Driver MQTT

Nome do Arquivo	MQTT.dll
Fabricante	Message Queuing Telemetry Transport
Equipamentos	Clientes e Brokers compatíveis com as especificações 3.1, 3.1.1 e 5 do protocolo MQTT
Protocolo	MQTT sobre Ethernet TCP/IP e TLS/SSL
Versão	1.0.45
Última Atualização	28/09/2026
Plataforma	Win32
Dependências	IOKit versão 2.0 e biblioteca OpenSSL
Leitura com Superblocos	Não
Nível	31312

Introdução

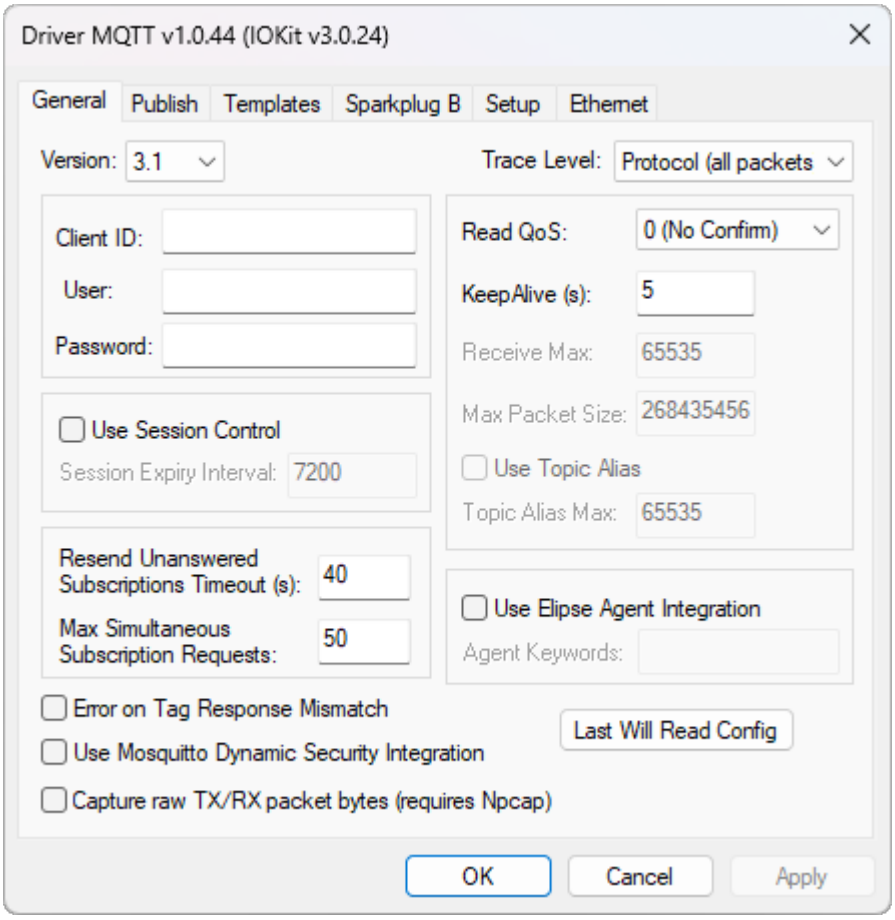
O Driver MQTT é um **Cliente** que implementa as versões **3.1**, **3.1.1** e **5** do protocolo MQTT padrão, ou **Vanilla**, com suporte adicional ao protocolo **Sparkplug B**, utilizando um *Broker* MQTT externo, não fornecido.

Este Driver é capaz de receber e extrair 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.



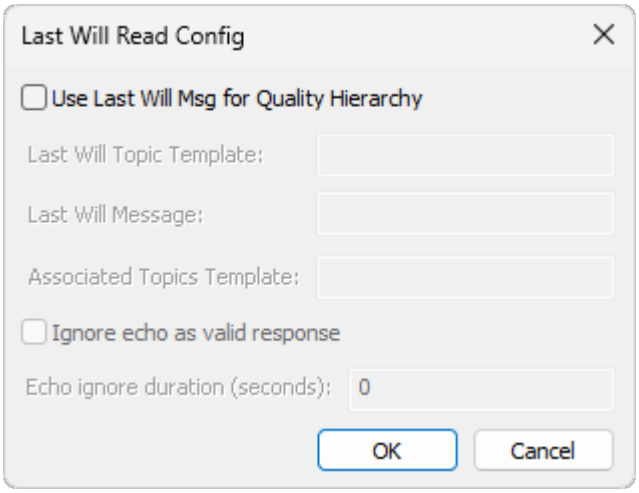
Aba General

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**, **3.1.1** ou **5**
Trace Level	Nível de detalhamento dos registros que este Driver grava no arquivo de log, incluindo os pacotes transmitidos e recebidos. As opções disponíveis são **Maximum (all + internals)**: registra todos os pacotes e também a atividade interna da biblioteca de comunicação, **Protocol (all packets)**: registra todos os pacotes MQTT transmitidos e recebidos (padrão), **Error only**: registra somente os erros ou **Fatal (minimal)**: registra somente os erros fatais. Estes registros não dependem da instalação de nenhuma ferramenta externa de captura de rede
Client ID	Nome que este **Cliente** utiliza como identificador para os demais **Clientes**. Não deve haver outro **Cliente** conectado em um mesmo *Broker* com este nome
User	Nome de usuário usado nas mensagens de conexão (**ConnAck**), caso desejado e permitido por um *Broker*
Password	Senha usada juntamente com o nome de usuário
Read QoS	Indica o código QoS (*Quality of Service*) usado para solicitar as assinaturas dos Tags. Os valores possíveis são **0**: No Confirm, **1**: Confirm ou **2**: Handshake. Um *Broker*

