Beckhoff - PLC Symbols



Hi Everyone

I am uploading here one of my big contribution projects of Beckhoff PLC symbols parser.

The symbol parser is a c# tool that read the internal memory of Beckhoff PLC memory using Twincat and produce a structure of the symbols that can be later used easily to read and write PLC symbols with c# visual studio intelisense uses.

It all start with a symbol file, Twincat allow you to read all its internal variable

For example:

"Constants.bFPUSupport,BOOL",

"Constants.bLittleEndian,BOOL",

"Constants.bSimulationMode,BOOL",

"Constants.CompilerVersion,VERSION",

"Constants.CompilerVersion.uiMajor,UINT",

"Constants.CompilerVersion.uiMinor,UINT",

"Constants.CompilerVersion.uiServicePack,UINT",

"Constants.CompilerVersion.uiPatch,UINT",

"Constants.CompilerVersionNumeric,DWORD",

"Constants.nPackMode,UINT",

"Constants.nRegisterSize,WORD",

"Constants.RuntimeVersion,VERSION",

"Constants.RuntimeVersion.uiMajor,UINT",

"Constants.RuntimeVersion.uiMinor,UINT",

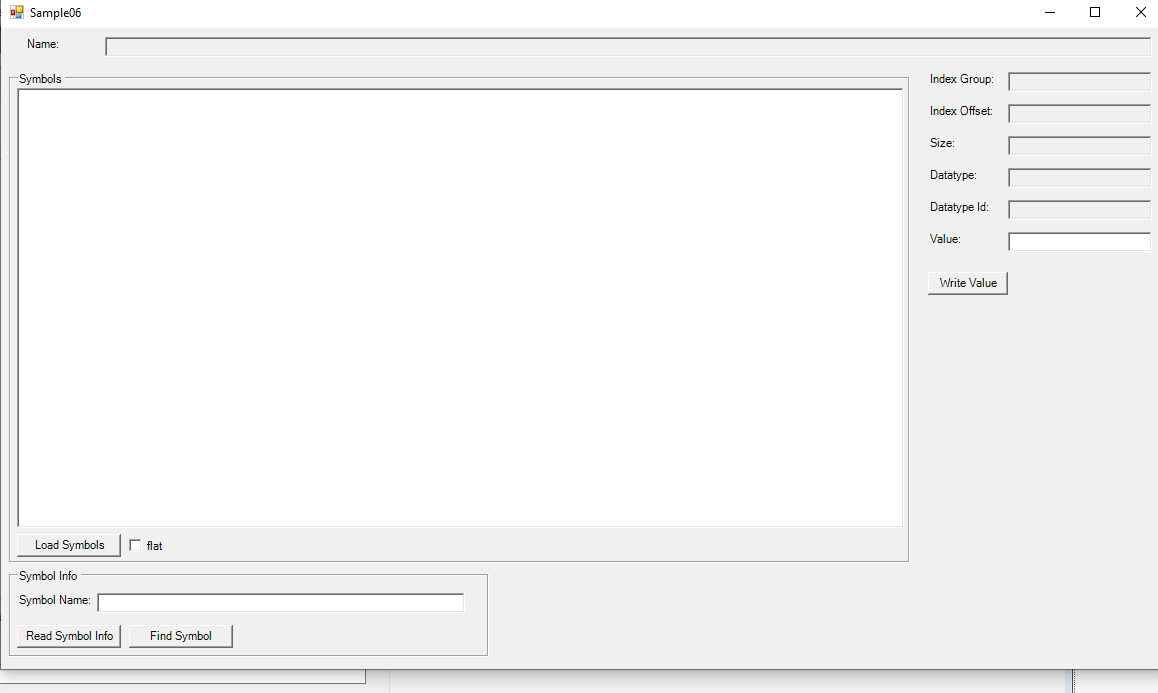
"Constants.RuntimeVersion.uiServicePack,UINT",

"Constants.RuntimeVersion.uiPatch,UINT",

"Constants.RuntimeVersionNumeric,DWORD",

This can be view with Example 06 from Beckhoff:

ReadPLCVariables



In this example we can view all the symbols , read , write and find symbols.

