

OPC Server Developer Studio

Develop OPC Servers with C++, C# or VB.NET and be prepared for OPC UA



OPC Server Developer Studio
from Technosoftware AG

Powerful, feature-rich Server Development

Flexibility | Efficiency | Quality



TABLE OF CONTENTS

Introduction	3
Why OPC?.....	3
What is OPC?.....	4
OPC Specifications.....	4
Data Access (DA).....	4
Alarms&Events (AE).....	5
Historical Data Access (HDA).....	5
XML-DA.....	6
Unified Architecture.....	6
OPC Server Developer Studio	8
Specification Coverage	8
Platform Coverage.....	8
DLL Version.....	9
.NET Version	9
Source Version	9
OPC Server Developer Studio DLL.....	10
OPC Server Developer Studio .NET	12
OPC Server Developer Studio Source	14
Server API.....	16
Functions called by the Generic Server.....	16
Functions called by the Application Adaptation	17
Operation and Sequences.....	17
Performance	18
Performance Test Cases.....	18
Performance Test Results	20
Performance Test Analyses.....	22
OPC XML Server Developer Studio .NET	23
License Model	25
Subscription Coverage	25
Why Technosoftware	26



Introduction ...

Over the last ten years OPC has become the de facto standard for open communication between SCADA systems and field devices.

The first versions of the OPC standards are based on the DCOM architecture defined by Microsoft. These standards are limited to Microsoft Windows® operating systems because of the use of the DCOM architecture. Only a few implementations based on other operating systems exist, like Technosoftware AG solutions.

With the .NET initiative, Microsoft is now promoting another distributed architecture beside DCOM: the web services environment based on XML and W3C protocols. In a similar move, OPC is also specifying XML based communication.

The OPC Foundation also released a first version of a “unified architecture” specification that will now be totally interoperable with full internet connectivity and unlimited scalability across platforms of the end-users choice.

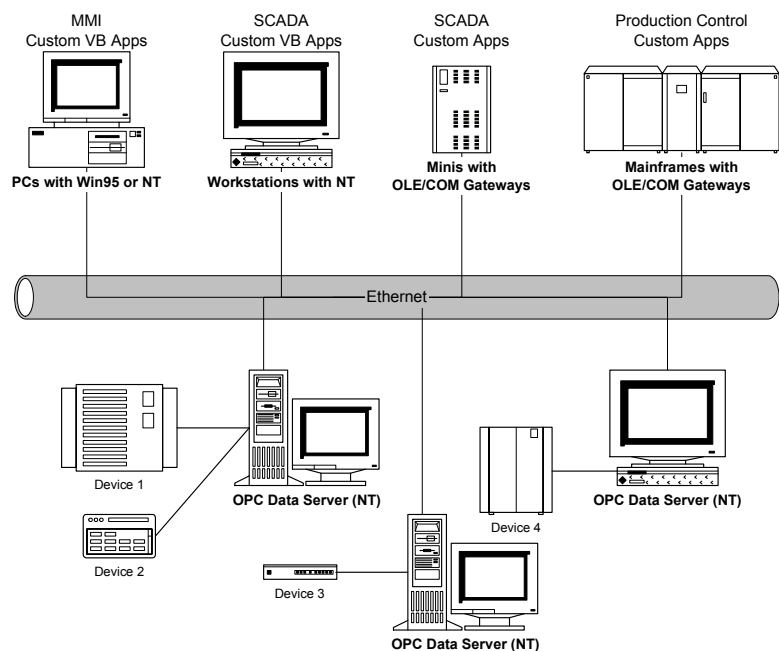
Most of the upcoming implementations are based on Microsoft Windows and the .NET architecture. At least for servers based on the new specifications this results in large and complex environments required by Microsoft OSs to run web services (.NET framework and IIS web server). But also for clients the .NET framework is still required. Implementing OPC clients or servers for these new specifications is complex and using well proven and tested developer tools can speed up development and ensure compliance with these specifications.

Why OPC?

In the automation industry very often devices from different hardware suppliers and software packages like visualization systems and process-control software from several software suppliers have to be combined to build a complete system. Within this system the different software components need to communicate. The application software should communicate with I/O devices as well as other applications. Getting the different software modules to work together is the biggest problem for process systems manufacturers.