OPÇÃO	DESCRIÇÃO
	compara o código QoS solicitado e o código QoS originalmente publicado, enviando um dado no mínimo compatível
KeepAlive (s)	Tempo, em segundos, que um **Cliente** envia uma mensagem de **ping** no protocolo MQTT (**PingReq**) para verificar se a conexão com um *Broker* ainda está ativa. O valor padrão para esta opção é 5 (cinco) segundos
Receive Max	Disponível apenas na versão **5** do protocolo MQTT, define quantas mensagens QoS 1 (um) ou 2 (dois) podem ser recebidas de um *Broker* simultaneamente antes de serem confirmadas
Max Packet Size	Disponível apenas na versão **5** do protocolo MQTT, define o tamanho máximo, em bytes, para os pacotes MQTT
Use Topic Alias	Disponível apenas na versão **5** do protocolo MQTT. Ao selecionar esta opção, internamente é usado um identificador numérico que substitui um tópico longo, reduzindo o tamanho dos pacotes
Topic Alias Max	Disponível quando a opção **Use Topic Alias** está selecionada. Limita quantos tópicos podem ser substituídos por identificadores numéricos durante a comunicação
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 (um) e 2 (dois) enviadas por outros **Clientes** podem ser recuperadas
Session Expiry Interval	Disponível apenas na versão **5** do protocolo MQTT, define o tempo, em segundos, que uma sessão permanece armazenada em um *Broker* após a desconexão
Resend Unanswered Subscriptions Timeout (s)	Informa o número de segundos para reenviar uma assinatura de um item se este não recebeu o primeiro valor dentro do tempo informado
Max Simultaneous Subscription Requests	Número máximo de pedidos de assinaturas (*subscriptions*) requisitados a cada mensagem
Error on Tags Response Mismatch	Informa se este Driver deve retornar erro para Tags cuja mensagem não coincida com o *Template* configurado
Use Mosquitto Dynamic Security Integration	Habilita a integração do controle ou configuração das permissões do Mosquitto, através de mensagens padronizadas MQTT, disparadas através de Tags
Use Elipse Agent Integration	Habilita a integração com um Agente da **Elipse Software** para computadores com Windows
Last Will Read Config	Abre a janela Last Will Read Config, mostrada na figura a seguir, para configuração da mensagem de *Last Will* recebida por este Driver
Capture raw TX/RX packet bytes (requires Npcap)	Habilita a captura dos bytes brutos dos pacotes transmitidos e recebidos diretamente na interface de rede, o que exige a instalação da biblioteca **Npcap**. Esta opção está desabilitada por padrão e não é necessária para registrar os pacotes MQTT no arquivo de log, que já são registrados conforme a opção **Trace Level**



Janela Last Will Read Config

As opções disponíveis na janela Last Will Read Config estão descritas na tabela a seguir.

Opções disponíveis na janela Last Will Read Config

OPÇÃO	DESCRIÇÃO
Use Last Will Msg for Quality Hierarchy	Ao selecionar esta opção, todas as demais configurações desta janela são habilitadas para edição, exceto a opção **Echo ignore duration (seconds)**. Se este Driver recebe a mensagem configurada na opção **Last Will Message** em um tópico que corresponda ao valor definido na opção **Last Will Topic Template**, todos os tópicos que coincidem com o valor configurado na opção **Associated Topics Template** são marcados com a qualidade ruim 20, 24 ou 28 (inválida)
Last Will Topic Template	Modelo de tópico MQTT para a mensagem de *Last Will*
Last Will Message	Mensagem de *Last Will*, como por exemplo "Adeus"
Associated Topics Template	Modelo de tópicos MQTT associados à mensagem de *Last Will*, ou seja, os tópicos que são marcados com qualidade ruim
Ignore echo as valid response	Ao selecionar esta opção, a opção **Echo ignore duration (seconds)** é habilitada para edição
Echo ignore duration (seconds)	Essa opção define o período, em segundos, durante o qual uma mensagem recebida em um tópico é considerada um eco enviado por um *Broker* quando este Driver publica neste tópico. Assim, esta mensagem é desconsiderada como uma comunicação válida e os tópicos correspondentes ao valor da opção **Associated Topics Template** não são marcados com qualidade boa (192)

Os modelos de tópicos MQTT obedecem as especificações a seguir:

- Podem incluir campos fixos com nomes pré-definidos
- O caractere + (sinal de mais) é usado como curinga para representar qualquer tópico em um nível específico

- Campos variáveis são usados para conferência ou diferenciação entre mensagens de *Last Will*
- O caractere # (sinal de número) é usado como curinga para múltiplos níveis, utilizado exclusivamente no final de um tópico

Estas especificações estão em conformidade com o padrão MQTT, exceto pelos campos variáveis, que seguem o formato **<NOME_DA_VARIAVEL>**.

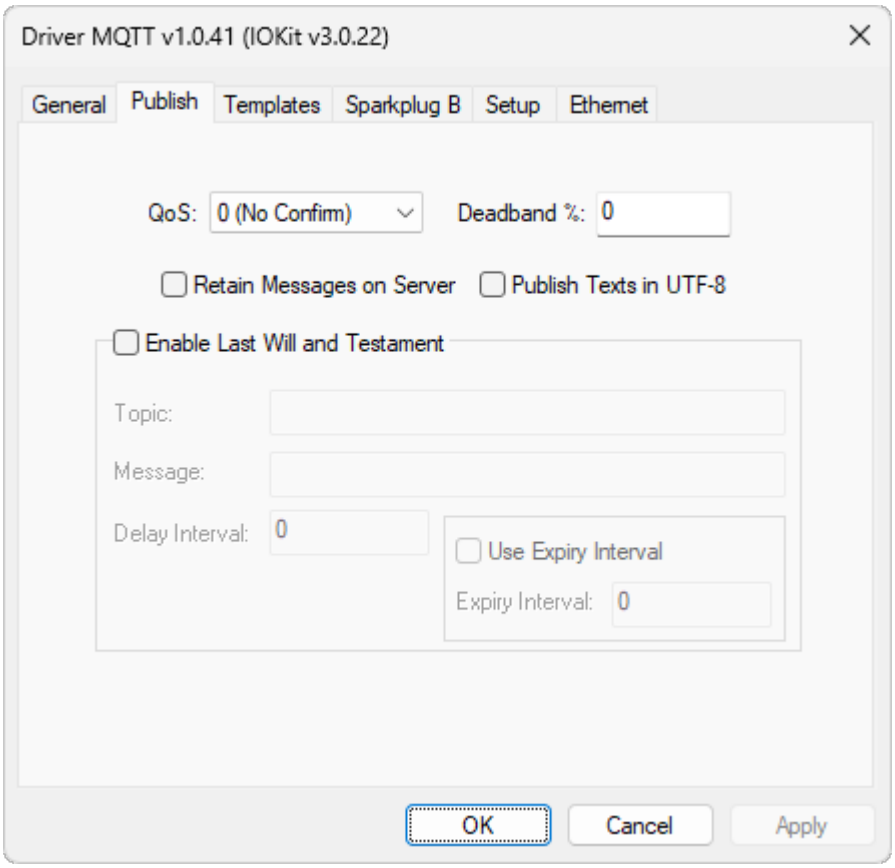
Por exemplo, considere uma estrutura de tópicos que segue o padrão *cidade*, *código da cidade*, *equipamento*, *identificador de equipamento* e, posteriormente, parâmetros correspondentes, como por exemplo "city/003/device/147/....".

Para configurar a opção **Last Will Topic Template**, pode-se usar o formato "city/<CITY_ID>/device/<DEVICE_ID>/lastwill". Já para a opção **Associated Topic Template**, pode-se usar o formato "city/<CITY_ID>/device/<DEVICE_ID>/#".

