



# Random

v1.0.12 · Português

# Conteúdo

Driver ELIPSE Random

Propriedades do Driver

Configuração dos Tags

Arquivo de Configuração no Formato XML

- Perfis

- Categorias

Adicionando Simulações em Aplicações Existentes

Documentação das Interfaces de Comunicação

Histórico de Revisões do Driver

# Driver ELIPSE Random

---

|                         |                       |
|-------------------------|-----------------------|
| Nome do Arquivo         | Random.dll            |
| Fabricante              | Eclipse Software Ltda |
| Equipamentos            |                       |
| Protocolo               |                       |
| Versão                  | 1.0.12                |
| Última Atualização      | 12/06/2025            |
| Plataforma              | Win32                 |
| Dependências            | Nenhuma               |
| Leitura com Superblocos | Não                   |
| Nível                   | 0                     |

## Introdução

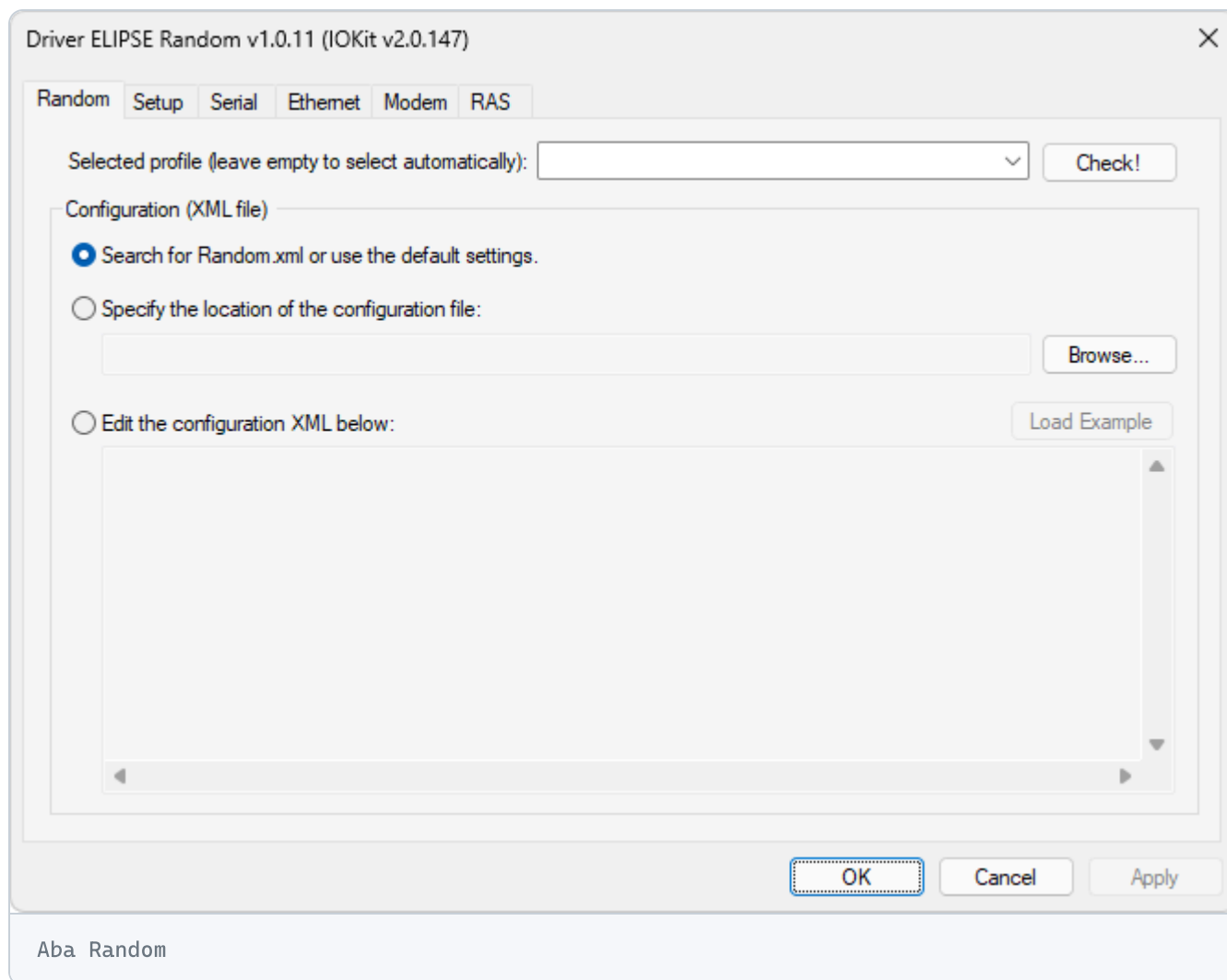
O Driver ELIPSE Random retorna valores simulados para os Tags de uma aplicação, permitindo que sejam configurados diferentes Perfis de simulação, de acordo com os parâmetros de cada Tag.

