

Driver Elipse VBScript

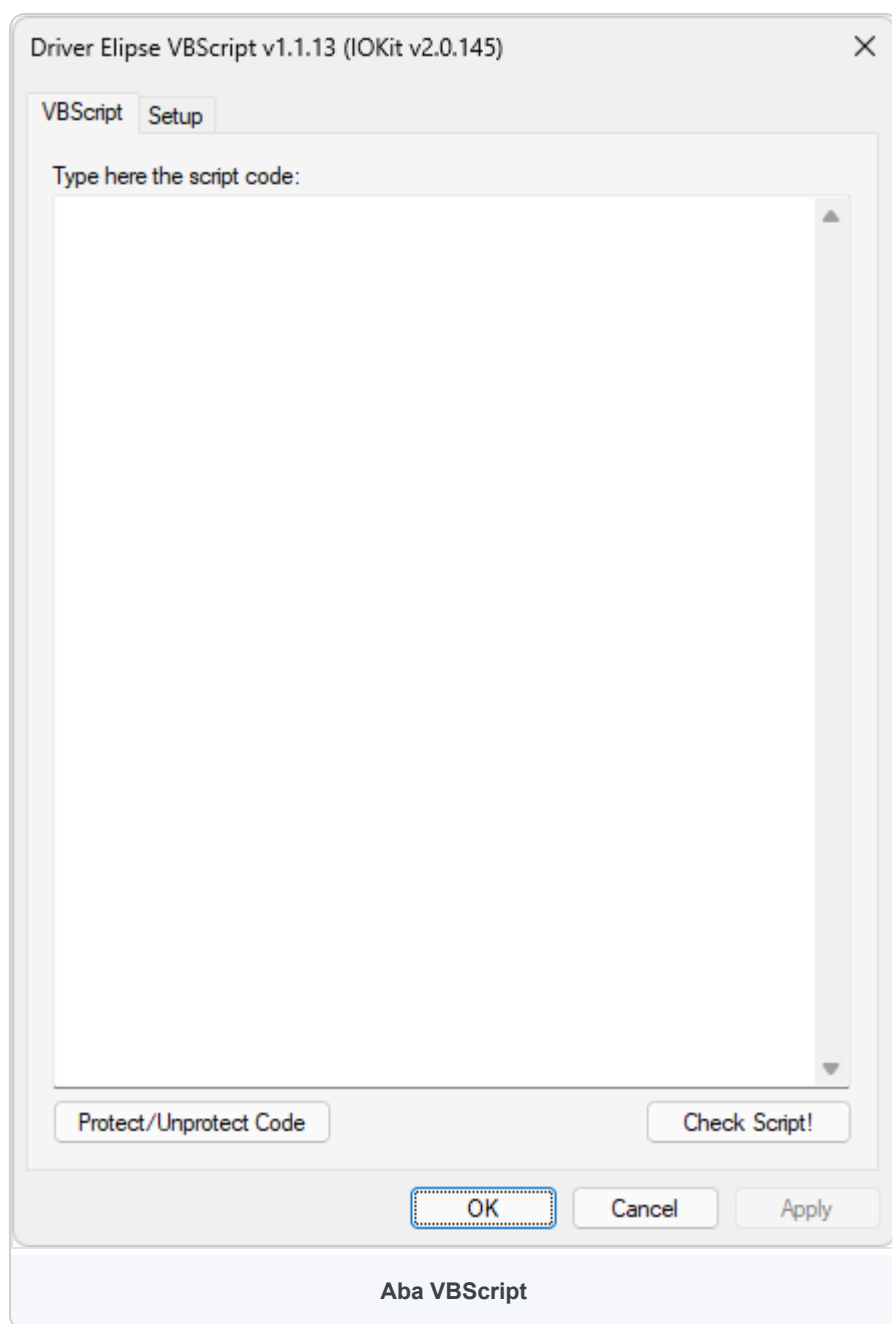
Nome do Arquivo	VBScriptDriver.dll
Fabricante	Elipse Software
Equipamentos	Não aplicável
Protocolo	Não aplicável
Versão	1.1.16
Última Atualização	10/08/2026
Plataforma	Win32
Dependências	Nenhuma
Leitura com Superblocos	Não
Nível	0

Introdução

Este Driver permite que um usuário implemente um script em linguagem VBScript, que é executado a cada leitura ou escrita de um Tag. Este Driver não utiliza a camada de comunicação da biblioteca **IOKit**.