Desta forma, quando uma mensagem de *Last Will* é recebida, todos os Tags que seguem o formato com o mesmo parâmetro **<CITY_ID>** e **<DEVICE_ID>** da mensagem de *Last Will* recebida são marcadas com qualidade ruim. Caso existam outros campos no tópico que variem e não sejam **Strings** fixas, tais como *city* ou *device*, nem se repitam nos tópicos associados, tais como **<CITY_ID>** e **<DEVICE_ID>**, pode-se usar o caractere + (sinal de mais). Por exemplo, se os tópicos associados seguem o padrão "<NOME_DA_APLICACAO>/<CITY_ID>/<DEVICE_ID>", a opção **Associated Topic Template** pode ser configurada como "+/<CITY_ID>/<DEVICE_ID>/#".

Isto demonstra a flexibilidade do sistema, permitindo configurar como a mensagem de *Last Will* afeta os Tags e possibilitando a distinção entre diferentes mensagens de *Last Will*. É essencial que os modelos ou *Templates* sejam seguidos e que todos os campos variáveis, definidos no formato **<NOME_DA_VARIAVEL>** na opção **Last Will Topic Template**, sejam encontrados com os mesmos valores na opção **Associated Topic Template**.

Para que estes Tags retornem à qualidade boa, o valor 192, o tópico correspondente deve receber uma mensagem com um valor diferente daquele configurado na opção **Last Will Message**. No entanto, se o parâmetro *Item* do Tag inicia com um caractere ! (ponto de exclamação), este Tag não tem a qualidade marcada como ruim ao receber a mensagem de *Last Will* e também não retorna à qualidade boa ao receber um novo valor.



Aba Publish

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

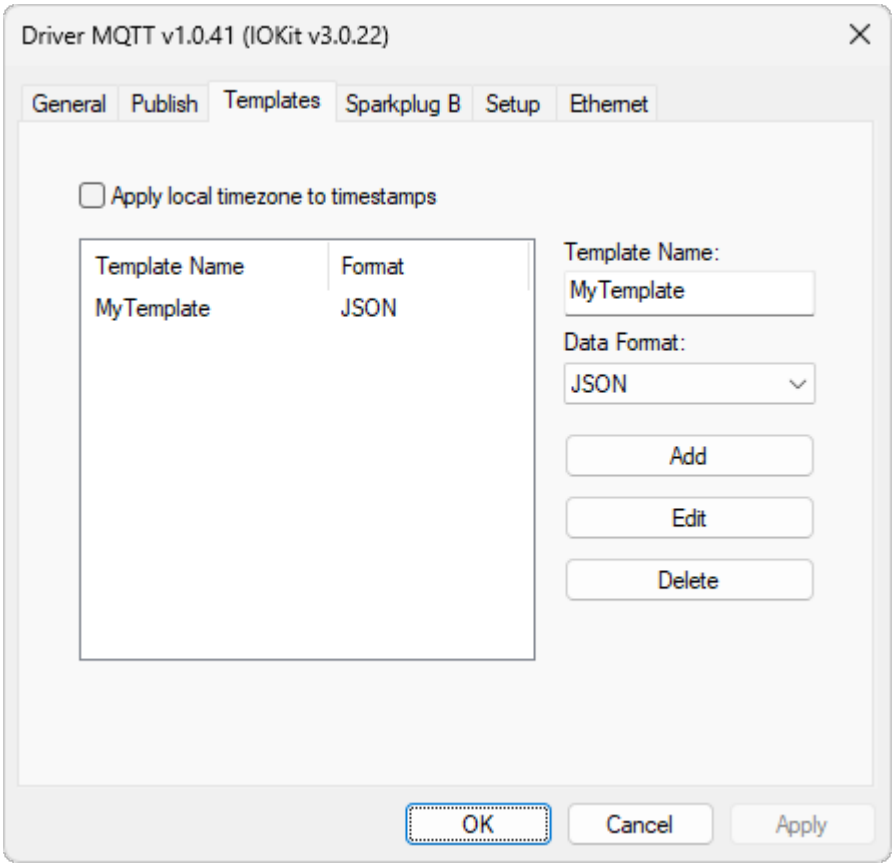
Opções disponíveis na aba Publish

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
Deadband %	Valor da banda morta relativa a ser aplicada em todos os Tags na escrita de valores numéricos sem *Templates*
Retain Messages on Server	Selecionar esta opção instrui um *Broker* a 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** se subscreve, a última mensagem de cada tópico, se houver, deve ser enviada a um **Cliente** por um *Broker*
Publish Texts in UTF-8	Informa se Tags do tipo **Texto** devem ter os valores convertidos para o formato **UTF-8** antes de serem escritos. Isto evita a interpretação errada por outros **Clientes** de caracteres com símbolos especiais
Enable Last Will and Testament	Ao habilitar esta opção, um servidor deve manter em memória um tópico e uma mensagem para um **Cliente**, que são enviados para todos os demais **Clientes** quando há uma desconexão não ordeira, ou seja, sem o envio da mensagem de *Disconnect*
Topic	Tópico ou endereço que é enviado na desconexão (*Last Will*), como por exemplo "ELIPSE\goodbye"
Message	Mensagem associada ao tópico, como por exemplo "Adeus"
Delay Interval	Disponível apenas na versão **5** do protocolo MQTT, define o tempo, em segundos, que um *Broker* espera após a desconexão inesperada antes de publicar a mensagem de *Last Will*
Use Expiry Interval	Disponível apenas na versão **5** do protocolo MQTT, habilita a opção **Expiry Interval**
Expiry Interval	Disponível apenas na versão **5** do protocolo MQTT, define o tempo, em segundos, após o qual a mensagem de *Last Will* é descartada por um *Broker* se ainda não foi entregue

NOTA

Para definir um valor diferente de **QoS**, **Retain** e **Expiry Interval** a cada Tag, é possível configurar um Tag com os parâmetros *N1* igual a 1 (um), *N2* igual a **QoS** (zero, um ou dois), *N3* igual a **Retain** (zero ou um) e *N4* igual a **Expiry Interval**, em segundos, se a versão do protocolo MQTT é igual a **5** (cinco).