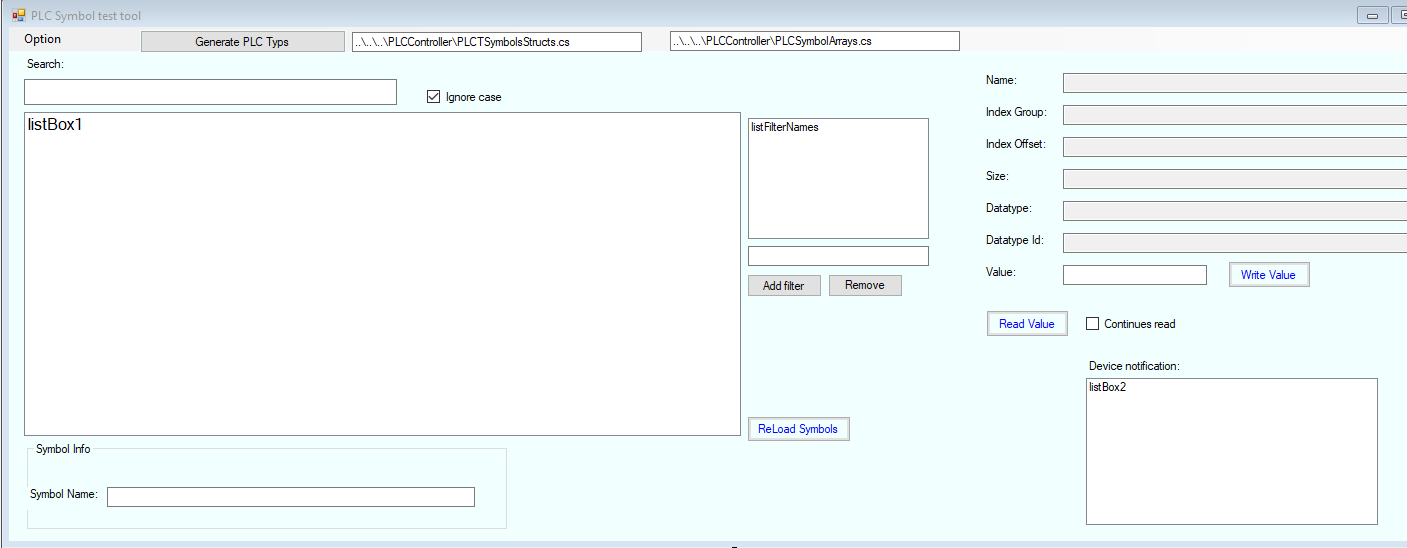
The Beckhoff symbols are recursive tree structure.

There are two kind of symbols

* Structures
* Array , and Array of structures

In the solution contains several projects

1. Array Parser creator
2. Structure Parser creator
3. Application that produce all in one button click



This application, similar to the sample 06, do the following

1. Create the initial plc\_symbols.txt
2. We can view the type, search variable, read, write, filter and more
3. Generate PLC Symbols struct and plc symbols array , both CS File ( CSharp files) that can be added into your application and then you can read and write the symbol easy with the intelisense typing.

Short example:

using System;

namespace PLCApi

{

public partial class PLCTypes : PLCSymbols

{

public struct STPLC

{

public struct ServoAxes

{

public static int NcAxesC

{

get

{

symbol = m\_plcSymbols["NcAxesC"];

return Convert.ToInt32(Client.ReadSymbol(symbol));

}

set

{

symbol = m\_plcSymbols["NcAxesC"];

Client.WriteSymbol(symbol, value.ToString());

}

}

}

}

}

}

After running the symbols parser on the plc\_symbols.txt , we then get cs file.

Now we can add this file into our application and access the variable like that:

STPLC.ServoAxes.NcAxesC = 1

Or

Int x = STPLC.ServoAxes.NcAxesC

The solution contains all you need:

