

Introduction

The .NET Controller Software Development Kit (SDK) for 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 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 supports Microsoft's .NET frameworks 4.0 through 4.5, Visual Studio 2010-2013, and Windows 7 Professional, Windows Server 2012 R2, and Windows 8.1 Professional. The SDK is available as a 100% managed DLL.

System Requirements

- Visual Studio 2010 through 2013 (express or higher versions)
- Windows 7 Professional SP1 (32- or 64-bit editions), Windows Server 2012 R2, or Windows 8.1 (32- or 64-bit)
- x86 and AMD64 processors, 1.5 GHz or higher.
- Ethernet interface (wired or wireless).
- SNAP PAC controller (standalone S-series, rackmounted R-series, or SoftPAC software-based controller)

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#, Visual Basic, or a CLR-enabled C++ project (Visual Studio 2008 and later).

To add the SDK, follow these instructions:

1. In your Visual Studio Project, from the Project menu, choose Add Existing Reference.
2. Navigate to the DLL folder with the Opto22.Controller.dll.
3. Select the Opto22.Controller.dll.
4. In your project files, choose one of the following:
 - a. For Visual Basic, add an Imports (namespace) into your code file. For example: `Imports Opto22.Controller`
 - b. For Visual C#, add a Using (namespace) into your code file. For example: `using Opto22.Controller`

- c. For Visual C++, add a Using namespace (namespace) in the beginning of your code file.
For example: `using namespace Opto22::Controller;`
5. 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, C#, or C++ source files are located.

Make sure that the `Opto22.Controller.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 `Opto22.Controller.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`

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

Introduction to Assemblies

Namespace `Opto22.Controller`

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 for a console application demonstrating both C# and VB .Net implementations.