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



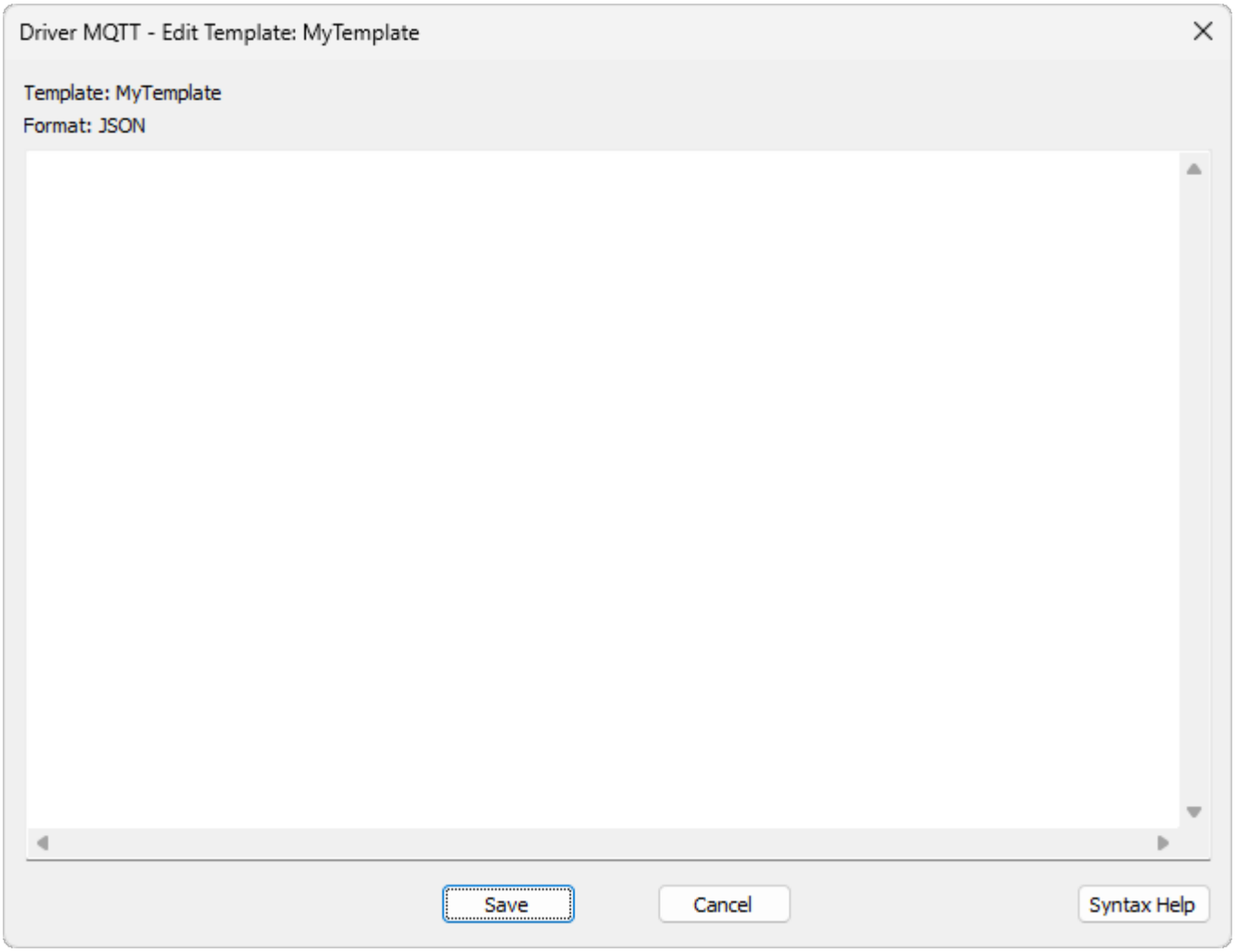
Aba Templates

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

Opções disponíveis na aba Templates

OPÇÃO	DESCRIÇÃO
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 de um *Template*. Não podem existir dois *Templates* com o mesmo nome
Data Format	Declara o formato da mensagem de um *Template*. 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 *Template* na lista de *Templates*
Edit	Abre uma janela para edição do *Template* selecionado
Delete	Apaga o *Template* selecionado da lista de *Templates*

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



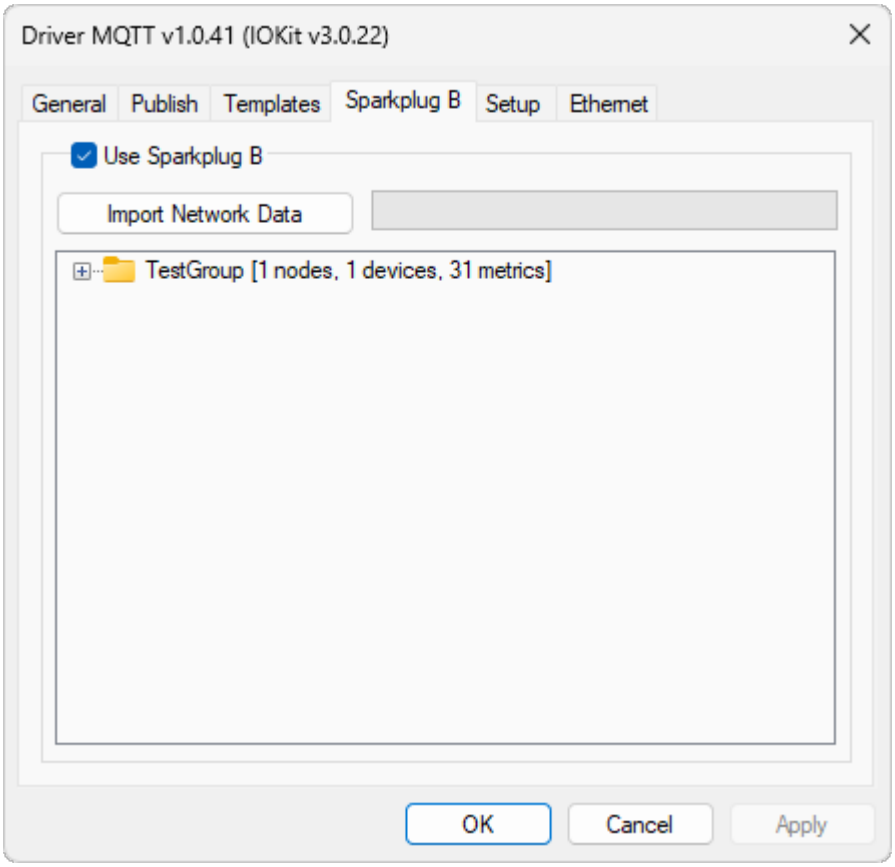
Janela para edição de um Template

As opções disponíveis na janela para edição de *Templates* estão descritas na tabela a seguir.

Opções disponíveis na janela para edição de Templates

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 de uma 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 **Sparkplug B** é possível habilitar este formato de comunicação via MQTT e importar os grupos, *nodes*, *devices* e métricas retidas em um *Broker*.