These problems are due to missing or incompatible standards for data exchange interfaces. In the past vendors developed proprietary hardware and software solutions. All process-control and information systems today have their own interface to access the information.

Often a driver for one I/O device was written several times by different vendors. This can cause inconsistencies among different custom drivers or upgrades. It may also be impossible to use different software packages with one device at the same time because they use independent drivers and hardware features that are not supported by a custom driver.





In the past hardware vendors tried to solve some of these problems by delivering their own drivers.

The solution today is having a standard plug-and-play software technology for process-control and automation. Having such a standard makes it possible that different software packages can freely connect and communicate with different devices. This results in a truly open and easy enterprise-wide communication between systems and devices on the field, process or business management hierarchy.

What is OPC?

OPC is **O**penness **P**roductivity **C**ollaboration and the vision of OPC is to be the Foundation for interoperability

- For moving information vertically from the factory floor through the enterprise of multi-vendor systems (with stops in between...)
- For moving information horizontally between devices on different industrial networks from different vendors;
- Not just data but information.....

OPC Specifications

The OPC Foundation released several specifications in the last 10 years starting from OPC Data Access up to the new OPC Unified Architecture specification. Below just the specifications are mentioned which are covered by the OPC Developer Studio .NET.

Data Access (DA)

An OPC DA Server allows OPC DA Clients to retrieve information about several objects: the server, the group and the items.

- The OPC server object maintains information about the server and acts as a container for OPC group objects.
- The OPC group object maintains information about itself and provides the mechanism for containing and logically organizing OPC items.
- The OPC items represent connections to data sources within the server.

The OPC DA Specification defines two read/write interfaces:

- Synchronous
The client can perform a synchronous read from cache (simple and reasonably efficient). This may be appropriate for fairly simple clients that are reading relatively small amounts of data.
- Asynchronous
The client can 'subscribe' to cached data using IAdviseSink or IOPCDataCallback which is more complex but very efficient. Asynchronous access is recommended because it minimizes the use of CPU and NETWORK resources.

In all cases the OPC DA Server gives the client access to current values of the OPC items. The OPC DA Server only holds current information in cache. Old information is overwritten. As a result of this it cannot be guaranteed that an OPC DA Client retrieves all changes in values (also not in asynchronous mode).

For such cases there exist two more OPC specifications, the OPC Alarms&Events and the OPC Historical Data Access Specification.



Alarms&Events (AE)

The OPC AE interface provides a mechanism for OPC AE clients to be notified when a specified event and/or alarm condition occurs. The browser interface also allows OPC AE clients to determine the list of events and conditions supported by an OPC AE Server as well as to get their current status.

Within OPC, an alarm is an abnormal condition and is thus a special case of a condition. A condition is a named state of the OPC Event Server or of one of its contained objects that is of interest to an OPC AE client. For example, the tag Temperature may have the following conditions associated with it: HighAlarm, HighHighAlarm, Normal, LowAlarm, and LowLowAlarm.

On the other hand, an event is a detectable occurrence that is of significance to the OPC Server, the device it represents, and its OPC AE clients. An event may or may not be associated with a condition. For example, the transition into HighAlarm and Normal conditions are events, which are associated with conditions. However, operator actions, system configuration changes, and system errors are examples of events, which are not related to specific conditions. OPC AE clients may subscribe to be notified of the occurrence of specified events.

The OPC AE specification provides methods enabling the OPC AE client to:

- Determine the types of events that are supported by the OPC AE server.
- Enter subscriptions to specified events so that OPC AE clients can receive notifications of their occurrences. Filters may be used to define a subset of desired events.
- Access and manipulate conditions implemented by the OPC AE server.

Historical Data Access (HDA)

Historical engines today produce an added source of information that should be distributed to users and software clients that are interested in this information. Currently most historical systems use their own proprietary interfaces for dissemination of data. There is no capability to augment or use existing historical solutions with other capabilities in a plug-n-play environment. This requires the developer to recreate the same infrastructure for their products, as all other vendors have had to develop independently with no interoperability with any other systems.

