OPC.UAFX.ADVANCED

The assembly „Opc.UaFx.Advanced“ is a software framework, which can be used to communicate with client / server applications, which support interaction using the unified architecture (UA) standard defined by the OPC Foundation. Using this assembly it is possible to transform any .NET application with just a few lines of code into a OPC UA conform software to interact with OPC UA clients / servers.

In contrast to the default implementation of the OPC UA stack provided by the OPC Foundation and additional assemblies to implement session-based clients and servers does provide this framework much more advanced capabilities. It does not only reduce the amount of time required to understand the approach behind the OPC UA standard it does also minimize the amount of time required to implement custom solutions.

SUPPORTED PLATFORMS

- Starting with Microsoft Windows Vista (with .NET Framework 3.5)
- Windows CE (using the .NET Compact Framework)
- Diverse Unix Distributions (using Mono)
    - Suse Linux
    - Ubuntu
    - Debian
    - RedHat
    - Mac OS X

REQUIREMENTS

To develop an application using the Opc.UaFx.Advanced framework you just need a basic knowledge in programming with any .NET programming language (e.g. VB.Net, C#, …) and understanding in primitive client/server techniques.

GETTING STARTED

The Opc.ClientFx.Advanced framework does provide different ways through that you can implement an OPC UA conform client / server application.

OPC CLIENT DEVELOPMENT

1. WAY: THE OPC CLIENT CLASS

The OpcClient class of the framework does provide the whole functionality required to implement a session-based communication with an OPC server. This class implements all known services defined by the OPC UA standard to interact with an OPC server. To see how to use the OpcClient class take a look at the Samples.

2. WAY: THE OPC CLIENT APPLICATION CLASS

The OpcClientApplication class of the framework does provide the same functionality as the OpcClient class. It just improves the use of the OpcClient class through a main loop after a connection to an OPC server has been established. When the application will be exited it will close the connection. To see how to use the OpcClientApplication class take a look at the Samples.

OPC SERVER DEVELOPMENT

1. WAY: THE OPC SERVER CLASS

The OpcServer class of the framework does provide the whole functionality required to implement an OPC server. This class implements all known services defined by the OPC UA standard to interact with OPC UA clients. To see how to use the OpcServer class take a look at the Samples.

2. WAY: THE OPC SERVER APPLICATION CLASS

The OpcServerApplication class of the framework does provide the same functionality as the OpcServer class. It just improves the use of the OpcServer class through a main loop after a the custom OPC server has been started. When the application will be exited it will stop the server. To see how to use the OpcServerApplication class take a look at the Samples.

3. WAY: THE OPC SERVER SERVICE APPLICATION CLASS

The OpcServerServiceApplication class of the framework does provide the same functionality as the OpcServerApplication class. In addition it does extend the base class with a service-based model, which allows implementing an OPC server that can be executed as service process. This application does also identify whether the process is running in interactive user mode and does start the server in the same way, as it would be executed in case of the OpcServerApplication class. Otherwise it will use a service shell to control the server while supporting system specific service controller applications (e.g. SCM on Windows).