Aba Sparkplug B

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

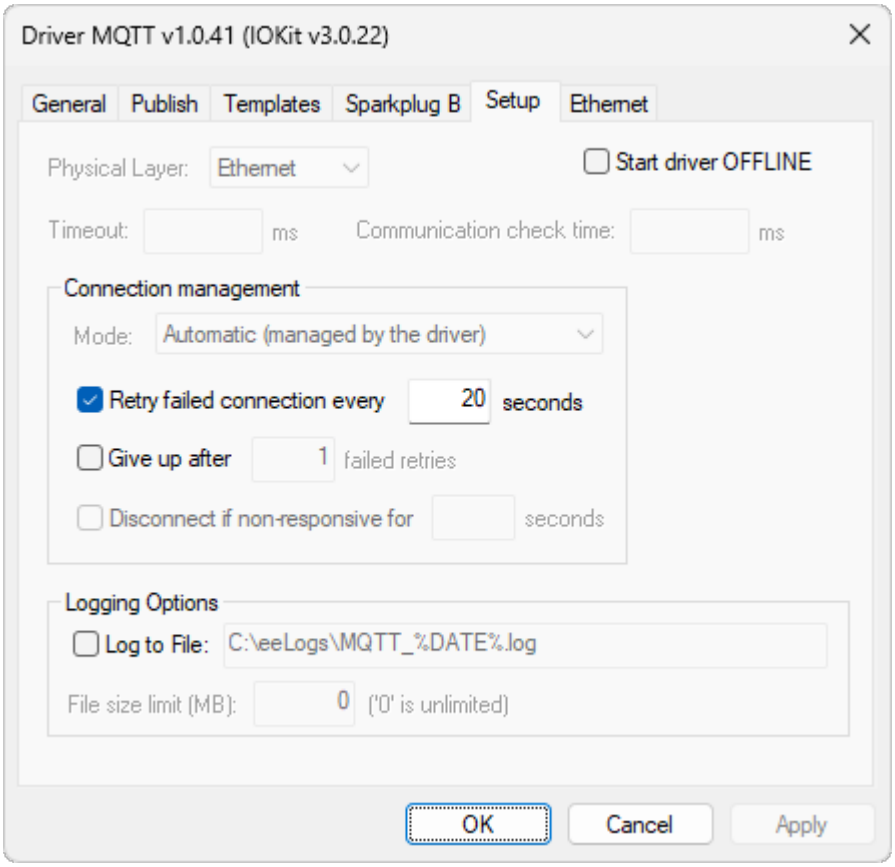
Opções disponíveis na aba Sparkplug B

OPÇÃO	DESCRIÇÃO
Use Sparkplug B	Selecione esta opção para que este Driver deixe de suportar o padrão **Vanilla** e passe a atuar como uma *Primary Application*, obedecendo a uma configuração distinta de Tags
Import Network Data	Descobre e importa os grupos, *nodes*, *devices* e métricas retidos em um *Broker*, montando uma árvore para visualização

NOTA

Para que esta importação funcione corretamente, as configurações de conexão devem estar preenchidas nas abas **General** e **Ethernet**. Esta importação cria automaticamente os Tags na janela **Tag Browser**.

A aba **Setup** contém a configuração geral deste Driver.



Aba Setup

As opções disponíveis para este Driver na aba **Setup** estão descritas na tabela a seguir.

NOTA

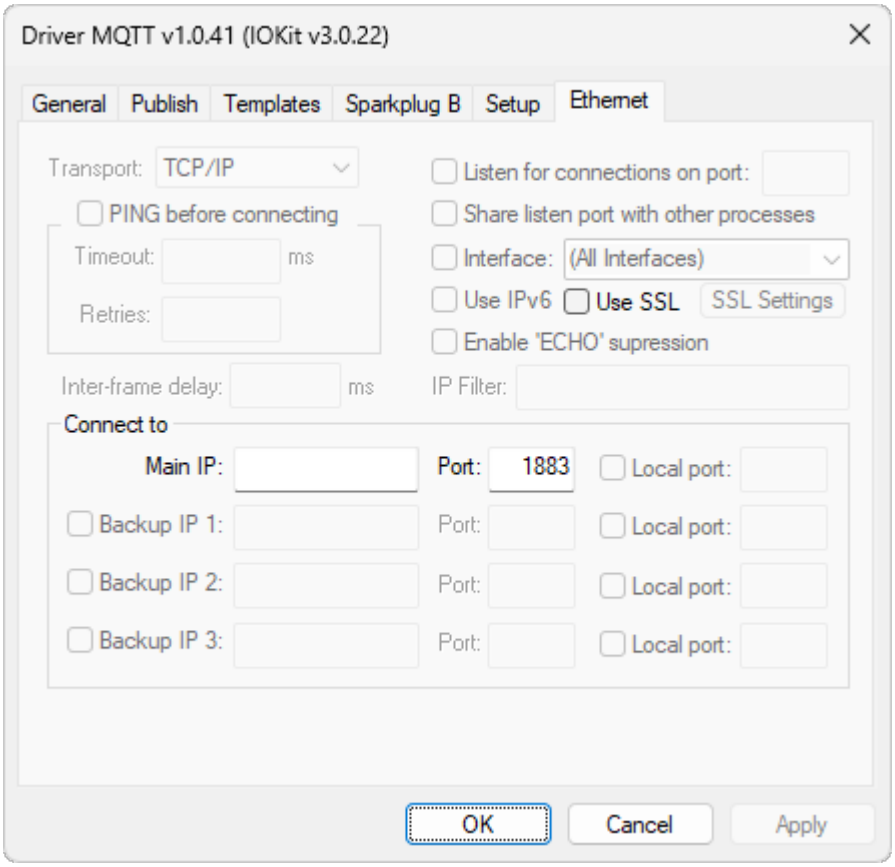
As opções da aba **Setup** que não estão descritas na tabela a seguir estão desativadas e não têm efeito na configuração deste Driver.

Opções disponíveis na aba Setup

OPÇÃO	DESCRIÇÃO
Start driver OFFLINE	Selecione esta opção para que este Driver inicie em modo **Offline** ou parado. Isto significa que a interface **Ethernet** não é criada até que se configure este Driver em modo **Online** utilizando-se um Tag em uma aplicação. Este modo possibilita a configuração dinâmica da interface **Ethernet** em tempo de execução
Retry failed connection every *n* seconds	Selecione esta opção para habilitar a retentativa de conexão deste Driver em um intervalo *n*, em segundos. Se a opção **Give up after *n* failed retries** não está selecionada, este Driver continua retentando até que uma conexão seja efetuada, ou que uma aplicação seja parada
Give up after *n* 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, este Driver vai para o modo **Offline**, assumindo que um problema de *hardware* foi detectado. Se este Driver estabelece uma conexão com sucesso, o número de

OPÇÃO	DESCRIÇÃO
	retentativas sem sucesso é zerado. Se esta nova conexão é perdida, então o contador de retentativas inicia do zero
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 efeito 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**, no **Elipse Power** ou no **Elipse Water**, permitindo assim que cada instância gere um arquivo separado de log. Por exemplo, ao configurar esta opção com o valor "c:\e3logs\drivers\mqtt_%PROCESS%.log", gera-se um arquivo c:\e3logs\drivers\mqtt_00000FDA.log para o processo **0FDAh**. Pode-se também utilizar a macro **%DATE%** no nome deste 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\mqtt_%DATE%.log", gera-se o arquivo c:\e3logs\drivers\mqtt_2005_12_31.log em 31/12/2025 e o arquivo c:\e3logs\drivers\mqtt_2026_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 de um arquivo de log, em megabytes. Um valor igual a 0 (zero) significa que não há limite de tamanho para um arquivo de log

A aba **Ethernet** permite configurar os parâmetros da interface **Ethernet**.