In keeping with the desire to integrate data at all levels of business, historical information can be considered to be another type of data.

There are several types of Historian servers. Some key types supported by the HDA specification are:

- Simple Trend data servers.
These servers provided little else then simple raw data storage. (Data would typically be the types of data available from an OPC Data Access server, usually provided in the form of a duple [Time Value & Quality])
- Complex data compression and analysis servers. These servers provide data compression as well as raw data storage. They are capable of providing summary data or data analysis functions, such as average values, minimums and maximums etc. They can support data updates and history of the updates. They can support storage of annotations along with the actual historical data storage.



XML-DA

The OPC Foundation has defined interfaces to Data Access Servers, Event Servers, Batch Servers, and History Data Access Servers. These servers have information that is valuable to the enterprise, and is currently being provided to enterprise applications via OLE/COM based interfaces.

XML, the eXtensible Markup Language, and XML-based schema languages provide another means to describe and exchange structured information between collaborating applications. XML is a technology that is more readily available across a wide range of platforms. OPC XML-Data Access (OPC XML-DA) is the OPC Foundation's adoption of the XML set of technologies to facilitate the exchange of plant data across the internet, and upwards into the enterprise domain.

The purpose of this specification is to continue OPC's goal of enabling and promoting interoperability of applications. The OPC XML-DA based interfaces simplify sharing and exchanging of data amongst the various levels of a plant hierarchy (from low level devices up to enterprise systems) and to a wider range of platforms. The goal for this specification is to provide:

- Support for OPC Data Access 2.0x/3.0 data
- Support for HTTP, and SOAP
- Support for Subscription based services
- Support for a Security approach

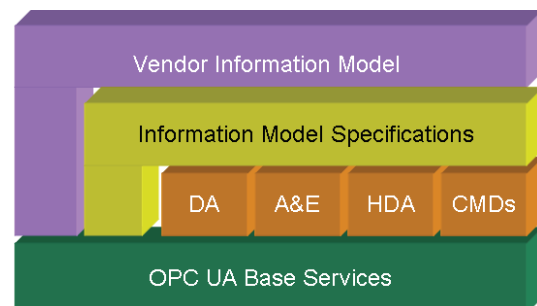
Unified Architecture

OPC Unified Architecture (UA) is a platform-independent standard through which various kinds of systems and devices can communicate by sending Messages between Clients and Servers over various types of networks. It supports robust, secure communication that assures the identity of Clients and Servers and resists attacks. OPC UA defines standard sets of Services that Servers may provide, and individual Servers specify to Clients what Service sets they support. Information is conveyed using standard

and vendor-defined data types, and Servers define object models that Clients can dynamically discover. Servers can provide access to both current and historical data, as well as Alarms and Events to notify Clients of important changes. OPC UA can be mapped onto a variety of communication protocols and data can be encoded in various ways to trade off portability and efficiency.

OPC UA provides a consistent, integrated AddressSpace and service model. This allows a single OPC UA Server to integrate data, Alarms and Events, and history into its AddressSpace, and to provide access to them using an integrated set of Services. These Services also include an integrated security model.

OPC UA also allows Servers to provide Clients with type definitions for the Objects accessed from the AddressSpace. This allows standard information models to be used to describe the contents of the AddressSpace. OPC UA allows data to be exposed in many different formats, including binary structures and XML documents. The format of the data may be defined by OPC, other standard organizations or vendors. Through the AddressSpace, Clients can query the Server for the metadata that describes the format for the data. In many cases, Clients with no pre-programmed knowledge of the data formats will be able to determine the formats at runtime and properly utilize the data.





OPC UA adds support for many relationships between Nodes instead of being limited to just a single hierarchy. In this way, an OPC UA Server may present data in a variety of hierarchies tailored to the way a set of Clients would typically like to view the data. This flexibility, combined with support for type definitions, makes OPC UA applicable to a wide array of problem domains. As illustrated below, OPC UA is not targeted at just the telemetry server interface, but also as a way to provide greater interoperability between higher level functions.

