

Driver MQTT

Nome do Arquivo	MQTT.dll
Fabricante	
Equipamentos	Clientes e Brokers compatíveis com as especificações 3.1 e 3.1.1 do protocolo MQTT
Protocolo	MQTT sobre Ethernet TCP/IP e TLS/SSL
Versão	1.0.30
Última Atualização	22/12/2021
Plataforma	Win32
Dependências	IOKit v2.00 e a biblioteca OpenSSL
Leitura com SuperBlocos	Não
Nível	31312

Introdução

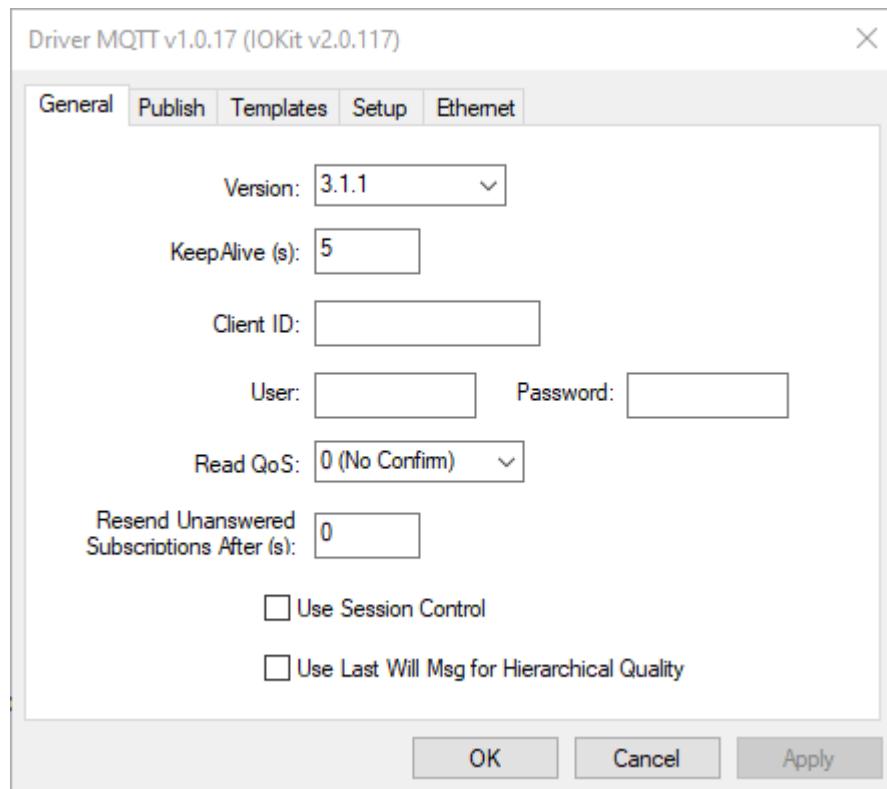
O Driver MQTT é um cliente que implementa a versão 3.1.X do protocolo MQTT usando um *Broker* MQTT, não fornecido.

Este Driver é capaz de receber e extraír valores de mensagens recebidas, bem como enviar mensagens que são processadas por outros clientes.

As conexões entre os clientes não são realizadas diretamente, somente através de um *Broker*.

Parâmetros de Configuração do Driver

Os parâmetros [P] de configuração deste Driver não são utilizados. Todas as configurações são executadas na janela de configurações, mostrada na figura a seguir.



Aba General

As opções disponíveis na aba **General** estão descritas na tabela a seguir.

Opções disponíveis na aba General

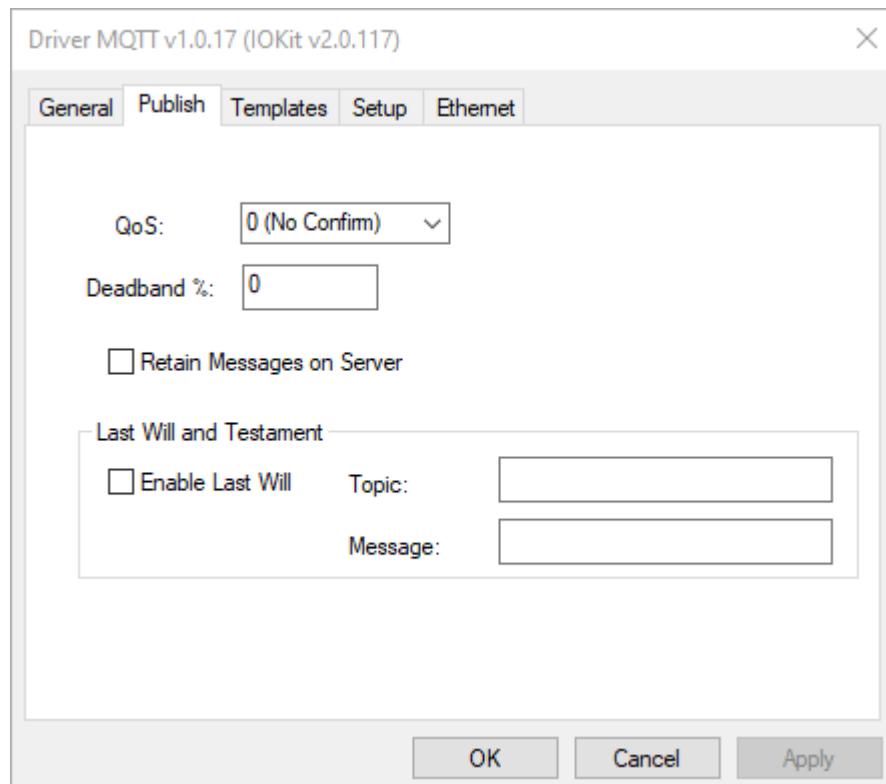
OPÇÃO	DESCRIÇÃO
Version	A versão do protocolo. As opções disponíveis são 3.1 ou 3.1.1 .
KeepAlive (s)	Tempo, em segundos, que o cliente envia uma mensagem de ping no protocolo MQTT (PingReq) para verificar se a conexão com o <i>Broker</i> ainda está ativa. O valor padrão para esta opção é 5 (cinco) segundos
Client ID	Nome que este cliente utiliza como identificador para os demais clientes. Não deve haver outro cliente conectado no mesmo <i>Broker</i> com este nome
User	Nome de usuário usado nas mensagens de conexão (ConnAck), caso desejado e permitido pelo <i>Broker</i>
Password	Senha usada juntamente com o nome de usuário
Read QoS	Indica que QoS é usado para solicitar as assinaturas dos Tags (QoS 0, 1 ou 2). O <i>Broker</i> compara o QoS solicitado e o QoS originalmente publicado, enviando um dado no mínimo compatível
Resend Unanswered Subscriptions After (s)	Informe o número de segundos para reenviar uma assinatura de um item se este não recebeu o primeiro valor dentro do tempo informado
Use Session Control	Informa se este Driver deve manter o estado da sessão ativa, ou seja, caso ocorra uma desconexão e posterior reconexão, as mensagens QoS 1 e 2 enviadas por outros clientes podem ser recuperadas

