag Bloco com 11 Elementos que seguem a mesma ordem descrita na seção **Dataset** do arquivo da Tabela Bilateral. Os valores para os Elementos são **1**: DataSetName, **2**: Delay, **3**: Interval, **4**: TLE, **5**: BufferTime, **6**: IntegrityCheck, **7**: DSConditionsRequested, **8**: BlockData, **9**: Critical, **10**: RBE e **11**: ReportAllException
UpdatePointDatabase	Somente escrita	1 (um)	Insere ou atualiza um ponto no banco de dados de um **Cliente**. Deve receber um Tag Bloco com 2 (dois), 3 (três), 4 (quatro) ou 5 (cinco) Elementos. Consulte a tabela **Elementos do Tag UpdatePointDatabase** para mais informações

TAG	OPERAÇÃO	PARÂMETRO N1	DESCRIÇÃO
UpdateDeviceDatabase	Somente escrita	1 (um)	Insere um equipamento no banco de dados de um **Cliente**. Deve receber um Tag Bloco com no mínimo 2 (dois) e no máximo 7 (sete) Elementos. Consulte a tabela **Elementos do Tag UdpateDeviceDatabase** para mais informações
DeleteDatabase	Somente escrita	1 (um)	Apaga todas as definições de um arquivo de Tabela Bilateral (BLT). Deve receber uma **String** com o nome de um arquivo BLT. O nome e o caminho deste arquivo não são modificados. Este Tag deve ser escrito somente quando este Driver inicia em modo **Offline**, permitindo apagar o banco de dados atual, redefinir o conteúdo e, por fim, utilizar o Tag **SaveDatabase**
SaveDatabase	Somente escrita	1 (um)	Salva um arquivo de Tabela Bilateral (BLT). Deve receber uma **String** com o nome de um arquivo BLT. O nome e o caminho deste arquivo não são modificados

Elementos de um Tag UpdatePointDatabase

ELEMENTO	DESCRIÇÃO
1	Nome de um ponto
2	Tipo de um ponto. Os valores possíveis são **0**: Discrete, **1**: DiscreteQ, **2**: DiscreteQTimeTag, **3**: DiscreteQTimeTagExtended, **4**: DiscreteExtended, **5**: Real, **6**: RealQ, **7**: RealQTimeTag, **8**: RealQTimeTagExtended, 9: RealExtended, **10**: State, **11**: StateQ, **12**: StateQTimeTag, **13**: StateQTimeTagExtended, **14**: StateExtended ou **15**: Tag_Value
3	Escopo. Os valores possíveis são **0**: VCC ou **1**: ICC (padrão)
4	Fonte. Os valores possíveis são **0**: Telemetria (padrão), **1**: Calculado, **2**: Digitado ou **3**: Estimado
5	Índice de um **DataSet** ou 0 (zero) se um ponto não está associado a um **DataSet**

Elementos de um Tag UpdateDeviceDatabase

ELEMENTO	DESCRIÇÃO
1	Uma **String** com o nome de um equipamento
2	Uma **String** com o tipo de um equipamento. Os valores possíveis são **N**: None, **C**: CMD, **D**: SP_DISCRETE ou **R**: SP_REAL
3	Escopo. Um Elemento opcional que pode conter os valores **0**: VCC ou **1**: ICC
4	Marcação. Um Elemento opcional que pode conter os valores **0**: Não ou **1**: Sim
5	SBO. Um Elemento opcional que pode conter os valores **0**: Não ou **1**: Sim
6	*CheckBackID*. Elemento opcional
7	Descrição. Uma **String** opcional

Configuração em Tempo de Execução

Este tópico contém informações sobre como alterar a configuração deste Driver e dos membros de Tabelas Bilaterais em tempo de execução.

Parâmetros deste Driver

Se a opção **Start Offline (Allows Runtime Configuration)** na aba **General** está selecionada, pode-se modificar os parâmetros deste Driver em tempo de execução. Cada opção nesta aba tem um identificador correspondente que pode ser usado para informar um novo valor, utilizando o Tag especial **Write Parameter** da biblioteca **IOKit**, definido pelo 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), e que deve receber um vetor com o identificador e o respectivo valor.

