INTRODUCTION

The .NET Controller Software Development Kit (SDK) for *groov* EPIC and SNAP PAC provides a simple, easy-to-use interface for a developer to quickly create an application to access objects in a PAC Control strategy running on an Opto 22 *groov* EPIC processor or SNAP PAC programmable automation controller. These objects typically include integer, float, and string variables and tables as well as analog and discrete input and output points.

The SDK targets the .NET Standard version 2.0. For an explanation of .NET frameworks and operating systems that support .NET Standard version 2.0, see Microsoft's® .NET Standard version 2.0. The SDK is available as a 100% managed DLL.

System Requirements

- Visual Studio appropriate for your .NET platform
- Windows 11 Professional
- Windows 10 Professional (32- or 64-bit)
- Ethernet interface (wired or wireless).

ADDING THE SDK TO YOUR PROJECT

The SDK must be added as a resource to your Visual Studio project. It may be added to a C# or Visual Basic® project.

To add the SDK, follow these instructions:

1. Unzip the PAC-DEV-CONTROLLER-DOTNET.zip file with your preferred decompression tool.
2. In your Visual Studio Project, from the Project menu, choose Add Existing Reference.
3. Navigate to the Developer folder with the Controller_Std_2_0.dll.
4. Select the Controller_Std_2_0.dll.
5. In your project files, choose one of the following:
   – For Visual Basic, add an Imports (namespace) into your code file.
   For example: `Imports PACController`
   – For Visual C#, add a Using (namespace) into your code file.
   For example: `using PACController`
6. From Visual Studio's Nuget manager, add NLog (4.7.15) to your project.
7. To add the Intellisense DLL documentation to your project, navigate to the SDK's Visual Studio XML Assembly Documentation folder. Copy the file Opto22.Controller.XML to your project's source directory where your VB or C# source files are located.

Make sure that the Controller_Std_2_0.dll is in the same directory where the final executable is located.

To check whether the reference is included, open your project's Properties. Click the references tab. You should see the Controller_Std_2_0.dll listed as a reference.

Accessing Assembly Documentation

Assembly documentation is provided in html format.

1. Navigate to the folder `SDK Documentation`
2. Open the html folder and open `index.html`

MADE IN THE USA

Your default web browser will show the assemblies within the SDK.

Introduction to Assemblies

Namespace PACController

Use this assembly to read and write objects in a PAC Control strategy. The assembly contains three types of methods: writes, reads, and grouped reads.

Writes—Write methods allow data to be written to objects defined within the controller or to control the operation of the strategy.

Reads—Reads access properties and data of a controller. Through reads, properties such as the strategy name or the controller's current time can be accessed. For simple applications with minimal data access requirements, direct methods to read integers, floats, strings, timers, integer tables, float tables, string tables, analog points, and discrete points are immediately available.

Grouped reads—When data access requirements are larger, grouped access provides the minimum amount of latency for a volume of tags. Grouped access supports only the following strategy types: integers, floats, strings, timers, integer tables, float tables, string tables, analog points, and discrete points.

Examples

In the Examples subfolder of the SDK you'll find source code demonstrating both C# and VB .NET implementations.