OPÇÃO	DESCRIÇÃO
Use Last Will Msg for Hierarchical Quality	Ao selecionar esta opção, se este Driver receber a mensagem configurada na opção Message do grupo Last Will and Testament da aba Publish , então o tópico da mensagem é utilizado para definir hierarquicamente outros Tags que são configurados com qualidade ruim

Supondo que a mensagem de *Last Will* de um equipamento é configurada com a expressão "Error" para o item **Device527**, ao receber esta mensagem todos os Tags endereçados que contiverem a expressão "Device527" a partir do início do endereço têm a qualidade configurada como ruim, ou seja, a propriedade **Quality** tem o valor configurado para 20, 24 ou 28 (inválida). Exemplos de endereçamentos de equipamentos:

```
Device527/Temperature
Device527/AnalogInput01
Device527
```

Para que estes Tags retornem à qualidade boa (192), o tópico deve receber algum valor diferente da mensagem configurada em **Last Will and Testament**.



Aba Publish

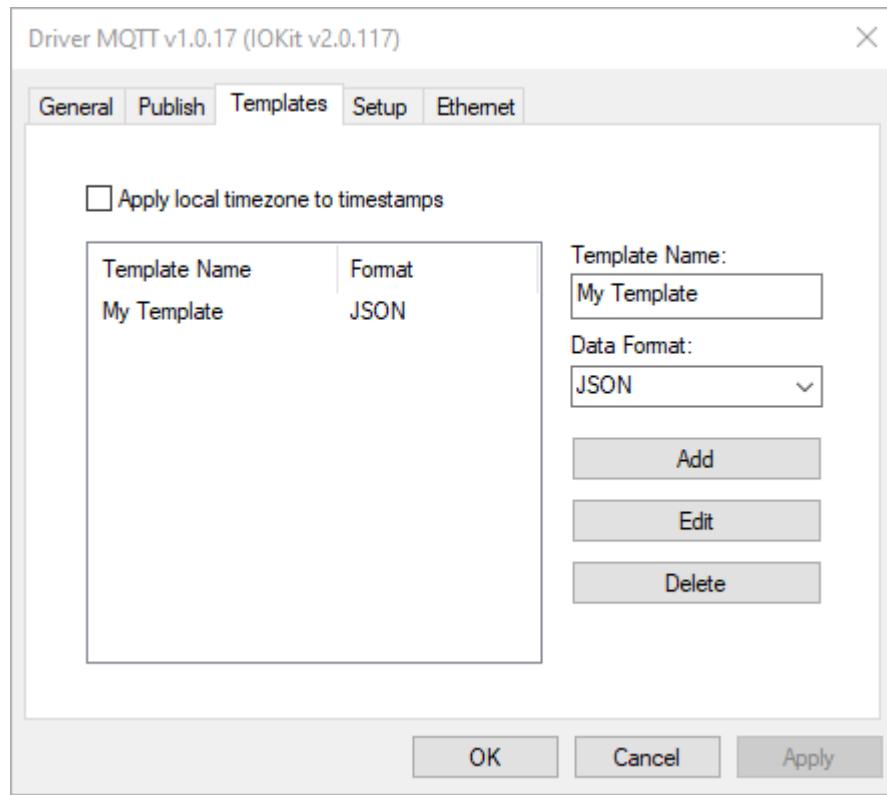
As opções disponíveis na aba **Publish** estão descritas na tabela a seguir.

Opções disponíveis na aba Publish

OPÇÃO	DESCRIÇÃO
QoS	Indica o nível de serviço das mensagens publicadas por este Driver. As opções disponíveis são QoS 0 : Quase sempre uma entrega (pode haver zero ou mais entregas), QoS 1 : Pelo menos uma ou mais entregas ou QoS 2 : Exatamente uma entrega

OPÇÃO	DESCRIÇÃO
Deadband %	Valor da banda morta relativa a ser aplicada em todos os Tags na escrita de valores numéricos sem <i>Templates</i>
Retain Messages on Server	Selecionar esta opção instrui ao <i>Broker</i> para manter as mensagens deste cliente juntamente com o QoS para que clientes que se subscreverem no futuro possam receber mensagens acumuladas. Quando um novo cliente subscreve, a última mensagem de cada tópico, se houver, deve ser enviada ao cliente pelo <i>Broker</i>
Enable Last Will	Ao habilitar esta opção, o servidor deve manter em memória um tópico e mensagem para o cliente, que são enviados para todos os demais clientes quando houver uma desconexão não ordenada, ou seja, sem o envio da mensagem de <i>Disconnect</i>
Topic	Tópico ou endereço que é enviado na desconexão (<i>Last Will</i>), como por exemplo "ELIPSE\goodbye"
Message	Mensagem associada ao tópico, como por exemplo "Adeus"

Na aba **Templates** é possível definir diferentes padrões para os dados da mensagem MQTT, cujos elementos podem ser automaticamente extraídos ou preenchidos por este Driver através de Associações com Tags ou Blocos de Comunicação.