Propriedades e identificadores

PROPRIEDADE	IDENTIFICADOR
ICCP_LOCAL_PSEL	ICCP.LocalPSel
ICCP_LOCAL_SSEL	ICCP.LocalSSel
ICCP_LOCAL_TSEL	ICCP.LocalTSel
ICCP_LOCAL_AE_QUALIFIER	ICCP.LocalAEQualifier
ICCP_LOCAL_APP_ID	ICCP.LocalAppID
ICCP_LOCAL_TSAP	ICCP.LocalTSAP
ICCP_REM_PSEL	ICCP.RemPSel
ICCP_REM_SSEL	ICCP.RemSSel
ICCP_REM_TSEL	ICCP.RemTSel
ICCP_REM_AE_QUALIFIER	ICCP.RemAEQualifier
ICCP_REM_APP_ID	ICCP.RemAppID
ICCP_REM_TSAP	ICCP.RemTSAP
ICCP_CBB	ICCP.CBB

PROPRIEDADE	IDENTIFICADOR
ICCP_CLIENT_SERVICES	ICCP.ClientServices
ICCP_SERVER_SERVICES	ICCP.ServerServices
ICCP_TIMEOUT	ICCP.Timeout
ICCP_BUFFER_TIME	ICCP.BufferTime
ICCP_INTEGRITY	ICCP.Integrity
ICCP_RPT_INTERVAL	ICCP.RptInterval
ICCP_RBE	ICCP.RBE
ICCP_REBUILDDS	ICCP.RebuildDS
ICCP_DISCARDDS	ICCP.DiscardDS
ICCP_PROFILE	ICCP.Profile
ICCP_REMOTE_DOMAIN	ICCP.RemoteDomain
ICCP_LOCAL_DOMAIN	ICCP.LocalDomain
ICCP_BLTFILEPATH	ICCP.BltFilePath
ICCP_AUTO_INCR_COV	ICCP.AutoIncrCOV
ICCP_MAX_DSTS	ICCP.MaxDSTS
ICCP_SELECT_TOUT	ICCP.SelectTout
ICCP_BLTNAME	ICCP.BLTName
ICCP_BLTVERSION	ICCP.BLTVersion
ICCP_USEOPTFLDS	ICCP.UseOptFlds
ICCP_INITLINK	ICCP.InitLink
ICCP_EXP_BLTNAME	ICCP.ExpBLTName
ICCP_EXP_BLTVERSION	ICCP.ExpBLTVersion
ICCP_ANALOG_DEADBAND	ICCP.AnalogDeadBand
ICCP_START_OFFLINE	ICCP.StartOffline
ICCP_CLIENT_OFFLINE	ICCP.ClientOffline
ICCP_SERVER_OFFLINE	ICCP.ServerOffline

O código a seguir contém um exemplo de configuração deste Driver.

```
Write -1, 0, 0, 3, Array("ICCP.ClientOffline", 1)
```

Parâmetros de um Servidor

Pode-se modificar Tabelas Bilaterais de **Servidores** ICCP em tempo de execução para adicionar ou modificar Tags. Entretanto, caso seja necessário remover Tags, um **Servidor** deve ser colocado em modo **Offline**.

Adicionar ou Modificar Tags

Utilize chamadas aos Tags **UpdatePointDatabase** e **UpdateDeviceDatabase**. Se estes Tags já existem, os parâmetros são atualizados. Se estes Tags não existem, um novo ponto é criado. Ao final deste processo, execute uma chamada ao Tag **SaveDatabase**.

Remover, Adicionar ou Modificar Tags

Se uma operação de remoção é necessária, uma sequência de passos deve ser realizada, conforme as opções descritas a seguir.

Opção 1: Este Driver Inicia em Modo Offline

1. Selecione a opção **Start Offline (Allows Runtime Configuration)**.
2. Coloque o **Servidor** ICCP em modo **Offline** escrevendo o valor 1 (um) no Tag **ICCP.ServerOffline**, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.ServerOffline", 1)
```

3. Configure o Tag **ICCP.StartOffline** com o valor 0 (zero). Este Driver cria um componente **Servidor**, pois agora está em modo **Online**. Entretanto, este componente **Servidor** ainda está em modo **Offline**, devido ao passo anterior, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.StartOffline", 0)
```