Configuração do Driver

O usuário é responsável por editar o script que é executado por este Driver na aba **VBScript**, mostrada na figura a seguir.



Clique em **Check Script!** para compilar o código informado na opção **Type here the script code**. Se há algum erro neste script, uma caixa de diálogo é mostrada indicando a linha em que este erro ocorre.

Clique em **Protect/Unprotect Code** para criptografar um código ainda não protegido através de uma senha. Uma vez protegido, este código não pode mais ser visualizado ou editado através desta aba, mas os scripts podem ser executados normalmente em tempo de execução. Para voltar a visualizar ou editar um código, clique novamente em **Protect/Unprotect Code** e digite a senha usada para criptografar aquele código.

Eventos OnStart e OnStop

Quando este Driver inicia, o script do evento **OnStart** é executado. Quando este Driver para, o script do evento **OnStop** é executado, conforme o exemplo a seguir.

```
Sub OnStart()  
    'Executado quando este Driver inicia  
End Sub  
  
Sub OnStop()  
    'Executado quando este Driver para  
End Sub
```

Variáveis Globais

Para armazenar algum valor temporário durante a execução deste Driver, é possível declarar variáveis globais em um script, conforme o exemplo a seguir.

```
' Declara uma variável global  
Dim gDict  
  
Sub OnStart()  
    'Inicializa as variáveis globais  
    Set gDict = CreateObject("Scripting.Dictionary")  
    gDict.Add "BR", "Português"  
    gDict.Add "US", "Inglês"  
End Sub  
  
Sub OnStop()  
    'Limpa as variáveis globais  
    Set gDict = Nothing  
End Sub
```

Lendo e Escrevendo Tags

Este Driver identifica cada Tag pela propriedade **ParamItem**, ou seja, a coluna **Item** no editor de Tags do **E3**, **Power** ou **Water**.

Por exemplo, se a propriedade **ParamItem** é igual a "Tag", então a função de leitura deste Tag é **OnRead_Tag** e a função de escrita é **OnWrite_Tag**. Estas funções devem ser implementadas em um script, conforme o exemplo a seguir.

```
Dim gTag

Sub OnRead_Tag(r, v)
    v.Value = gTag
End Sub

Sub OnWrite_Tag(r, v)
    gTag = v.Value
End Sub
```

Se o usuário não implementa as funções **OnRead_xxx** ou **OnWrite_xxx** de um determinado Tag, então este Driver executa as funções genéricas **OnRead** e **OnWrite**, conforme o exemplo a seguir.

```
Dim gTag

Sub OnRead_Tag(r, v)
    v.Value = gTag
End Sub

Sub OnWrite_Tag(r, v)
    gTag = v.Value
End Sub

'Retorna uma String de erro
' nas leituras dos demais Tags
Sub OnRead(r, v)
    v.Value = "ERRO!"
    v.Quality = 20
End Sub
```

Estas funções de leitura e escrita possuem os parâmetros descritos a seguir.

- **r (*Request Object*)**: Identifica esta requisição e possui as propriedades descritas na tabela a seguir

Propriedades do parâmetro Request Object

PROPRIEDADE	DESCRIÇÃO
r.ParamDevice	Propriedade ParamDevice de um Tag, com o tipo de dados String
r.ParamItem	Propriedade ParamItem de um Tag, com o tipo de dados String
r.N1	Propriedade N1 de um Tag, com o tipo de dados Long
r.N2	Propriedade N2 de um Tag, com o tipo de dados Long
r.N3	Propriedade N3 de um Tag, com o tipo de dados Long