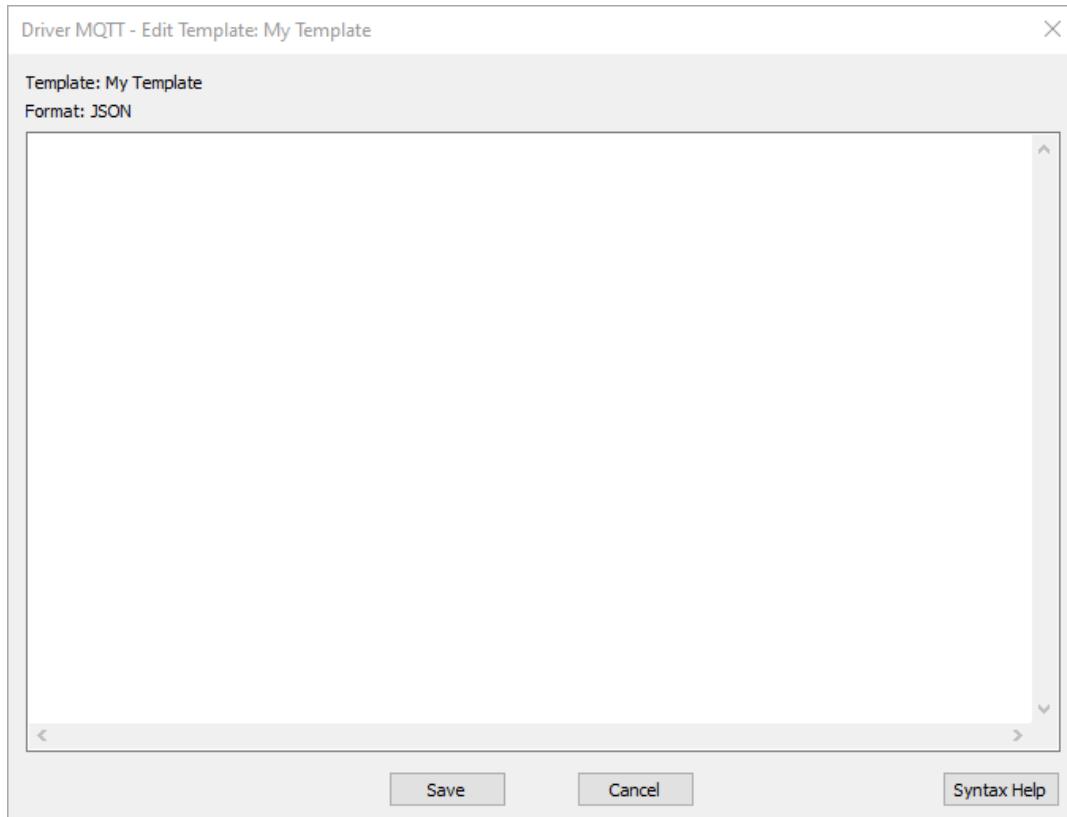
Aba Templates

As opções disponíveis na aba **Templates** estão descritas na tabela a seguir.

Opções disponíveis na aba Templates

OPÇÃO	DESCRÍÇÃO
Apply local timezone to timestamps	Quando esta opção está selecionada, as estampas de tempo recebidas são convertidas para o fuso horário utilizado pelo sistema operacional. Do contrário, as estampas de tempo recebidas são exibidas no fuso horário UTC
Template Name	Informa o nome do <i>Template</i> desejado. Não podem existir dois <i>Templates</i> com o mesmo nome
Data Format	Declara o formato da mensagem do <i>Template</i> . As opções disponíveis são JSON , CSV ou XML . O formato JSON é o preferido para uso pelo MQTT, pois permite a definição de qualquer tipo de estrutura de dados com o menor número de bytes
Add	Adiciona um <i>Template</i> na lista de <i>Templates</i>
Edit	Abre uma janela para edição do <i>Template</i> selecionado
Delete	Apaga o <i>Template</i> selecionado da lista de <i>Templates</i>

Clique em **Edit** para abrir a janela para edição do *Template* selecionado. O nome e o formato do *Template* são exibidos na parte superior. O *Template* propriamente dito deve ser inserido na caixa de edição.

**Janela para edição de um Template**

As opções disponíveis na janela para edição do *Template* estão descritas na tabela a seguir.

Opções disponíveis na janela para edição do Template

OPÇÃO	DESCRIÇÃO
Save	Registra as modificações realizadas e fecha esta janela
Cancel	Descarta as modificações realizadas e fecha esta janela
Syntax Help	Informa a sintaxe da mensagem, no formato selecionado (JSON , CSV ou XML) juntamente com palavras-chave que são usadas durante o processamento para substituição por estampas de tempo, qualidades ou valores. Para mais informações, consulte o tópico Sintaxe dos Templates

Na aba **Ethernet** deve ser configurado o endereço IP e a porta TCP/IP do *Broker MQTT*. A norma MQTT estabelece a porta TCP/IP 1883 como padrão para conexões diretas sem criptografia. A porta TCP/IP 8883 é definida para conexões criptografadas TLS/SSL sem certificado e a porta TCP/IP 8884 é definida para conexões TLS/SSL com certificado.

Sintaxe dos Templates

Para extrair o conteúdo de uma mensagem é necessário declarar um *Template*, que permite informar o formato da mensagem e quais partes devem ser transformadas em dados. Os *Templates* podem ser do tipo **JSON**, **CSV** ou **XML**. Cada *Template* deve utilizar palavras-chave, que devem ser colocadas no lugar dos valores a serem extraídos. As palavras-chave disponíveis estão descritas na tabela a seguir.

Palavras-chave disponíveis para Templates

PALAVRA-CHAVE	DESCRIÇÃO
TS_TEXT(format)	Estampa de tempo, em formato textual, que é usada como estampa de tempo do Tag Bloco ou Tag de Comunicação. O significado de cada campo está descrito na tabela Opções disponíveis para a palavra-chave TS_TEXT
TS_UNIX	Estampa de tempo, no formato de segundos desde 1970 (formato UNIX). Este valor pode ser numérico ou textual, e é usado como estampa de tempo do Tag Bloco ou Tag de Comunicação, como por exemplo 1504198675 ou "1504198675"
TS_DAYS1900	Estampa de tempo, em formato numérico, cujo valor é o número de dias desde primeiro de janeiro de 1900, também conhecido como Calendário Gregoriano, e é usado como estampa de tempo do Tag Bloco ou Tag de comunicação, como por exemplo 43595.37373843. Esta palavra-chave só está disponível em <i>Templates</i> do tipo JSON ou XML
TS_UNIX_TZ_SECONDS+ TS_UNIX_TZ_SECONDS-	Fuso horário, em segundos, informado separadamente da palavra-chave TS_UNIX . Deve ser informado um símbolo de + (mais) ou - (menos) para indicar se os segundos devem ser adicionados ou removidos, respectivamente, da palavra-chave TS_UNIX para representar a estampa de tempo da mensagem. Esta palavra-chave só está disponível em <i>Templates</i> do tipo JSON
QL_OPC	Qualidade, no padrão OPC DA, utilizando um byte. Este valor é usado diretamente como valor de qualidade no Tag Bloco ou Tag de Comunicação, sem transformações. Consulte a tabela Padrão OPC DA para mais informações
QL_BOOL	Qualidade no padrão Booleano. Se o valor é maior que 0 (zero) ou a expressão "true" ou "TRUE", a qualidade é boa. Se o valor é igual a 0 (zero) ou a expressão "false" ou "FALSE", a qualidade é ruim

PALAVRA-CHAVE	Descrição
V1, V2, ... V50 Template JSON	Declara um ou mais valores, que podem ser extraídos individualmente para Tags de Comunicação, através da especificação do terceiro parâmetro de endereçamento (<i>param</i>). Se todos os valores de uma mensagem são mapeados para um Tag Bloco único, então é possível usar a palavra-chave E3VAL para qualquer valor a ser extraído, indistintamente
E3VAL	Especifica um valor que é extraído na sequência que ocorre para um Tag Bloco, cada valor em um Elemento de Bloco. Se houver apenas um valor E3VAL na mensagem, o Template também pode ser usado com um Tag de Comunicação
DUMMY	Campo variável, mas cujo valor não deve ser enviado para Tags de Comunicação
Repeat_E3VAL	Palavra chave que só pode ser usada dentro de um array JSON, para indicar que os membros que se repetem dentro desse array devem ser lidos de forma independente pelo driver, retornando uma leitura para cada conjunto. Para mais informações, veja a seção Template JSON.