OPC UA is designed to provide robustness of published data. A major feature of all OPC servers is the ability to publish data and Event Notifications. OPC UA provides mechanisms for Clients to quickly detect and recover from communication failures associated with these transfers without having to wait for long timeouts provided by the underlying protocols.

OPC UA is designed to support a wide range of Servers, from plant floor PLCs to enterprise Servers. These Servers are characterized by a broad scope of size, performance, execution platforms and functional capabilities. Therefore, OPC UA defines a comprehensive set of capabilities, and Servers may implement a subset of these capabilities. To promote interoperability, OPC UA defines standard subsets, referred to as Profiles, to which Servers may claim conformance. Clients can then discover the Profiles of a Server, and tailor their interactions with that Server based on the Profiles. Profiles are defined in [UA Part 7].

The OPC UA specifications are layered to isolate the core design from the underlying computing technology and network transport. This allows OPC UA to be mapped to future technologies as necessary, without negating the basic design. Mappings and data encodings are described in [UA Part 6]. Two data encodings are defined in this part:

- XML/text
- UA Binary

In addition, two transport mappings are defined in this part:

- TCP
- SOAP Web services over HTTP

Clients and Servers that support multiple transports and encodings will allow the end users to make decisions about tradeoffs between performance and XML Web service compatibility at the time of deployment, rather than having these tradeoffs determined by the OPC vendor at the time of product definition.

OPC UA is designed as the migration path for OPC clients and servers that are based on Microsoft COM technology. Care has been taken in the design of OPC-UA so that existing data exposed by OPC COM servers (DA, HDA and A&E) can easily be mapped and exposed via OPC UA. Vendors may choose to migrate their products natively to OPC UA or use external wrappers to convert from OPC COM to OPC UA and vice-versa. Each of the previous OPC specifications defined its own address space model and its own set of Services. OPC UA unifies the previous models into a single integrated address space with a single set of Services.



OPC Server Developer Studio

The Server Developer Studio offers a fast and easy access to the OPC Server technology. Develop an OPC Server with any compiler capable of

- generating Windows DLL's (DLL version)
- generating a .NET assembly (.NET version)

or use the source version with Visual Studio 2005 C++.

The Server Developer Studio differs from other OPC development products; all versions base on the same common source code and are prepared for OPC UA. The Developer Studio API defines easy to use interface functions required for developing OPC DA, AE or UA compliant servers.

Develop now an OPC DA/AE server based on the DLL, .NET or Source version and use the same application adaptation part code later for support of OPC UA.

There are only a few functions that have to be implemented. The functions handle the configuration of the server and the read/write functionality of items. For more flexibility and because of performance issues the Server Developer Studio offers in addition a rich set of functions which allows optimizations.

Specification Coverage

	DA 1.0a	DA 2.05a	DA 3.00	AE 1.10	UA
DLL Version					
.NET Version					
Source Version					

Platform Coverage

	Windows 2000/XP	Windows 2003 Server	Windows Vista	Linux/UNIX	.NET 2.0 required
DLL Version					
.NET Version					
Source Version					

Note: For Linux/UNIX separate products are available. OPC UA for Linux/UNIX not yet planned

Explanation:



Already Supported



Coming Soon



Planned



Not Available



DLL Version

The DLL version offers a fast and easy access to the OPC DA Server technology. Develop an OPC Server with any compiler capable of generating Windows DLL's.

The DLL version differs from other rapid development products; all common code is provided by a single supplied EXE file, while the part which must be adapted resides in a DLL. This provides an elegant and straightforward development route.

The OPC server is a complete 32bit free-threaded OPC DA Server, which supports both the Custom and Automation interface.

The OPC server is supplied as an EXE file with the application adaptation part in a DLL. The source of sample application DLLs is included. The split into an EXE and DLL part imposes some limitations on the adaptation possibilities but makes the adaptation much easier and quicker. There are only 5 functions that have to be implemented. The functions handle the configuration of the server and the read/write functionality of items. For more flexibility and because of performance issues the DLL version offers a rich set of additional functions which allows optimizations until now only available with the Server Source Toolkit version.