PROPRIEDADE	DESCRIÇÃO
r.N4	Propriedade N4 de um Tag, com o tipo de dados Long
r.Size	Indica o tamanho de um Tag envolvido na leitura ou escrita atual. Para mais informações, consulte a tabela Propriedades r.Size, r.Count e r.Offset
r.Count	Indica a quantidade de Elementos envolvidos na leitura ou escrita atual. Para mais informações, consulte a tabela Propriedades r.Size, r.Count e r.Offset
r.Offset	Indica, a partir do primeiro Elemento ou valor 0 (zero), o índice do enésimo Elemento envolvido em uma operação. Para mais informações, consulte a tabela Propriedades r.Size, r.Count e r.Offset
r.FailRequest()	Esta função deve ser utilizada para provocar uma falha na leitura ou escrita atual
r.IsSucceeded()	Esta função indica se a requisição atual foi bem-sucedida ou falhou, até o momento

Propriedades r.Size, r.Count e r.Offset

LEITURA OU ESCRITA	R.SIZE	R.COUNT	R.OFFSET
Tag PLC (um Elemento)	1 (um)	1 (um)	0 (zero)
Tag Bloco (n Elementos)	Tamanho de um Tag Bloco	Tamanho de um Tag Bloco	0 (zero)
Tag Elemento de Bloco (um Elemento)	Tamanho de um Tag Bloco	1 (um)	Índice de um Elemento de um Tag Bloco. Os valores possíveis são Elemento 1 : <i>Offset</i> é igual a 0 (zero), Elemento 2 : <i>Offset</i> é igual a 1 (um), Elemento 3 : <i>Offset</i> é igual a 2 (dois) e Elemento N : <i>Offset</i> é igual a $N - 1$

- **v (Value Object)**: Contém o valor, a qualidade e a estampa de tempo de um Tag. Possui as propriedades descritas na tabela a seguir, que permitem a escrita e a leitura

Propriedades do parâmetro Value Object

PROPRIEDADE	DESCRIÇÃO
v.Value	Valor de um Tag, com o tipo de dados Variant . NOTA: Qualquer atribuição de valor ajusta a qualidade para Boa (192) e a estampa de tempo para a hora atual do computador
v.Timestamp	Estampa de tempo do Tag, com o tipo de dados Date
v.Quality	Qualidade de um Tag no padrão OPC, variando entre 0 (zero) e 255, com o tipo de dados Byte . Na leitura, esta propriedade já vem com qualidade Boa (192). Para indicar erros de leitura, configure a qualidade para Ruim , como por exemplo um valor igual a 20. NOTA: A qualidade e a estampa de tempo devem ser atribuídas depois do valor
v.DimAsList()	Dimensiona este valor como uma lista de valores. Isto permite que uma função de leitura retorne 0 (zero) ou mais valores para um mesmo Tag em uma mesma operação
v.AddToList(Variant)	Adiciona um valor ao final da lista de valores a serem retornados para um Tag

O código a seguir contém um exemplo de leitura de uma lista de valores.

```
Sub OnRead_List(r, v)
    v.DimAsList()
    v.AddToList("List")
    v.AddToList("of")
    v.AddToList("values")
End Sub
```

O código a seguir contém um exemplo de leitura com acesso a Elementos de um Tag Bloco. Neste exemplo o Tag Bloco tem 5 (cinco) Elementos.

```
Sub OnRead_Tag(r, v)
    v.value = Array(1,2,25,4,33)
End Sub
```

O código a seguir contém um exemplo de escrita com acesso a Elementos de um Tag Bloco. Neste exemplo o Tag Bloco tem 5 (cinco) Elementos.

```
Sub OnWrite_Tag(r, v)
    v.value = Array(1,2,25,4,33)
End Sub
```

Funções Globais

Além das funções pré-definidas da linguagem VBScript, como por exemplo **CreateObject**, **Chr**, **CLng** e **Replace**, entre outras, este Driver define as funções de uso global descritas na tabela a seguir.