Opções disponíveis para a palavra-chave **TS_TEXT**

OPÇÃO	Descrição
%a	Dia da semana abreviado, em inglês
%A	Dia da semana completo, em inglês
%b	Nome do mês abreviado, em inglês
%B	Nome do mês completo, em inglês
%C	Século
%d	Dia do mês, começando com 0 (zero)
%e	Dia do mês, começando com um espaço
%f	Milissegundos, de 0 (zero) a 999
%h	Hora, no formato de 12 horas
%H	Hora, no formato de 24 horas
%m	Mês
%M	Minuto
%p	AM ou PM
%S	Segundos
%y	Ano com dois dígitos
%Y	Ano com quatro dígitos
%Z	Nome do fuso horário, um código internacional que transforma o horário para GMT
%+	Deslocamento (<i>offset</i>) GMT no formato +-HH:MM

Exemplos de estampas de tempo formatados pela palavra-chave **TS_TEXT**:

```
"2014-07-11T15:26:37Z" -> "TS_TEXT(%y-%m-%dT%H:%M:%SZ)"
"Mon Jul 10 11:04:47 BRT 2017" -> "TS_TEXT(%a %b %d %H:%M:%S %Z %Y)"
"2018-05-02T10:29:28.622-02:00" -> "TS_TEXT(%Y-%m-%dT%H:%M:%S.%f%+)"
```

Padrão OPC DA

BIT	7	6	5	4	3	2	1	0
Descrição	Q	Q	S	S	S	S	L	L

Opções disponíveis para o padrão OPC DA

OPÇÃO	DESCRIÇÃO
QQ	Dois bits de qualidade
SSSS	Quatro bits de sub-status
LL	Dois bits de limite
QQ	0: BAD, 1: UNCERTAIN ou 3: GOOD
SSSS	0: BAD_NONSPECIFIC, 1: BAD_CONFIGERROR, 2: BAD_NOTCONNECTED, 3: BAD_DEVICEFAILURE ou 4: BAD_SENSORFAILURE
SSSS	0: UNCERT_NONSPECIFIC, 1: UNCERT_LASTUSABLEVALUE ou 4: UNCERT_SENSORNOTACCURATE
SSSS	0: GOOD_NONSPECIFIC, 1: GOOD_LOCALOVERRIDE ou 6: GOOD_NONSPECIFICLOCALTIMESTAMP
LL	0: FREE, 1: LOW, 2: HIGH ou 3: CONST

Template JSON

JSON (JavaScript Object Notation) é um formato de troca de dados de fácil compreensão para humanos, e também simples para que programas realizem o processamento e extração de informações. O formato **JSON** é construído sobre duas estruturas, uma coleção de pares contendo nome e valor e uma lista ordenada destes pares.

Um **Objeto** é uma lista não ordenada de pares de nomes e valores. Um objeto é delimitado por chaves, cada nome é definido dentro de aspas duplas seguido por dois pontos e o respectivo valor, e cada par é separado por vírgulas.

Um **Array** é uma lista ordenada de valores. Um **array** é delimitado por colchetes e os valores são separados por vírgulas. Valores podem ser números ou textos, que devem ser representados entre aspas duplas. Exemplo de uma mensagem no formato **JSON**:

```
{"s":1, "t":"2014-07-11T15:26:37Z", "q":192, "c":1, "x":-1.234, "y":0.234, "z":-0.234}
```

Para extrair o conteúdo da mensagem anterior, um possível *Template* seria o seguinte:

```
{"s": "DUMMY", "t": "TS_TEXT(%y-%m-%dT%H:%M:%SZ)", "q": "QL_OPC", "c": "DUMMY", "x": "E3VAL", "y": "E3VAL", "z": "E3VAL"}
```

Neste caso, deve ser criado um Tag Bloco com três Elementos de Bloco, e cada Elemento de Bloco recebe um dos valores **x**, **y** e **z**, respectivamente.

O código a seguir contém outro exemplo de mensagem no formato **JSON**:

```
{
  "n_channels":2,
  "timestamp":1504198675,
  "hash":"1842E0F97392F08BDF996961A8333832AB06D113",
  "battery":5.13,
  "gmt":-3,
  "tag_channels":["Analog1","Analog2"],
  "value_channels":[28.100,27.200],
  "tag_units":["°C","°C"],
  "alarm_low":[0,0],
  "alarm_high":[0,0],
  "buzzer_state":0
}
```

Um possível *Template* para o exemplo anterior seria o seguinte:

```
{
  "n_channels":"DUMMY",
  "timestamp":"TS_UNIX",
  "hash":"DUMMY",
  "battery":"V1",
  "gmt":"V2",
  "tag_channels":["V3","V4"],
  "value_channels":["V5","V6"],
  "tag_units":["V7","V8"],
  "alarm_low":["V9","V10"],
  "alarm_high":["V11","V12"],
  "buzzer_state":"V13"
}
```

Observação: Tanto a mensagem como os templates devem ser objetos ou arrays, portanto devem começar com um { ou [.

Repetição de Valores

Os arrays podem ser usados para receber um número variável de elementos. Vamos tomar como exemplo as mensagens abaixo:

Mensagem 1:

```
{
  "MessageType": "Teste",
  "TagData": [
    {
      "Temperature": 11,
      "Humidity": 50
    },
    {
      "Temperature": 14,
      "Humidity": 80
    }
  ]
}
```

Mensagem 2:

```
{
  "MessageType": "Teste",
  "TagData": [
    {
      "Temperature": 11,
      "Humidity": 12
    }
  ]
}
```

```
{
  "Temperature": 14,
  "Humidity": 66
}
{
  "Temperature": 70,
  "Humidity": 55
}
}
```

Vejam que dentro do array, é possível que tenhamos 1, 2, 3 ou N conjuntos de temperatura e umidade em uma única mensagem, e seria interessante se um tag bloco "Teste" com dois elementos (Temperature e Humidity) pudesse receber uma leitura a cada conjunto.