.NET Version

The .NET version offers a fast and easy access to the OPC DA and AE Server technology and is prepared for OPC UA. Develop an OPC Server with any compiler capable of generating a .NET assembly.

The .NET version differs from other rapid development products; all common code is provided by a single supplied EXE file, while the part which must be adapted resides in a .NET assembly. This provides an elegant and straightforward development route.

The OPC server is supplied as an EXE file with the application adaptation part in a .NET assembly. The source of sample application .NET assemblies is included. As with the DLL version only 5 functions have to be implemented. The functions handle the configuration of the server and the read/write functionality of items. For more flexibility and because of performance issues with .NET wrapping of unmanaged COM code the .NET version offers a rich set of additional functions which allows optimizations until now only available with the Server Source Toolkit version.

Source Version

The Source Version provide complete source for OPC Server Development. The provided sample servers and sample clients, together with detailed documentation and online help provides essential information about OPC, OLE and (D)COM. The Source Version also includes the .NET wrapper used for the .NET version of the OPC Server Developer Studio. This gives you the most complete and powerful OPC Server Development Solution on the market.

The server class structure clearly separates the application specific part from the generic OPC server code. Therefore a server can be developed within a few days without the need to work through the OPC specification.



OPC Server Developer Studio DLL

The OPC Server Developer Studio DLL offers a fast and easy access to the OPC Data Access (DA) Server technology. Develop OPC DA 1.0a, 2.05a, 3.00 compliant Servers with any compiler capable of generating Windows DLL's.

The developer can concentrate on his application and servers can be developed fast and easily without the need to spend a lot of time learning how to implement the OPC specifications. The server API is easy to use and many OPC specific functions, e.g. creating a group or adding an item to a group are handled by the framework. Even the complex asynchronous read/write handling is handled by the framework.

The **"Framework"** includes all OPC DA 1.0a, 2.05a, 3.00 handling and ensures the OPC compliance. It is implemented as a generic C++ based executable.

The **"Server API"** defines easy to use interface functions required for developing OPC DA compliant servers. The OPC server is supplied as an EXE file with the application adaptation part in a Windows DLL. This imposes some limitations on the adaptation possibilities but makes the adaptation much easier and quicker. By using this API DA servers can be easily implemented by adapting just a few functions, e.g. there are only 5 functions that have to be implemented for an OPC DA Server. The functions handle the configuration of the server and the read/write functionality of items.

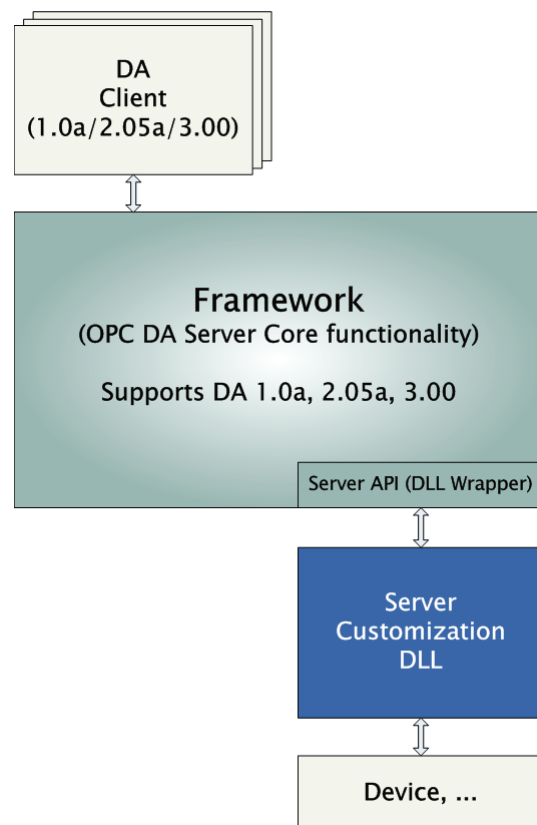
The server supports all mandatory and optional OPC DA 1.0a / DA 2.05a interfaces and all mandatory OPC DA 3.00 interfaces. The provided sample servers and sample clients, together with the detailed documentation and online help provides all essential information and tools you need to successfully develop your server application.