Aba Ethernet

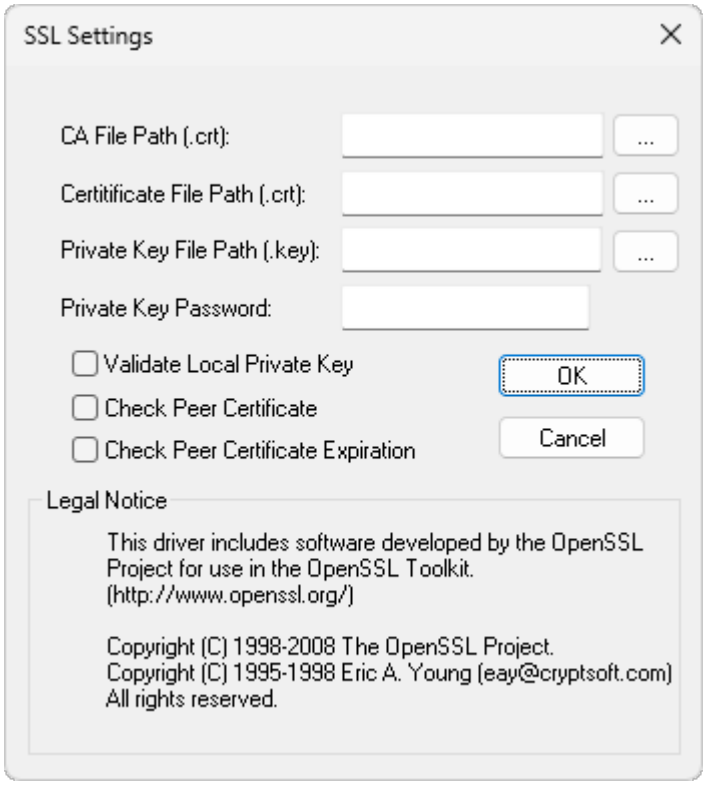
As opções disponíveis para este Driver na aba **Ethernet** estão descritas na tabela a seguir.

NOTA

As opções da aba **Ethernet** que não estão descritas na tabela a seguir estão desativadas e não têm efeito na configuração deste Driver.

Opções disponíveis na aba Ethernet

OPÇÃO	DESCRIÇÃO
Use SSL	Ao selecionar esta opção, o protocolo SSL (*Secure Sockets Layer*) é utilizado para uma conexão segura com um *Broker*. A opção **SSL Settings** é habilitada para configurar os certificados
SSL Settings	Abre a janela SSL Settings para configuração de certificados, mostrada na figura a seguir
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, este 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 TCP/IP de um dispositivo remoto, entre 0 (zero) e 65535



Janela SSL Settings

As opções disponíveis na janela SSL Settings estão descritas na tabela a seguir.

Opções disponíveis na janela SSL Settings

OPÇÃO	DESCRIÇÃO
CA File Path (.crt)	Nome e caminho de um arquivo para um Certificado de Autoridade (CA). Clique em ... para selecionar um arquivo
Certificate File Path (.crt)	Nome e caminho de um arquivo para um Certificado SSL de um **Cliente**. Clique em ... para selecionar um arquivo
Private Key File Path (.key)	Nome e caminho de um arquivo de chaves privadas para um Certificado SSL de um **Cliente**. Clique em ... para selecionar um arquivo
Private Key Password	Senha de uma chave privada, se aplicável
Validate Local Private Key	Habilita ou desabilita a validação do arquivo definido na opção **Private Key File Path (.key)**
Check Peer Certificate	Habilita ou desabilita a verificação de um Certificado SSL de um **Cliente**
Check Peer Certificate Expiration	Habilita ou desabilita a verificação da data de expiração de um Certificado SSL de um **Cliente**

Os Tags de Comunicação disponíveis para este Driver estão descritos a seguir.

IO.CommunicationStatus

Tag de Comunicação somente de leitura que informa o estado da comunicação deste Driver. Indica o funcionamento da comunicação em função do recebimento de dados válidos dentro de um período de tempo arbitrado na configuração. Os valores possíveis são **0 - Comunicação inativa**: Este Driver não recebeu dados válidos ou deixou de receber dados depois de *n* milissegundos, conforme configurado na aba **Setup**, ou **1 - Comunicação ativa**: Este Driver está recebendo dados válidos.

IO.PhysicalLayerStatus

Tag de Comunicação somente de leitura que indica o estado da camada física. Os valores possíveis são **0**: A camada física parada, ou seja, este Driver está em modo **Offline**, a camada física falhou ao inicializar ou excedeu o número máximo de tentativas de reconexão, **1**: A camada física foi iniciada mas não conectada, ou seja, este Driver está em modo **Online**, mas a camada física não está conectada ou **2**: A camada física está 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

Tag Bloco somente de leitura que permite modificar qualquer propriedade da configuração deste Driver em tempo de execução. Este Tag funciona somente enquanto este Driver está em modo **Offline**. Para isto, selecione a opção **Start driver OFFLINE** na aba **Setup**. As escritas de Elementos de Bloco individuais não são suportadas, um Bloco inteiro precisa ser escrito de uma vez só.

IO.WorkOnline

Tag de Comunicação de leitura e escrita que informa o estado atual deste Driver e permite iniciar ou parar a camada física. Os valores possíveis são **0 - Driver Offline**: A camada física está fechada ou parada. Este modo permite uma configuração dinâmica dos parâmetros deste Driver usando o Tag **IO.SetConfigurationParameters** ou **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**.

Sintaxe dos Templates

Para extrair o conteúdo de uma mensagem é necessário declarar um *Template*, que permite informar o formato de uma 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 de um 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, ou formato **UNIX**. Este valor pode ser numérico ou textual, e é usado como estampa de tempo de um 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

PALAVRA-CHAVE	DESCRIÇÃO
	usado como estampa de tempo de um Tag Bloco ou Tag de Comunicação, como por exemplo 43595.37373843. Esta palavra-chave só está disponível em *Templates* 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 de uma mensagem. Esta palavra-chave só está disponível em *Templates* do tipo **JSON**
QL_OPC	Qualidade, no padrão OPC DA, utilizando um byte. Este valor é usado diretamente como valor de qualidade em um Tag Bloco ou Tag de Comunicação, sem transformações. Consulte a tabela **Padrão OPC DA** para mais informações
QL_BOOL	Qualidade no padrão Booleano. Se este valor é maior que 0 (zero) ou a expressão "true" ou "TRUE", a qualidade é boa. Se este valor é igual a 0 (zero) ou a expressão "false" ou "FALSE", a qualidade é ruim
V1, V2, ... V50 Template JSON	Declara um ou mais valores, que podem ser extraídos individualmente para Tags de Comunicação, através da especificação de um terceiro parâmetro de endereçamento (*param*). 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 há apenas um valor **E3VAL** em uma mensagem, o *Template* também pode ser usado com um Tag de Cofmunicação
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* do tipo **JSON**, para indicar que os membros que se repetem dentro deste *array* devem ser lidos de forma independente por este Driver, retornando uma leitura para cada conjunto. Para mais informações, consulte o tópico **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

OPÇÃO	DESCRIÇÃO
%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 2 (dois) dígitos
%Y	Ano com 4 (quatro) dígitos
%Z	Nome do fuso horário, um código internacional que transforma um horário para **GMT**
%+	Deslocamento (*offset*) GMT no formato **+-HH:MM**

O código a seguir contém 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

OPÇÃO	DESCRIÇÃO
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 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. O código a seguir contém um 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}
```

O código a seguir contém um possível *Template* para extrair o conteúdo da mensagem anterior.