Para isto, devemos declarar o template com uma keyword Repeat_E3VAL imediatamente após o colchete que marca o início do array. Exemplo:

```
{"MessageType":"DUMMY","TagData":[Repeat_E3VAL{"Temperature":"E3VAL","Humidity":"E3VAL"}]}
```

Desta forma, ao recebermos a mensagem 2, por exemplo, teremos uma chamada do evento **OnRead** do bloco para cada conjunto. Exemplo:

```
Sub [Sensor-001_OnRead]()
Application.Trace Item("Temperatura").Value
Application.Trace Item("Umidade").Value
End Sub
```

Log:

910	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	11	14
911	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	12	14
912	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	14	14
913	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	66	14
914	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	70	14
915	2021-08-05 10:06:15.826	0x3FF8 0x3210 APPTRACE	55	14

Caso tivessem sido declarados mais valores E3VAL antes da posição de repetição, esses valores seriam repetidos para cada conjunto. Como exemplo, se tivéssemos declarado o template como

```
{"MessageType":"E3VAL","TagData":[Repeat_E3VAL{"Temperature":"E3VAL","Humidity":"E3VAL"}]}
```

poderíamos ter o bloco com 3 elementos (Type, Temperatura e Umidade), sendo que receberíamos os seguintes valores a cada evento OnRead:

"Teste";11;12 => primeiro OnRead
 "Teste";14;66 => segundo OnRead
 "Teste";70;55 => terceiro OnRead

Template CSV

Um **Template CSV** permite processar valores que estejam separados por ponto e vírgula. Os valores, estampas de tempo e qualidades a ser extraídos da mensagem devem ser substituídos pelas palavras-chave. Um exemplo de mensagem no formato **CSV** é o seguinte:

18/10/2016 22:30:45; 22BAC2300P10; "Nível Alto";98,3;m
--

Um exemplo de *Template* para este formato, para ser associado a um Tag Bloco com quatro Elementos de Bloco, seria o seguinte:

```
TS_TEXT(%d/%m/%y %H:%M:%S);E3VAL;E3VAL;E3VAL;E3VAL
```

Template XML

Um **Template XML** permite processar valores em uma mensagem MQTT que sejam enviados no formato **XML**. Deve-se declarar o formato da mensagem e substituir os campos em nós a serem transformados em valores pelas palavras-chave. Um exemplo de mensagem no formato **XML** é o seguinte:

```
<updates>
  <update>
    <id>TEMP23A</id>
    <value>34.5</value>
    <quality>true</quality>
    <timestampl>18/10/2016 22:30:45</timestampl>
  </update>
</updates>
```

Um possível *Template* para a mensagem anterior seria o seguinte:

```
<updates>
  <update>
    <id>V1</id>
    <value>V2</value>
    <quality>QL_BOOL</quality>
    <timestampl>TS_TEXT(%d/%m/%y %H:%M:%S)</timestampl>
  </update>
</updates>
```

Configuração dos Tags

Utilize a sintaxe a seguir para a configuração de Tags no **E3** ou **Elipse Power**:

- **Dispositivo**: Não utilizado
- **Item**: Este campo deve obedecer a sintaxe a seguir

```
<Tópico>[;<Template>[;<Param>]]
```

Ou

```
<Tópico>[(Deadband:<Banda Morta Absoluta>)][(Deadband:<Banda Morta Relativa>%)]
```

Onde:

- **<Tópico>**: Endereço do item, ou tópico, no banco de dados do cliente. De acordo com a norma MQTT, cada item no banco de dados de um cliente pode corresponder a um conjunto de dados livre, transmitido em formato **Texto**. Os tópicos podem estar organizados em árvore, e neste caso é utilizada uma barra para a separação de níveis. Exemplo:

```
"station12/pump01/pressure1"
"station12/pump01/pressure2"
```

Um cliente também pode utilizar o sinal numérico (#) e o sinal de mais (+) para endereçar tópicos. No primeiro caso devem ser incluídos na resposta do pedido todos os itens que contiverem no nome o item atual ou os filhos. Por exemplo, para o valor "station12#" são retornados os seguintes valores:

```
"station12/pump01"
"station12/pump01/pressure1"
"station12/pump01/pressure2"
"station12/pump02"
"station12/pump02/pressure1"
"station12/pump02/pressure2"
```

No segundo caso devem ser incluídos na resposta do pedido todos os itens no mesmo nível. Por exemplo, para o valor "station12/pump01/+" são retornados os seguintes valores:

```
"station12/pump01/pressure1"
"station12/pump01/pressure2"
```

- **<Template>**: Informa o nome de um *Template* usado como modelo de interpretação dos dados da mensagem, com o objetivo de mapear campos dentro da mensagem para Tags (estampa de tempo, qualidade e valores). Este *Template* deve estar cadastrado na aba **Templates**
- **<Param>**: Informa o nome de um dos parâmetros no *Template*, caso este Tag deva receber apenas um dos parâmetros
- **<Banda Morta Absoluta>**: Banda morta em valor absoluto, a ser aplicada na escrita de valores numéricos
- **<Banda Morta Relativa>**: Banda morta em valor relativo, a ser aplicada na escrita de valores numéricos

A tabela a seguir contém exemplos de endereçamento de mensagens.

Exemplos de endereçamento

ITEM	DESCRIÇÃO
"station12/pump01"	Neste caso a mensagem lida ou escrita para o item é um Tag do tipo Texto (String) sem qualquer processamento
"station12/pump01;pumpdata"	Aplica-se o <i>Template pumpdata</i> ao conteúdo de station12/pump01 . Se o <i>Template</i> descreve mais de um valor, deve-se utilizar um Tag Bloco em que cada Elemento de Bloco é um valor do <i>Template</i>
"station12/pump01;pumpdata;V1"	Caso idêntico ao anterior, porém recuperando somente o primeiro valor do <i>Template</i> e associando-o ao Tag, que não precisa ser um Tag Bloco. Neste caso o <i>Template</i> precisa descrever o valor V1 . Para mais informações, consulte a aba Template no tópico Parâmetros de Configuração do Driver
"station12/pump01(Deadband:0.5)"	Caso o valor atual deste tópico seja numérico e um novo valor também numérico seja escrito neste Tag, então este novo valor só é publicado no tópico se a diferença entre o valor atual e o valor novo é maior ou igual a 0,5
"station12/pump01(Deadband:5%)"	Caso o valor atual deste tópico seja numérico e um novo valor também numérico seja escrito neste Tag, então este novo valor só é publicado no tópico se a diferença entre o valor atual e o valor novo é maior ou igual a 5% do valor antigo
"station12/pump01(Deadband:0.5)(Deadband:5%)"	Caso o valor atual deste tópico seja numérico e um novo valor também numérico seja escrito neste Tag, então este novo valor só é publicado no tópico se a diferença entre o valor atual e o valor novo é maior ou igual a 0,5 ou maior ou igual a 5% do valor antigo

NOTA

Quando a banda morta está configurada diretamente em um Tag, este Tag não deve ser associado a um *Template*. Neste caso, a banda morta configurada na aba **Publish** da janela de configurações do Driver é ignorada para este Tag.

Para superar a banda morta absoluta, é preciso satisfazer a seguinte condição:

```
|(Valor Atual) - (Valor Novo)| ≥ (Banda Morta Absoluta)
```

Para superar a banda morta relativa, é preciso satisfazer a seguinte condição:

```
|(Valor Atual) - (Valor Novo)| ≥ |(Valor Atual)| * (Banda Morta Relativa) / 100%
```

Se um Tag está configurado com banda morta relativa e banda morta absoluta simultaneamente, a satisfação de qualquer uma das condições anteriores resulta na publicação do valor.

Caso uma informação apareça em mais de uma mensagem e seja necessário que um Tag receba valores de todas estas mensagens, pode-se especificar um *array* de endereços, conforme a sintaxe a seguir.

```
["Topico1;Template1;Param1", "Topico2;Template2;Param2", "Topico3;Template3;Param3"]
```

Por exemplo, suponha que um equipamento envie dois tipos de mensagens. A mensagem **1** contém os valores **A**, **B** e **C** e a mensagem **2** contém os valores **D**, **E**, **F** e **B** novamente, nesta ordem. Caso deseje que um Tag receba os valores de **B** vindos de qualquer das duas mensagens, configure o *array* no formato a seguir.

```
["device023/msg1;template1;V2", "device023/msg2;template2;V4"]
```

Neste caso, o segundo valor configurado em **template1**, recebido no tópico **device023/msg1**, e o quarto valor configurado em **template2**, recebido no tópico **device023/msg2**, são enviados para este Tag.

NOTA

Pode-se repetir tópicos, como por exemplo dois ou mais Tags usando o mesmo tópico, mas um dos demais parâmetros, *template* ou *param*, deve ser diferente.

Exemplos de Tags:

```
Tag1: "device023/msg1" - OK
Tag2: "device023/msg1;T1" - OK
Tag3: "device023/msg1;T1;V2" - OK
Tag4: "["device023/msg1;T1;V2", "device023/msg1;T1;V3"]" - Não aceito. O primeiro elemento repete Tag3
```

Publicação de Valores

Quando é necessário que o driver publique mensagens para outros clientes, basta realizar a escrita do tag desejado. Se o tag não possui um template associado, então o valor informado será enviado para o tópico. Neste caso, se for um valor numérico ele será convertido para texto, e se já for um texto será enviado como está.

Se o tag possui um template associado, então as variáveis que forem declaradas dentro do template serão preenchidas com os valores vindos do tag e depois enviados. Note que se você usar um tag simples, o template deverá ter apenas uma variável declarada. Se você precisar de mais de uma variável, deve usar um tag bloco, onde cada elemento do bloco vai então corresponder a uma variável dentro do template, na mesma ordem.

Para executar então a escrita de um bloco, é necessário que a sua aplicação realize a escrita do bloco inteiro de uma vez (Block.Write()) através de um script, e para isso devemos então primeiramente setar o valor de cada um dos elementos, e depois escrever o bloco inteiro. As propriedades AllowRead e AllowWrite devem estar em FALSE.

Exemplo de escrita em um template que tem 3 variáveis:

```
Set Block1 = Application.GetObject("DriverMQTT").Item("Block1")
Block1.Item("Elemento1").Value = 123
Block1.Item("Elemento2").Value = "Abnormal Situation"
Block1.Item("Elemento3").Value = 555
Block1.Write 1  'escrita síncrona, 2 = escrita assíncrona
```

Documentação das Interfaces de Comunicação

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

Configurações do Driver

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

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

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

1. Abra o Organizer.
2. Selecione o 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, se for necessário o acesso a duas portas seriais, é preciso adicionar dois Drivers na aplicação e configurar cada um destes Drivers para cada porta serial.

Caixa de Diálogo de Configuração

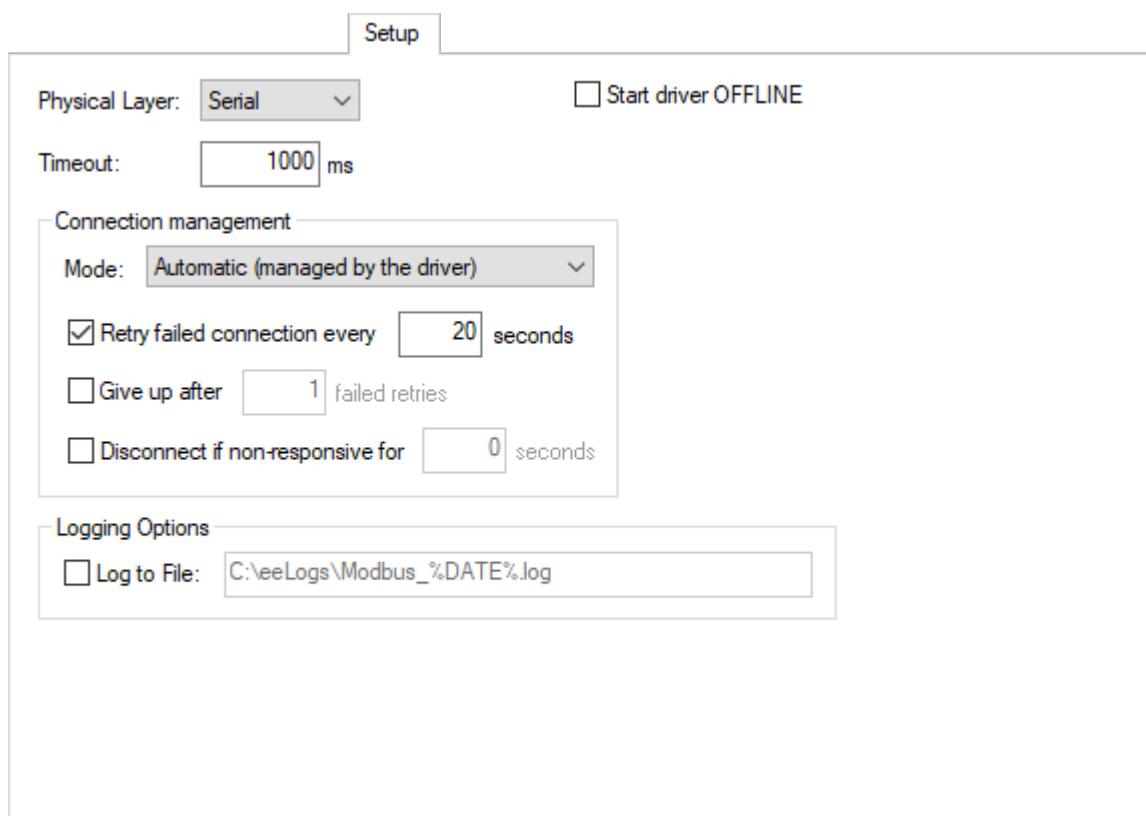
A caixa de diálogo das Interfaces de Configuração permitem configurar a conexão de I/O que é utilizada pelo Driver. Esta caixa de diálogo contém as abas **Setup**, **Serial**, **Ethernet**, **Modem** e **RAS**, descritas nos tópicos a seguir. Se um Driver não implementa uma conexão de I/O específica,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