---

## Propriedades do Driver

---

Este Driver não utiliza os parâmetros **[P]**. Pode-se configurar qualquer meio físico para uso com este Driver, tais como **Serial** ou **Ethernet**, mas em tempo de execução este Driver sempre troca o meio físico para **None**.



## Seleção do Perfil Ativo

A configuração deste Driver pode conter diversos Perfis. Selecione na opção **Selected profile** o nome do Perfil ativo, ou deixe esta opção em branco para que este Driver selecione um Perfil automaticamente, de acordo com o nome da biblioteca deste Driver, ou seja, o nome do arquivo com a extensão .dll. Clique em **Check** para carregar e validar a configuração atual deste Driver.

## Busca Automática pelo Arquivo Random.xml

O modo padrão de funcionamento deste Driver é procurar pelo arquivo Random.xml na pasta atual ou nas pastas superiores. Para manter o comportamento padrão, selecione a opção **Search for Random.xml or use the default settings**.

Neste modo, este Driver procura pelo arquivo Random.xml na mesma pasta em que se encontra o arquivo .dll deste Driver. Se não encontrar este arquivo nesta pasta, a procura se repete nas pastas superiores até a pasta raiz. Por exemplo, se o arquivo Random.dll se encontra na pasta **D:\Supervisório\Elipse\Usina1**, este Driver procura pelos seguintes arquivos, nesta ordem:

- D:\Supervisório\Elipse\Usina1\Random.xml
- D:\Supervisório\Elipse\Random.xml
- D:\Supervisório\Random.xml
- D:\Random.xml

Se o arquivo no formato **XML** não é encontrado, este Driver assume a configuração padrão compatível com o Driver ELIPSE Random original.

## Informando um Nome ou Caminho Diferente para o Arquivo XML

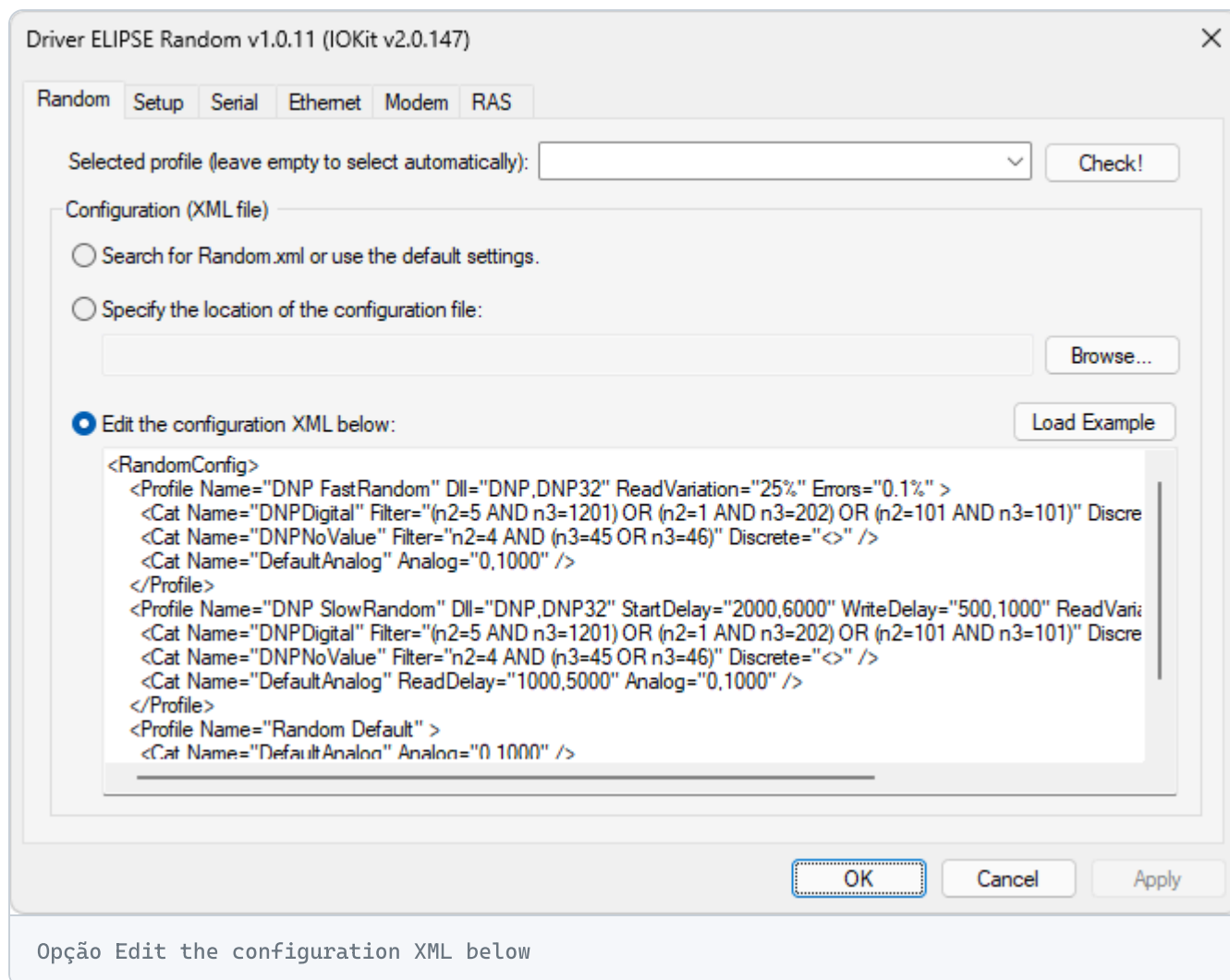
Para informar um nome ou um caminho alternativo para o arquivo de configuração no formato **XML**, selecione a opção **Specify the location of the configuration file**.