4. Execute chamadas aos Tags **UpdatePointDatabase** e **UpdateDeviceDatabase**. Como o banco de dados está vazio, todos os itens devem ser inseridos.
5. Execute uma chamada ao Tag **SaveDatabase**.
6. Coloque o **Servidor** ICCP de volta ao modo **Online** escrevendo o valor 0 (zero) no Tag **ICCP.ServerOffline**, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.ServerOffline", 0)
```

Opção 2: Este Driver Inicia em Modo Online

1. Desmarque a opção **Start Offline (Allows Runtime Configuration)**. Esta configuração desconecta todos os canais de comunicação, tanto **Clientes** quanto **Servidores**, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.StartOffline", 1)
```

2. Repita os passos seguintes do procedimento de **iniciar em modo Offline**.

Parâmetros de um Cliente

Pode-se modificar Tabelas Bilaterais de **Clientes** ICCP, em tempo de execução, para adicionar ou modificar Tags, por exemplo, de forma a evitar que este Driver seja parado e reiniciado para aplicar as modificações.

No lado **Cliente**, esta operação pode ser realizada enquanto um **Cliente** está em modo **Offline** ou durante a comunicação normal.

Durante esta comunicação, pode-se modificar parâmetros dos **DataSets**, adicionar, modificar e remover **DataPoints** e equipamentos. Em modo **Offline**, pode-se criar uma Tabela Bilateral totalmente nova.

Mudanças em Modo Offline

Esta operação pode ser realizada ao iniciar este Driver, configurando a opção **Start Offline (Allows Runtime Configuration)**. Neste caso, execute os passos descritos a seguir.

1. Selecione a opção **Start Offline (Allows Runtime Configuration)**.

2. Coloque o **Cliente** ICCP em modo **Offline** escrevendo o valor 1 (um) no Tag **ICCP.ClientOffline**, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.ClientOffline", 1)
```

3. Configure o Tag **ICCP.StartOffline** com o valor 0 (zero). Este Driver cria um componente **Cliente**, pois agora está em modo **Online**. Entretanto, este componente **Cliente** ainda está em modo **Offline**, devido ao passo anterior, conforme o exemplo a seguir.

```
Write -1, 0, 0, 3, Array("ICCP.StartOffline", 0)
```

4. Escreva no Tag **StartBatchDatabaseUpdate**. No exemplo a seguir, este Tag foi criado dinamicamente antes da chamada.

```
Set StartBatch = AddObject("IOTag", False, "StartBatch")
StartBatch.ParamItem = "StartBatchDatabaseUpdate"
StartBatch.N1 = 1
StartBatch.Activate
StartBatch.WriteEx 1
```

5. Adicione **DataSets**, **DataPoints** e equipamentos usando os Tags **UpdateDatasetDatabase**, **UpdatePointDatabase** e **UpdateDeviceDatabase**. Deve ser realizada uma chamada para cada componente a ser criado, conforme o exemplo a seguir.

```
'Neste caso, há uma pasta chamada SYS
'com um Tag Bloco de cinco Elementos
'chamado UpdatePoint.
Set UpdtPoint = Item("SYS").Item("UpdatePoint")
UpdtPoint.WriteEx Array("TAGNAME_AAA",13,0,0,1)
UpdtPoint.WriteEx Array("TAGNAME_BBB",13,0,0,1)
UpdtPoint.WriteEx Array("TAGNAME_CCC",13,0,0,1)
```

6. Execute uma chamada ao Tag **SaveDatabase**.

```
Set SaveDB = Item("SYS").Item("SaveDatabase")
SaveDB.WriteEx "BLT_Name"
```

7. Coloque o **Cliente** novamente em modo **Online**.

```
Write -1, 0, 0, 3, Array("ICCP.ClientOffline", 0)
```

Mudanças em Tempo de Execução

1. Escreva no Tag **StartBatchDatabaseUpdate**. No exemplo a seguir, este Tag foi criado dinamicamente antes da chamada.

```
Set StartBatch = AddObject("IOTag", False, "StartBatch")
StartBatch.ParamItem = "StartBatchDatabaseUpdate"
StartBatch.N1 = 1
StartBatch.Activate
StartBatch.WriteEx 1
```

2. Informe os **DataSets**, **DataPoints** e equipamentos utilizando os Tags **UpdateDatasetDatabase**, **UpdatePointDatabase** e **UpdateDeviceDatabase**. Deve ser realizada uma chamada para cada componente a ser criado, conforme o exemplo a seguir.