Funções globais

FUNÇÃO	DESCRIÇÃO
<code>NewValue(Value, Quality, TimeStamp)</code>	Cria um objeto do tipo Value (<i>Value Object</i>), que possui um valor, uma qualidade e uma estampa de tempo. Os parâmetros <i>Quality</i> e <i>TimeStamp</i> são opcionais e, se não especificados, assumem os valores 192 e Now() , respectivamente
<code>Trace(String)</code>	Escreve uma mensagem no arquivo de log deste Driver
<code>CreateDotNetObject(String, String)</code>	Cria um objeto do tipo .NET . O primeiro parâmetro corresponde ao caminho completo do <i>assembly</i> .NET e o segundo parâmetro se refere ao nome do tipo presente neste <i>assembly</i> , incluindo o <i>Namespace</i> , separados por um ponto, como por exemplo "Namespace.Class". O retorno desta função é o objeto criado. NOTA: O <i>assembly</i> .NET criado deve estar configurado para se comunicar via interface COM, ou seja, com o atributo ComVisible habilitado. A função CreateDotNetObject suporta <i>assemblies</i> compilados no <i>framework</i> .NET versão 3.5 ou anterior

O código a seguir contém um exemplo de uso das funções globais.

```
Dim obj

Sub OnStart()
    Set obj = CreateDotNetObject("C:\MyAssembly.dll", "MyNamespace.MyClass")
    obj.MyClassMethod()
End Sub

Sub OnRead_List(r, v)
    v.DimAsList()

    Dim tStamp
    tStamp = Now() - 2 'Anteontem

    Trace "Retornando uma lista de valores"
    v.AddToList(NewValue("Ontem", 192, Now() - 1))
    v.AddToList(NewValue("Hoje"))
    v.AddToList(NewValue("Amanhã", 192, Now() + 1))
End Sub

Sub OnStop()
    Set obj = Nothing
End Sub
```

Documentação das Interfaces de Comunicação

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

Esta documentação é compartilhada e mantida no manual do IOKit:

- Configurações de um Driver
- Configurações Gerais
- Configuração de Estatísticas
- Configuração da Interface Ethernet
- Configuração da Interface Modem
- Configuração da Interface RAS
- Configuração da Interface Serial

Histórico de Revisões do Driver

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.16	10/08/2026	C. Mello	• Ajustes para executar corretamente o conteúdo de um código em VBScript definido pelo usuário quando protegido por senha (<i>Case 39496</i>).
1.1.14	16/09/2025	M. Ludwig	• Driver atualizado para a biblioteca IOKit versão 3 e Visual Studio 2022 (<i>Case 37969</i>).
1.1.13	16/12/2024	F. Englert	• Ajustes adicionais para prevenir problemas na manipulação das propriedades ParamDevice e ParamItem das Tags (<i>Case 36703</i>).
1.1.8	30/09/2024	F. Englert	• Adicionadas as propriedades Count e Offset no objeto Request (<i>Case 36703</i>).
1.1.7	18/01/2024	M. Salvador	• Atualizações na documentação do VBScriptDriver (<i>Case 34408</i>).
1.1.6	18/01/2024	M. Salvador	• Criada uma opção para proteger um código contra visualização ou edição, mas que permite a execução deste código mesmo protegido (<i>Case 34408</i>).
1.1.5	30/03/2023	M. Ludwig	• Corrigido um problema esporádico de Strings vazias (<i>Case 34049</i>).
1.1.4	31/07/2019	M. Ludwig	• Driver portado para o Visual Studio 2017 (<i>Case 27083</i>).
1.1.3	16/08/2013	G. Taschetto	• Ajustes finais de compatibilidade com a biblioteca IOKit versão 2 (<i>Case 13504</i>).
1.1.2	10/06/2013	G. Taschetto	• Ajustes de compatibilidade com os "Logs" da biblioteca IOKit versão 2 (<i>Case 14226</i>).

VERSÃO	DATA	AUTOR	COMENTÁRIOS
1.1.1	28/05/2013	G. Taschetto	• Correções na documentação deste Driver (<i>Case 13524</i>). • Driver atualizado para a biblioteca IOKit versão 2 (<i>Case 13504</i>).
1.0.1	18/03/2010	M. Zani	• Adicionada a função CreateDotNetObject (<i>Case 10285</i>).
0.02 (Beta)	10/03/2009	F. Englert	• Versão inicial deste Driver (para testes).