Se um caminho absoluto é informado, como por exemplo **C:\Supervisório\MyConfig.xml**, então este Driver procura o arquivo no formato **XML** apenas na pasta indicada.

Se um caminho relativo é informado, como por exemplo **MyConfig.xml** ou **Drivers\MyConfig.xml**, então este Driver procura o arquivo no formato **XML** na pasta do arquivo com a extensão .dll ou nas pastas superiores se não foi encontrado.

## Informando o Arquivo XML Diretamente na Janela de Propriedades

Para editar o conteúdo de um arquivo de configuração no formato **XML** diretamente na janela de propriedades deste Driver, selecione a opção **Edit the configuration XML below**, mostrada na figura a seguir.



Neste modo, a configuração deste Driver é salva junto com a aplicação e nenhum arquivo externo é consultado em tempo de execução.

A opção **Load Example** carrega uma configuração de exemplo, conforme a figura anterior.

## Configuração dos Tags

Este Driver aceita Tags PLC e Tags Bloco com qualquer parâmetro. Em tempo de execução, cada Tag é associado a uma Categoria de simulação e recebe valores de acordo com o configurado na respectiva Categoria.

---

# Arquivo de Configuração no Formato XML

---

O arquivo de configuração no formato **XML** deve conter um nó **RandomConfig** na raiz. Este nó pode ter um ou mais Perfis. Cada Perfil pode ter uma ou mais Categorias, e cada Categoria pode ainda ter sub-Categorias. O código a seguir contém um exemplo de arquivo de configuração.

```
<RandomConfig>
  <Profile Name="Random Default" Digital="0,1" Delay="100">
    <Cat Name="DefaultAnalog" Filter="N2=1" Analog="0,1000">
      <Cat Name="SlowDefaultAnalog" Filter="N2=1 AND N3=1" Delay="500" />
    </Cat>
  </Profile>
</RandomConfig>
```

A configuração anterior define um Perfil **Random Default** com as seguintes características:

- Os Tags com os parâmetros *N2* e *N3* iguais a 1 (um) estão na Categoria **SlowDefaultAnalog**, possuem valor analógico variando entre 0 (zero) e 1000, herdado da Categoria superior **DefaultAnalog**, e um atraso de 500 milissegundos em todas as operações de leitura e escrita
- Os demais Tags com o parâmetro *N2* igual a 1 (um) estão na Categoria **DefaultAnalog**, possuem valor analógico variando entre 0 (zero) e 1000 e um atraso de 100 milissegundos nas operações de leitura e escrita, ou seja, o atraso padrão configurado no Perfil
- Os demais Tags não estão em nenhuma Categoria e possuem as características padrão configuradas no Perfil, ou seja, retornam valores digitais, 0 (zero) ou 1 (um), e possuem um atraso de 100 milissegundos em todas as operações de leitura e escrita

---

## Perfis

---

Cada arquivo de configuração no formato **XML** pode ter vários Perfis. Isto permite que se configurem diversos Drivers utilizando apenas um arquivo no formato **XML**.