```
'Neste caso, há uma pasta chamada SYS
'com um Tag Bloco de cinco Elementos
'chamado UpdatePoint.
Set UpdtPoint = Item("SYS").Item("UpdatePoint")
UpdtPoint.WriteEx Array("TAGNAME_AAA",13,0,0,1)
UpdtPoint.WriteEx Array("TAGNAME_BBB",13,0,0,1)
UpdtPoint.WriteEx Array("TAGNAME_CCC",13,0,0,1)
```

3. Execute uma chamada ao Tag **SaveDatabase**.

```
Set SaveDB = Item("SYS").Item("SaveDatabase")
SaveDB.WriteEx "BLT_Name"
```

4. Ao executar a chamada ao Tag **SaveDatabase**, a Tabela Bilateral atual é comparada com a nova Tabela, com as seguintes regras:

- Se uma coincidência é encontrada em ambas as tabelas, a nova tabela é copiada sobre a existente
- Se uma entrada é encontrada na tabela existente e não na nova tabela, a entrada existente é apagada
- Se uma entrada é encontrada na tabela nova e não na tabela existente, a entrada nova é criada

5. Se qualquer uma destas mudanças implica em mudança nos **DataSets**, como por exemplo adicionar ou modificar membros, os **DataSets** são apagados e recriados imediatamente.

IMPORTANTE

Se um novo Tag é criado, este deve ser colocado em modo **Advise** (em *scan*) somente após salvar o banco de dados, ou seja, para adicionar um novo Tag ao **Cliente** ICCP em tempo de execução, primeiro adicione o ponto à Tabela Bilateral e salve-a, usando o Tag **SaveDatabase**, e depois crie um Tag de Comunicação neste Driver de uma aplicação **Elipse E3** ou **Elipse Power** e coloque-o em modo **Online**, configurando a propriedade **AllowRead** com o valor Verdadeiro.

6. Exemplo de criação de um Tag em uma aplicação no **Elipse E3** ou no **Elipse Power** em tempo de execução.

```
Set NewTag = Item("ICCPDriver").AddObject("IOTag", False, "MyTagName")
NewTag.N1 = 1 'Este é um Tag Cliente
NewTag.AllowRead = True
NewTag.Activate
```

Arquivo CEPEL SAGE

Os parâmetros compatíveis com arquivos de configuração CEPEL SAGE (*Sistema Aberto de Gerenciamento de Energia*) são os seguintes:

- Na aba **ICCP General**:
  - **Local P Selector (CNF - Config, PS)**: 1 (um)
  - **Local S Selector (CNF - Config, SS)**: 1 (um)
  - **Local T Selector (CNF - Config, TS)**: 1 (um)
  - **Local App ID (CNF - Config, Aptitle RRR)**: 1, 1, 3
  - **Local AE Qualifier (CNF - Config, AeQ)**: 1 (um)

- Na aba **ICCP Client**:
  - **Remote Domain (CNF - ID)**: Nome de um domínio remoto
  - **Expected Billateral Table Name (LSC - VerBD)**: 0 (zero)
  - **Remote P Selector (CNF - Config, PS)**: 1 (um)
  - **Remote S Selector (CNF - Config, SS)**: 1 (um)

- **Remote T Selector (CNF - Config, TS)**: 1 (um)
- **Remote App ID (CNF - Config, Aptitle LLL)**: 1, 1, 2
- **Remote AE Qualifier (CNF - Config, AeQ)**: 1 (um)

Documentação das Interfaces de Comunicação

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

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.

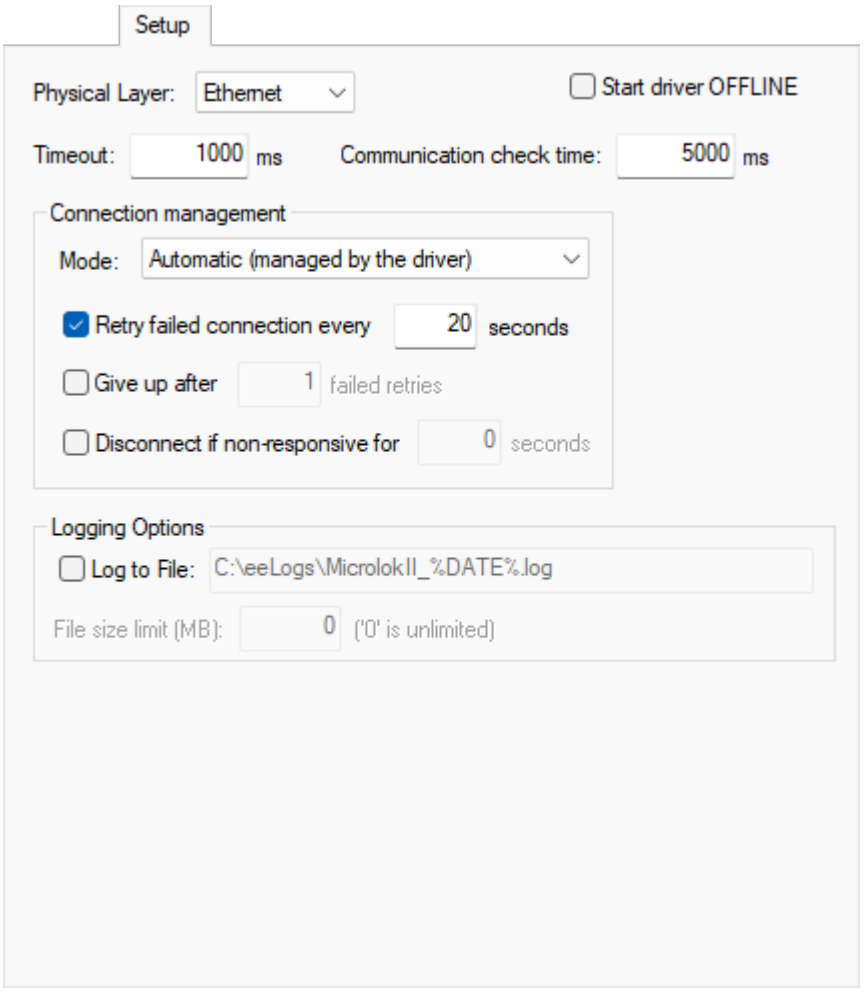
Caixa de Diálogo de Configuração

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

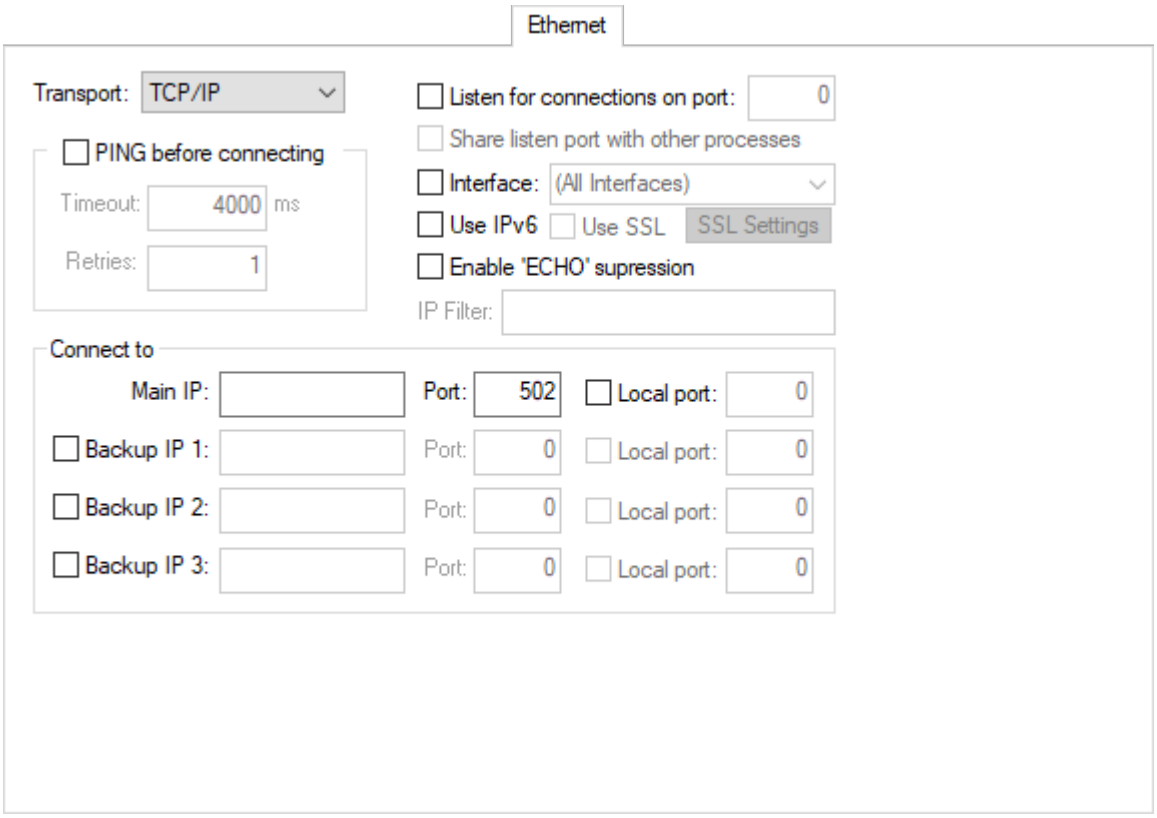
OPÇÃO	DESCRIÇÃO
Mode	Seleciona o modo de gerenciamento de conexão. Selecionar a opção **Automatic** permite que um Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção **Manual** permite que uma aplicação gerencie a conexão completamente
Retry failed connection 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

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**.



Aba Ethernet

Opções disponíveis na aba Ethernet

OPÇÃO	DESCRIÇÃO
Transport	Selecione o valor **TCP/IP** para um *socket* TCP (*stream*) ou selecione o valor **UDP/IP** para utilizar um *socket* UDP (*connectionless datagram*)
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 *eco* dos dados recebidos. O *eco* 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 (*Firewall*). 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 *socket*. Esta é uma maneira rápida de determinar uma conexão bem sucedida antes de tentar abrir um *socket* com um dispositivo. O *time-out* de uma conexão com um *socket* 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 *socket* 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 *backup* 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

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.

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:*:*)**: Aceita conexões de endereços IPv6 no intervalo entre fe80:03bf:0877:0000:0000:0000:0000:0000 e fe80:03bf:0877:0000:0000:0000: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

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

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

IO.Ethernet.PingTries

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

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

IO.Ethernet.UseIPv6

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

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.36	10/08/2026	M. Ludwig	• Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37979*).
2.0.30	01/09/2023	M. Salvador	• Corrigido o comportamento dos Tags **COV** (*Case 34009*).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
2.0.29	20/03/2023	M. Salvador	• Criada uma opção de atraso para reportar uma qualidade ruim no lado **Cliente** (*Case 33894*).
2.0.24	08/02/2022	M. Salvador	• Criada uma opção para envio de números inteiros com o menor número possível de bytes nas camadas OSI. Inclui também melhorias nas mensagens de logs (*Case 29240*). • Criada uma opção para definir se um **Servidor** suporta o uso de *Blocked Transfers*, reportados na leitura da variável reservada **Supported_Features** (*Case 29985*). • Quando a criação de um **DataSet** no modo **Cliente** falha, agora este Driver tenta ler cada um dos Tags individualmente para descobrir se algum destes Tags está causando esta falha (*Case 30099*). • Melhorias na interpretação de erros na criação de **DataSets** (*Case 30647*). • Corrigido o objeto **ConnectionPresentationPDU**, que continha um membro com o tipo de dados errado, com somente 4 (quatro) bytes (*Case 29239*). • Relatórios enviados com o nome dos campos em **ListOfVariables** agora permitem que o escopo seja tanto **ICC** quanto **VCC** (*Case 29984*). • Padronizado o comportamento de desconexão operando como **Servidor**, na ausência de quitação de eventos. Corrigida também uma possível situação em que, após uma reconexão, este Driver ainda exigia a quitação de eventos que

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			não existem mais (*Case 30866*). • Criada uma opção para descartar leituras com estampas de tempo nulas (NTR, *Null Timestamp Responses*).
2.0.18	05/08/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (*Case 27151*). • No modo **Servidor**, após uma reconexão a transferência de dados com um **Cliente** poderia não funcionar novamente, mesmo com este **Cliente** recriando os mesmos **DataSets** que existiam antes (*Case 27037*). • Corrigido um erro aleatório de processamento de relatório bloqueado, além de implementadas melhorias em logs ao receber uma notificação de evento (*Case 26139*).
2.0.14	24/09/2018	M. Salvador	• Driver portado para a biblioteca **IOKit** versão **2.0**. • Implementada uma função para importação de arquivos no formato **XML**. • Melhorias em equipamentos. • Implementado o modo **Offline**.
1.0.7	22/11/2012	M. Salvador	• Status das conexões em **Servidores** e **Clientes**. • Operações em batelada em um **Servidor**.
1.0.6	03/11/2011	M. Salvador	• Correções de erros.
1.0.2	01/08/2011	M. Salvador	• 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 © 2008-2026 Elipse Software Ltda. Todos os direitos reservados.