Aba Setup

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

- **Configurações gerais:** Configurações da camada física do 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 são Serial , Ethernet , Modem e RAS . A interface selecionada deve ser configurada na sua 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
Start driver OFFLINE	Seleciona esta opção para que o Driver inicie em modo Offline ou parado. Isto significa que a interface de I/O não é criada até que se configure o Driver em modo Online utilizando-se um Tag na 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 o Driver gerencie a conexão automaticamente, como especificado nas opções seguintes. Selecionar a opção Manual permite

OPÇÃO	DESCRIÇÃO
	que a aplicação gerencie a conexão completamente. Consulte o tópico Estados do Driver para mais informações
Retry failed connection every ... seconds	Selecione esta opção para habilitar a retentativa de conexão do Driver em um determinado intervalo, em segundos. Se a opção Give up after failed retries não está selecionada, o 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 for atingido, o Driver vai para o modo Offline , assumindo que um problema de hardware foi detectado. Se o 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 o 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	<p>Habilite esta opção e configure o nome do arquivo onde o log é escrito. Arquivos de log podem ser bem extensos, portanto utilize esta opção por curtos períodos de tempo, apenas para o propósito de testes e depurações.</p> <p>Caso se utilize a macro %PROCESS% no nome do arquivo de log, esta é substituída pelo ID do processo atual. Esta opção é particularmente útil ao se utilizar várias instâncias do mesmo Driver no E3, permitindo assim que cada instância gere um arquivo separado de log. Por exemplo, ao configurar esta opção como c:\e3logs\drivers\sim_%PROCESS%.log, gera-se o arquivo c:\e3logs\drivers\sim_0000FDA.log para o processo 0FDAh.</p> <p>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 como c:\e3logs\drivers\sim_%DATE%.log, gera-se o arquivo c:\e3logs\drivers\sim_2005_12_31.log no dia 31 de dezembro de 2005 e o arquivo c:\e3logs\drivers\sim_2006_01_01.log no dia primeiro de janeiro de 2006</p>

Aba Ethernet

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

Ethernet

Transport: TCP/IP ▼ <input type="checkbox"/> PING before connecting Timeout: 4000 ms Retries: 1	<input type="checkbox"/> Listen for connections on port: 0 <input type="checkbox"/> Share listen port with other processes <input type="checkbox"/> Interface: (All Interfaces) ▼ <input type="checkbox"/> Use IPv6 <input type="checkbox"/> Use SSL SSL Settings <input type="checkbox"/> Enable 'ECHO' suppression IP Filter:																								
Connect to <table border="0"> <tr> <td>Main IP:</td> <td><input type="text"/></td> <td>Port:</td> <td>502</td> <td><input type="checkbox"/> Local port:</td> <td>0</td> </tr> <tr> <td><input type="checkbox"/> Backup IP 1:</td> <td><input type="text"/></td> <td>Port:</td> <td>0</td> <td><input type="checkbox"/> Local port:</td> <td>0</td> </tr> <tr> <td><input type="checkbox"/> Backup IP 2:</td> <td><input type="text"/></td> <td>Port:</td> <td>0</td> <td><input type="checkbox"/> Local port:</td> <td>0</td> </tr> <tr> <td><input type="checkbox"/> Backup IP 3:</td> <td><input type="text"/></td> <td>Port:</td> <td>0</td> <td><input type="checkbox"/> Local port:</td> <td>0</td> </tr> </table>		Main IP:	<input type="text"/>	Port:	502	<input type="checkbox"/> Local port:	0	<input type="checkbox"/> Backup IP 1:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0	<input type="checkbox"/> Backup IP 2:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0	<input type="checkbox"/> Backup IP 3:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0
Main IP:	<input type="text"/>	Port:	502	<input type="checkbox"/> Local port:	0																				
<input type="checkbox"/> Backup IP 1:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0																				
<input type="checkbox"/> Backup IP 2:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0																				
<input type="checkbox"/> Backup IP 3:	<input type="text"/>	Port:	0	<input type="checkbox"/> Local port:	0																				

Aba Ethernet**Opções disponíveis na aba Ethernet**

OPÇÃO	DESCRIÇÃO
Transport	Selecione TCP/IP para um socket TCP (<i>stream</i>) ou selecione UDP/IP para utilizar um socket 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, o Driver se conecta ao endereço e porta especificados na opção 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 por seu endereço IP, que o Driver utiliza para efetuar e receber conexões, ou selecione o item (All Interfaces) para permitir conexões em qualquer interface de rede
Use IPv6	Selecione esta opção para forçar o 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>echo</i> dos dados recebidos. O <i>echo</i> é uma cópia dos dados que foram enviados, que pode ser retornada antes da mensagem de resposta
IP Filter	Lista de endereços IP restringidos ou permitidos de onde o 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, verificar se o dispositivo pode ser encontrado na rede,

OPÇÃO	DESCRÍÇÃO
	<p>no dispositivo antes de tentar uma conexão 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 o dispositivo. O <i>time-out</i> de uma conexão com um <i>socket</i> pode ser bem alto. As opções disponíveis são:</p> <ul style="list-style-type: none"> • Timeout: Especifique o número de milissegundos de espera por uma resposta do comando ping. Deve-se usar o 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 um e quatro segundos • Retries: Número de retentativas do comando ping, sem contar a tentativa inicial. Se todos as tentativas falharem, então a conexão com o <i>socket</i> é abortada

Opções disponíveis no grupo Connect to

OPÇÃO	DESCRÍÇÃO
Main IP	Digite o endereço IP do dispositivo remoto. Pode-se usar tanto o endereço IP separado por pontos quanto uma URL. No caso de uma URL, o 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 do dispositivo remoto, entre 0 (zero) e 65535
Local port	Selecione esta opção para utilizar uma porta IP local fixa ao conectar ao 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> do dispositivo remoto

Configurações Gerais

Esta seção contém informações sobre a configuração dos **Tags de Comunicação** e as **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.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** do Bloco representa o momento em que o evento ocorreu. 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 do evento. Os valores possíveis são **0**: Driver (específico do 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 do evento, uma **String** específica de cada evento

NOTA

O 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 possíveis valores são os seguintes:

- **0:** Camada física parada, ou seja, o 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, o Driver está em modo **Online**, mas a camada física não está conectada. Se a opção **Connection management** estiver configurada para **Automatic**, a camada física pode estar

conectando, desconectando ou esperando por uma tentativa de reconexão. Se a opção **Connection management** estiver configurada para **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 o 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 do Driver em tempo de execução.