Cada Perfil é configurado por um nó **Profile**, que pode ter os atributos a seguir:

- **Name:** Nome do Perfil, no formato **String**, como por exemplo **Name="DNPMaster Slow"**
- **Dll:** Lista de nomes de bibliotecas, separados por vírgula e sem a extensão, para as quais este Perfil é selecionado automaticamente, como por exemplo **Dll="DNPMaster,DNPSlave"**. Se o nome da biblioteca atual é DNPMaster.dll ou DNPSlave.dll, este Perfil é automaticamente selecionado
- **StartDelay:** Tempo de espera para iniciar este Driver, em milissegundos. Pode ser um valor único ou dois valores separados por vírgula, e neste caso é selecionado um valor aleatório no intervalo especificado, como por exemplo **StartDelay="4000,8000"**, que aguarda de 4 (quatro) a 8 (oito) segundos quando este Driver é iniciado
- **StopDelay:** Tempo de espera para finalizar este Driver, em milissegundos, como por exemplo **StopDelay="2000,2500"**, que aguarda de dois a dois segundos e meio quando este Driver é finalizado
- **Errors:** Define o percentual de erros em leituras e escritas para este Perfil, como por exemplo **Errors="0.5%"**, em que a cada 1000 leituras, cinco podem falhar
- **ReadVariation:** Define o percentual de leituras que retornam valores diferentes da leitura anterior, como por exemplo **ReadVariation="10%"**, em que apenas 10% das leituras retornam valores diferentes da leitura anterior, e portanto em 90% das leituras os Tags não têm variação
- **ReadDelay:** Tempo de espera antes de cada operação de leitura, em milissegundos, como por exemplo **ReadDelay="100,300"**, que aguarda de 100 a 300 milissegundos em cada leitura
- **WriteDelay:** Tempo de espera antes de cada operação de escrita, em milissegundos, como por exemplo **WriteDelay="2000,3000"**, que aguarda de 2 (dois) a 3 (três) segundos em cada escrita
- **Delay:** Tempo de espera antes de cada operação de leitura ou escrita, em milissegundos. Só é utilizado se os atributos **ReadDelay** ou **WriteDelay** não são especificados. Configurar este atributo com o valor 50, como por exemplo **Delay="50"**, permite aguardar 50 milissegundos antes de todas as operações de leitura e escrita
- **Analog:** Define todos os Tags como valores analógicos, como por exemplo **Analog="-100,500"**, em que um Tag varia de forma aleatória com valores entre -100 e 500, ou **Analog="50"**, em que um Tag retorna sempre o valor 50
- **Discrete:** Define todos os Tags como valores discretos ou digitais, como por exemplo **Digital="0,1"**, em que um Tag retorna o valor 0 (zero) ou 1 (um), **Discrete="5,10,15,20"**, em que um Tag retorna um dentre os valores 5 (cinco), 10, 15 ou 20, ou **Discrete="<>"**, em que um Tag não retorna nenhum valor



## NOTA

Não é permitido especificar os atributos **Analog** e **Discrete** ao mesmo tempo. Para especificar valores analógicos e digitais em um mesmo Driver, é necessário configurar uma Categoria.

# Categorias

Uma Categoria possui um filtro, que define os Tags que pertencem àquela Categoria, e configurações de variação para o valor dos Tags que atendem àquele filtro.

Cada Categoria é configurada por um nó **Cat** no arquivo de configuração no formato **XML** e possui os atributos a seguir:

- **Name:** Nome da Categoria, como por exemplo **Name="AnalogInputs"**
- **Filter:** Expressão, no padrão VBScript, que determina quais os Tags que pertencem a esta Categoria, como por exemplo **Filter="(N1=1 OR N1=2) AND N2 < 50"**, em que apenas os Tags que tem o parâmetro *N1* igual a 1 (um) ou igual a 2 (dois) e o parâmetro *N2* é menor que 50 são afetados por esta Categoria
- Podem ser utilizados os identificadores **N1, N2, N3, N4, B1, B2, B3, B4, Size, Count, ParamDevice** e **ParamItem** nos filtros
- Pode-se utilizar constantes no formato **String** com aspas simples, que são interpretadas como aspas duplas, como por exemplo **Filter="Left(ParamItem,6)='Analog'"**, que seleciona Tags cujo parâmetro **ParamItem** inicia com a expressão "Analog"
- As Categorias são avaliadas na ordem em que estão declaradas no arquivo de configuração no formato **XML**. Um Tag é associado à primeira Categoria encontrada em que um filtro retorna Verdadeiro
- Se um filtro está vazio, todos os Tags pertencem a esta Categoria
- Um filtro de uma sub-Categoria é avaliado antes de um filtro da Categoria que a contém

- **Errors, ReadVariation, ReadDelay, WriteDelay, Delay, Analog e Discrete:** Estes atributos possuem a mesma função definida em Perfis. Se um atributo não é definido em uma Categoria, utiliza-se o valor definido na Categoria superior ou Perfil

---

## Adicionando Simulações em Aplicações Existentes

---

O Driver ELIPSE Random pode ser utilizado para simular toda a coleta de dados de uma aplicação existente. Este Driver pode ser injetado simplesmente copiando o arquivo Random.dll e substituindo as bibliotecas dos Drivers utilizados por uma aplicação, sem necessidade de alterar qualquer configuração dos projetos existentes.

Inicie descompactando o conteúdo do arquivo Random.zip, que contém os arquivos Random.xml, Random.dll, ReplaceDrivers.bat e RollbackOriginalDrivers.bat, na pasta raiz da aplicação.

Para substituir todos os Drivers de uma aplicação pelo Driver ELIPSE Random, execute o arquivo ReplaceDrivers.bat. Este script busca nas sub-pastas de uma aplicação por todas as bibliotecas, como por exemplo **Unit1\DNPMaster.dll**, gera uma cópia de segurança, como por exemplo **Unit1\DNPMaster.dll.original**, e copia este Driver por cima daquela biblioteca, usando o comando **copy Random.dll Unit1\DNPMaster.dll**. O resultado é que os nomes das bibliotecas são mantidos, mas todos os Drivers agora são do tipo **Random**. Este procedimento deve ser efetuado com a aplicação parada e fechada.

O segundo passo é personalizar o arquivo Random.xml, que é utilizado por todos os Drivers de uma aplicação. O comportamento padrão deste Driver é procurar por este arquivo em disco, partindo da pasta onde se encontra a biblioteca deste Driver, neste caso o arquivo DNPMaster.dll, e subindo na hierarquia de pastas até encontrar o arquivo Random.xml. Ao colocar o arquivo Random.xml na pasta raiz de um Domínio, garante-se que todos os Drivers utilizam o mesmo arquivo de configuração, sem precisar alterar nenhuma configuração na aplicação.

Podem ser criados Perfis diferentes no arquivo de configuração para cada tipo de Driver, utilizando o nome do Driver no atributo **Dll** de cada Perfil. Por exemplo, em um aplicação que contém os Drivers Modbus32.dll, DNPMaster.dll e BACnet.dll, pode-se ter um arquivo de configuração com o código a seguir.

```
<RandomConfig>
  <Profile Name="DNP" Dll="DNPMaster">
    Configurações do Driver DNPMaster
  </Profile>
  <Profile Name="Modbus" Dll="Modbus32">
    Configurações do Driver Modbus
  </Profile>
  <Profile Name="BACnet" Dll="BACnet">
    Configurações do Driver BACnet
  </Profile>
</RandomConfig>
```

Execute a aplicação para que cada um dos Drivers, que na verdade são uma cópia renomeada do arquivo Random.dll, encontre o arquivo Random.xml e carregue o Perfil que equivale ao nome da respectiva biblioteca.

Para voltar aos Drivers originais, execute o arquivo RollbackOriginalDrivers.bat para substituir os arquivos com a extensão .dll pelos arquivos com a extensão .dll.original correspondentes.

---

## Documentação das Interfaces de Comunicação

---

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

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                                                                                                    |
|--------|------------|------------|----------------------------------------------------------------------------------------------------------------|
| 1.0.12 | 12/06/2025 | M. Ludwig  | • Driver atualizado para a biblioteca <b>IOKit</b> versão <b>3</b> e Visual Studio 2022 ( <i>Case 37964</i> ). |
| 1.0.3  | 17/07/2014 | F. Englert | • Versão original deste Driver.                                                                                |