```
{"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 3 (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
}
```

O código a seguir contém um possível *Template* para o exemplo anterior.

```
{
  "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 mensagens quanto *Templates* devem ser objetos ou *arrays*, portanto devem começar com um caractere de chaves ({) ou colchetes ([), respectivamente.

Repetição de Valores

Os *arrays* podem ser usados para receber um número variável de elementos. Os códigos a seguir mostram exemplos de mensagens.

Mensagem 1 (um):

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

Mensagem 2 (dois):

```
{
  "MessageType": "Teste",
  "TagData": [
    {
      "Temperature": 11,
      "Humidity": 12
    },
   {
      "Temperature": 14,
      "Humidity": 66
    }
   {
      "Temperature": 70,
      "Humidity": 55
    }
  ]
}
```

Dentro do *array* é possível ter 1 (um), 2 (dois), 3 (três) 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, declare um *Template* com uma palavra-chave **Repeat_E3VAL** imediatamente após o colchete que marca o início do *array*, como no exemplo a seguir.

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

Desta forma, ao receber a mensagem 2 (dois), por exemplo, tem-se uma chamada do evento **OnRead** de um Tag Bloco para cada conjunto, conforme o exemplo a seguir.

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

Exemplo de 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
```

Ao declarar mais valores **E3VAL** antes da posição de repetição, estes valores são repetidos para cada conjunto. Como exemplo, para um *Template* declarado conforme o código a seguir.

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

Pode-se ter um Tag Bloco com 3 (três) Elementos, tipo, temperatura e umidade, e seriam recebidos os valores a seguir 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 serem extraídos de uma mensagem devem ser substituídos pelas palavras-chave. O código a seguir contém um exemplo de mensagem no formato **CSV**.

```
18/10/2016 22:30:45; 22BAC23OOP10; "Nível Alto";98,3;m
```

O código a seguir contém um exemplo de *Template* para este formato, para ser associado a um Tag Bloco com 4 (quatro) Elementos de Bloco.

```
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 de uma mensagem e substituir os campos em nós a serem transformados em valores pelas palavras-chave. O código a seguir contém um exemplo de uma mensagem no formato **XML**.

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

O código a seguir contém um possível *Template* para a mensagem anterior.

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

Configuração dos Tags

Esta seção contém a sintaxe de configuração de Tags no **Elipse E3**, no **Elipse Power** ou no **Elipse Water** quando utilizados os padrões **Vanilla MQTT** ou **MQTT Sparkplug B**.

Tags Vanilla MQTT

- **Item**: Este campo deve obedecer a sintaxe indicada a seguir

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

Ou

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

O caractere ! (ponto de exclamação) pode ser utilizado antes do **<Tópico>** para indicar uma exceção à opção **Associated Topic Template**.

Em que:

- **<Tópico>**: Endereço do item, ou tópico, no banco de dados de um **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, conforme o exemplo a seguir

```
"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 de um pedido todos os itens que contenham no nome o item atual ou os filhos do item atual. Por exemplo, para o valor "station12#" são retornados os valores a seguir.

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

No segundo caso devem ser incluídos na resposta de um pedido todos os itens no mesmo nível do item atual. Por exemplo, para o valor "station12/pump01/+" são retornados os valores a seguir.

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

- **<Template>**: Informa o nome de um *Template* usado como modelo de interpretação dos dados de uma mensagem, com o objetivo de mapear campos dentro desta 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*, se este Tag deve 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 uma mensagem lida ou escrita para o item é um Tag do tipo **Texto** (**String**) sem qualquer

ITEM	DESCRIÇÃO
	processamento
"station12/pump01;pumpdata"	Aplica-se o *Template* **pumpdata** ao conteúdo de **station12/pump01**. Se o *Template* descreve mais de um valor, deve-se utilizar um Tag Bloco em que cada Elemento de Bloco é um valor do *Template*
"station12/pump01;pumpdata;V1"	Caso idêntico ao anterior, porém recuperando somente o primeiro valor do *Template* e associando-o ao Tag, que não precisa ser um Tag Bloco. Neste caso o *Template* 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 deste Driver é ignorada para este Tag.

Para superar a banda morta absoluta, é preciso satisfazer a condição a seguir.

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

Para superar a banda morta relativa, é preciso satisfazer a condição a seguir.

```
|(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 deste 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 uma mensagem **1** que contém os valores **A**, **B** e **C** e uma mensagem **2** que 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 mensagens, configure este *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 um 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 este Driver publique mensagens para outros **Clientes**, realize a escrita do Tag desejado. Se um Tag não possui um *Template* associado, então o valor informado é enviado para o tópico. Neste caso, se é um valor numérico, este valor é convertido para texto. Se já é um texto, é enviado sem alteração.

Se um Tag possui um *Template* associado, então as variáveis declaradas dentro de um *Template* são preenchidas com os valores vindos deste Tag e depois enviados. Note que, ao usar um Tag simples, o *Template* deve ter apenas uma variável declarada. Se precisar de mais de uma variável, use um Tag Bloco, em que cada Elemento de Bloco corresponde a uma variável dentro deste *Template*, na mesma ordem.