Este Tag funciona somente enquanto o Driver estiver em modo **Offline**. Para iniciar o Driver em modo **Offline**, selecione a opção **Start driver OFFLINE** na caixa de diálogo de configuração do 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, o 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 três parâmetros, então o tamanho do Bloco deve ser 6 (3 × 2). O primeiro Elemento é o nome da propriedade, como uma **String**, e o segundo Elemento é o valor da 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 **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** do Driver para enviar os parâmetros diretamente para o 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

```

O Driver não valida nomes de parâmetros ou valores passados, por isto tenha cuidado ao escrever parâmetros e valores. O comando **Write** falha se o *array* de configuração é criado incorretamente. Pode-se consultar o log do Driver ou usar o parâmetro *writeStatus* do método **WriteEx** para descobrir a causa exata do 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 do 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 do Driver através do Tag **IO.SetConfigurationParameters**
- **1 - Driver Online:** A camada física está aberta ou em execução. Enquanto estiver em modo **Online**, a camada física pode ser conectada ou desconectada e seu estado atual pode ser conferido no Tag **IO.PhysicalLayerStatus**

No exemplo a seguir, utilizando o **E3**, o 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 quando estiver configurando o Driver em modo **Online**, ou seja, escrevendo o valor 1 (um). Neste caso, o 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**
- O Driver pode ter ficado sem memória
- A camada física pode ter deixado de criar seu *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 do 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 o Driver gerencia a conexão ou **1**: Modo manual, em que a aplicação gerencia a conexão.

IO.GiveUpEnable

■ Quando configurada para Verdadeiro, define um número máximo de tentativas de reconexão. Se todas as reconexões falharem, o Driver entra em modo **Offline**. Se configurada para Falso, o 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), o Driver tenta apenas uma reconexão quando a conexão é perdida. Se esta falhar, o 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 estiver 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 estiver inativa por este período de tempo, então é desconectada.

IO.RecoverEnable

Configure em Verdadeiro para habilitar o Driver a recuperar conexões perdidas e em Falso para deixar o 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 o Driver em modo **Offline** e em Falso para iniciar o Driver em modo **Online**.

NOTA

Não faz sentido modificar esta propriedade em tempo de execução, já que esta só pode ser modificada quando o Driver já estiver em modo **Offline**. Para configurar o 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 pelo Driver. Os valores possíveis são os seguintes:

- **N ou None:** Não utiliza uma interface física, ou seja, o Driver deve fornecer uma interface personalizada
- **S ou Serial:** Utiliza uma porta serial local (COMn)
- **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*). O Driver conecta-se ao 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 as **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 o Driver está conectado na conexão atual ou 0 (zero) se o 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 o Driver está desconectado desde o término da última conexão ou 0 (zero) se o 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 o 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 o 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 o 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 o 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 o 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 as **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** estiver selecionada.

IMPORTANTE

Estes Tags estão disponíveis **SOMENTE** enquanto o Driver estiver 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 o Driver está atualmente conectado. Se o 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 o Driver estiver desconectado, este Tag configura o endereço IP ativo para a próxima tentativa de conexão.

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 o Driver não deve aceitar conexões externas, ou seja, o Driver se comporta como mestre, ou configure em Verdadeiro para habilitar a recepção de conexões, ou seja, o 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, o Driver tenta utilizar um endereço IP alternativo ou de *backup*. Configure em Falso para desabilitar sua utilização.

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

 Número da porta local a ser utilizada na conexão ao endereço IP alternativo ou de *backup* do 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* do 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 o 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. Exemplos:

- **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::ffff:ffff*): Aceita conexões de endereços IPv6 no intervalo entre fe80:03bf:0877:0000:0000:0000:0000 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 o 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 corresponder 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 aparecer 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*, onde o 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 o 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 pelo Driver para escutar conexões.

IO.Ethernet.MainIP

A Endereço IP do equipamento remoto. Pode-se utilizar tanto o endereço numérico como o nome de *host* do 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 do 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 no 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 do 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.SuppressEcho

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 o Driver enviou para o 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
1.0.30	22/12/2021	M. Salvador	<ul style="list-style-type: none"> Templates e mensagens agora podem começar como sendo Arrays (Case 31200) Criado suporte de apenas leitura para repetição de objetos dentro de arrays (case 31239) Publicação de mensagens agora funciona com mais de um valor de template. Permitidas escritas de bloco (case 31242)
1.0.22	13/05/2021	G. Beal	<ul style="list-style-type: none"> Corrigido o tratamento das palavras-chave TS_UNIX, TS_UNIX_TZ_SECONDS+ e TS_UNIX_TZ_SECONDS- em <i>Templates</i> no formato JSON (Case 30878).
1.0.20	01/04/2021	G. Beal	<ul style="list-style-type: none"> Corrigido um problema que atrasava a conexão inicial com um <i>Broker</i> (Case 30683).
1.0.18	01/03/2021	G. Beal	<ul style="list-style-type: none"> Incluída uma nova configuração de banda morta individual para cada Tag (Case 30517).
1.0.15	16/06/2020	M. Salvador	<ul style="list-style-type: none"> Adicionada a configuração de banda morta relativa (Deadband %), aplicada a todos os Tags numéricos (Case 28886).
1.0.14	04/05/2020	G. Beal	<ul style="list-style-type: none"> Implementada uma nova janela para edição dos <i>Templates</i> (Case 28650). Modificada a comparação entre mensagens e <i>Templates</i> em formato JSON para que a ordem dos pares de nome e valor dentro de um objeto não afete a correspondência (Case 28678).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.0.8	05/12/2019	M. Ludwig	<ul style="list-style-type: none"> • Adicionada uma interpretação sintática no formato JSON somente com nomes de atributos que sejam idênticos com <i>Template</i>. Atributos inexistentes são ignorados (<i>Case 26316</i>).
1.0.7	01/02/2019	M. Salvador	<ul style="list-style-type: none"> • Correções de problemas de desempenho e verificação de <i>time-out</i>. • Atualização de estampa de tempo quando um item não está associado a um <i>Template</i>. • Adicionado suporte a diversas referências para um mesmo item com <i>Templates</i> diferentes.
1.0.1	07/11/2018	M. Salvador	<ul style="list-style-type: none"> • 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/SI.
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

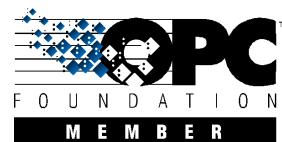
forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, 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 or fitness for a particular purpose.



Microsoft Partner
Gold Independent Software Vendor (ISV)