Para executar a escrita de um Tag Bloco, é necessário que uma aplicação realize a escrita de um Tag Bloco inteiro, usando o método **Write** em um script. Para isto, primeiro configure o valor de cada um dos Elementos e depois escreva este Tag Bloco inteiro. As propriedades **AllowRead** e **AllowWrite** devem estar configuradas com o valor Falso. O código a seguir contém um exemplo de escrita de um *Template* contendo 3 (três) 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
```

Tags MQTT Sparkplug B

Recomenda-se utilizar os Tags disponíveis na janela **Tag Browser** montados automaticamente durante a importação. No entanto, caso alguma métrica não tenha sido descoberta ou seja necessário configurar manualmente, os Tags obedecem o padrão a seguir:

- **Item**: Este campo deve obedecer a sintaxe indicada a seguir

```
<Grupo>/<Node>;<Métrica>
```

Ou

```
<Grupo>/<Node>/<Device>;<Métrica>
```

Em que:

- **<Grupo>**: Identificador de um grupo **Sparkplug** (*Group ID*), que agrupa logicamente os nós em um *Broker* MQTT
- **<Node>**: Identificador de um nó **Sparkplug** (*Node ID*), que representa um dispositivo ou *gateway* que publica ou subscreve dados em um grupo

- **<Device>**: Identificador de um dispositivo **Sparkplug** (*Device ID*), usado opcionalmente para representar um equipamento específico conectado a um nó
- **<Métrica>**: Nome da métrica, ou propriedade, **Sparkplug** lida ou escrita, conforme definido no *payload* do protocolo

NOTA

Os valores dos Tags são atualizados por meio de *callbacks* ao serem recebidos pelos tópicos **NDATA** ou **DDATA** do padrão **Sparkplug B**. Ao realizar uma escrita em um Tag, o conteúdo é analisado, ou *parsed*, convertido para o formato esperado pelo destinatário e, em seguida, enviado via comando **NCMD** (*Node Command*) ou **DCMD** (*Device Command*).

Eclipse Distribution License Versão 1.0

This project includes Paho MQTT C Client DLLs, distributed under the Eclipse Distribution License v1.0.

Eclipse Distribution License - v1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.0.45	28/09/2026	M. Ludwig	• Criada a opção **Trace Level** na aba **General**, que define o nível de detalhamento dos registros no arquivo de log deste Driver, incluindo os pacotes transmitidos e recebidos, sem depender de nenhuma ferramenta

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			externa de captura de rede (*Case 40088*). • A captura dos bytes brutos dos pacotes transmitidos e recebidos, que exige a instalação da biblioteca **Npcap**, passou a ser opcional e está desabilitada por padrão na opção **Capture raw TX/RX packet bytes (requires Npcap)** da aba **General** (*Case 40088*). • Corrigido um problema em que a estampa de tempo (**TS_TEXT**) de leituras dentro de uma palavra-chave **Repeat_E3VAL** pode ser gravada incorretamente, especialmente em mensagens com múltiplas repetições (*Case 40694*).
1.0.43	15/06/2026	A. Fetzner	• Corrigido um GPF (*General Protection Failure*) intermitente no processamento de mensagens no formato **JSON** (*Case 39692*).
1.0.42	24/03/2026	A. Fetzner	• Corrigida uma falha de conexão SSL/TLS ao utilizar uma chave privada com senha (*Case 39619*).
1.0.41	11/11/2025	A. Fetzner	• Implementado suporte ao padrão **Sparkplug B** atuando como *Primary Application* (*Case 30445*).
1.0.40	03/09/2025	A. Fetzner M. Ludwig	• Corrigido um alto consumo de CPU que ocorria quando a conexão com um *Broker* falhava ou era interrompida (*Case 38571*). • Driver atualizado para a biblioteca **IOKit** versão **3.0** e Visual Studio 2022 (*Case 37988*).
1.0.39	14/05/2025	A. Fetzner	• Atualizada a janela de propriedades deste Driver e ajustes nas configurações de reconexão com um *Broker* (*Case 37814*). • Corrigido um GPF (*General Protection Failure*) que ocorria quando um *Broker*

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			iniciava uma desconexão (*Case 37794*). • Adicionadas novas configurações às mensagens de *Last Will* recebidas por este Driver, possibilitando a individualização por meio de modelos ou *Templates* próprios (*Case 37678*).
1.0.38	02/04/2025	A. Fetzner	• Corrigido um vazamento de memória durante a publicação de dados (*Case 37217*). • Implementado o suporte à versão **5** do protocolo MQTT. O protocolo OpenSSL foi atualizado para a versão **3** e foi corrigido um erro de conexão com *Brokers* HiveMQ (*Case 32094*).
1.0.37	20/01/2025	M. Salvador	• Criada uma opção na aba **Publish** para definir se textos (**Strings**) são enviados no formato **UTF-8**. A mesma opção é usada na recepção para processar as **Strings** recebidas do formato **UTF-8** para o formato **ASCII**, mantendo a coerência entre o envio e a recepção (*Case 34202*). • Criada uma opção para retornar erro para Tags quando uma mensagem não coincide com o *Template* configurado (*Case 34301*). • Agora, ao escrever em um Tag, este Driver realiza o *parsing* com base apenas no *Template* referenciado no próprio Tag, não em outros *Templates* que possam estar referenciados para o mesmo tópico (*Case 34797*). • Criada uma opção **Return Error for Tags With Response Mismatch** no caso em que uma mensagem recebida não

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			esteja na mesma ordem que o *Template* associado ao Tag (*Case 35658*). • Criada uma opção através dos parâmetros *N* para informar o valor das propriedades **QoS** e **Retain** por Tag (*Case 35659*). • Corrigida uma falha geral ao utilizar o protocolo SSL recebendo uma mensagem muito grande (*Case 37042*).
1.0.30	22/12/2021	M. Salvador	• *Templates* e mensagens agora podem iniciar como *arrays* (*Case 31200*). • Adicionado suporte para repetição de objetos dentro de *arrays* (*Case 31239*). • A publicação de mensagens agora funciona com mais de um valor de *Template* e permitidas escritas em bloco (*Case 31242*).
1.0.22	13/05/2021	G. Beal	• Corrigido o tratamento das palavras-chave **TS_UNIX**, **TS_UNIX_TZ_SECONDS+** e **TS_UNIX_TZ_SECONDS-** em *Templates* no formato **JSON** (*Case 30878*).
1.0.20	01/04/2021	G. Beal	• Corrigido um problema que atrasava a conexão inicial com um *Broker* (*Case 30683*).
1.0.18	01/03/2021	G. Beal	• Incluída uma nova configuração de banda morta individual para cada Tag (*Case 30517*).
1.0.15	16/06/2020	M. Salvador	• Adicionada a configuração de banda morta relativa (**Deadband %**), aplicada a todos os Tags numéricos (*Case 28886*).
1.0.14	04/05/2020	G. Beal	• Implementada uma nova janela para edição de *Templates* (*Case 28650*). • Modificada a comparação entre mensagens e *Templates* em formato **JSON** para que a ordem dos pares de nome e valor dentro de um objeto não

VERSÃO	DATA	AUTOR	COMENTÁRIOS
			afete a correspondência (*Case 28678*).
1.0.8	05/12/2019	M. Ludwig	• Adicionada uma interpretação sintática no formato **JSON** somente com nomes de atributos que sejam idênticos com *Template*. Atributos inexistentes são ignorados (*Case 26316*).
1.0.7	01/02/2019	M. Salvador	• Correções de problemas de desempenho e verificação de *time-out*. • Atualização de estampa de tempo quando um item não está associado a um *Template*. • Adicionado suporte a diversas referências para um mesmo item com *Templates* diferentes.
1.0.1	07/11/2018	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 © 2016-2026 Elipse Software Ltda. Todos os direitos reservados.