The OPC Server Developer Studio DLL offers unique features for performance and functionality improvements of the developed OPC Server like Event Driven Mode for Device Access; Dynamic address space with items added when they are first accessed by a client and removed when they are no longer in use; Item browsing can be implemented to browse the cache or the device/database.

The Base Edition is the low cost solution for OPC DA Server Development with a few limitations, like no Dynamic address space, no Item browsing of the device/database (browsing always use the cache), no Deadband Handling and no custom properties.

A free evaluation version is available at www.technosoftware.com featuring full functionality including the DLL and source code of all provided sample servers. It can be used for 14 days.

"The OPC Server Developer Studio DLL offers a fast and easy access to the OPC DA Server technology. Develop an OPC DA 1.0a / 2.05a / 3.00 compliant Servers with any compiler capable of generating Windows DLL's."



Picture 2: OPC Server Developer Studio DLL Structure



Features Include

The OPC Server Developer Studio DLL includes features like:

- ✓ Short time-to-market for standard compliant products. Develop OPC servers in days instead of weeks or months.
- ✓ Highly utilizable tool for quick and easy development of OPC compliant servers.
- ✓ Powerful possibilities for performance optimizations.
- ✓ Event Driven Mode for Device Access.
- ✓ Dynamic address space handling allowing adding and removing items during run-time.
- ✓ Item browsing can be implemented browsing the cache or device/database.
- ✓ Custom Properties.
- ✓ Can be used as NT Service.
- ✓ Servers are fully compliant to the OPC Specifications DA 1.0a, DA 2.05a and DA 3.00.
- ✓ Tested with the OPC Foundation Compliance Test Tool.
- ✓ Generic Server is implemented as a C++ based executable.
- ✓ The application specific part can be implemented with any compiler capable of generating a Windows DLL.
- ✓ Extensive diagnostic and installation support.

Self-Tested for Compliance

The OPC Foundation Self-Tested for Compliance logo indicates that these products have been tested by the manufacturer to be compliant with the following OPC Specifications: Data Access 1.0a, Data Access 2.00, Data Access 2.05a, and Data Access 3.00. Additional information about compliance testing, logo program and a summary of test results for the OPC Swift Development Kit can be found at www.opcfoundation.org.

Product Specification

Product Type

- Server Development

OPC Compatibility

- OPC Data Access 1.0a
- OPC Data Access 2.05a
- OPC Data Access 3.00

Supported Operating Systems

- Windows 2000, XP
- Windows 2003 Server
- Windows Vista™

Development Environment

- Microsoft Visual Studio .NET 2005 (C++)
- Borland Delphi 2007 for Win32

Development Interface

- OPC Server Implementation with any compiler capable of generating Windows DLL's



Order Information

Product ID	Description	List Price (EURO) *
TSOPC-SDD-BAS	Developer Studio DLL Base Edition	1295.-
TSOPC-SDD-BAS-SUPT	1 st year support and updates	395.-
TSOPC-SDD	Developer Studio DLL	2495.-
TSOPC-SDD-SUPT	1 st year support and updates	795.-

License

- Per-Developer Basis
- One Run-time license included

Delivery

- Downloadable product
- Free Standard Support for a 90 days period included.

* Pricing is subject to change / Call us for updated price



OPC Server Developer Studio .NET

The OPC Server Developer Studio .NET offers a fast and easy access to the OPC Data Access (DA) and OPC Alarms&Events (AE) technology. Develop OPC DA 1.0a, 2.05a, 3.00 and OPC AE 1.00, 1.10 compliant Servers with C#/VB.NET or any other compiler capable of generating a .NET assembly.

The developer can concentrate on his application and servers can be developed fast and easily without the need to spend a lot of time learning how to implement the OPC specifications. The server API is easy to use and many OPC specific functions, e.g. creating a group or adding an item to a group are handled by the framework. Even the complex asynchronous read/write handling is handled by the framework.

The **"Framework"** includes all OPC DA 1.0a, 2.05a, 3.00 and OPC AE 1.00, 1.10 handling and ensures the OPC compliance. It is implemented as a generic C++ based executable.

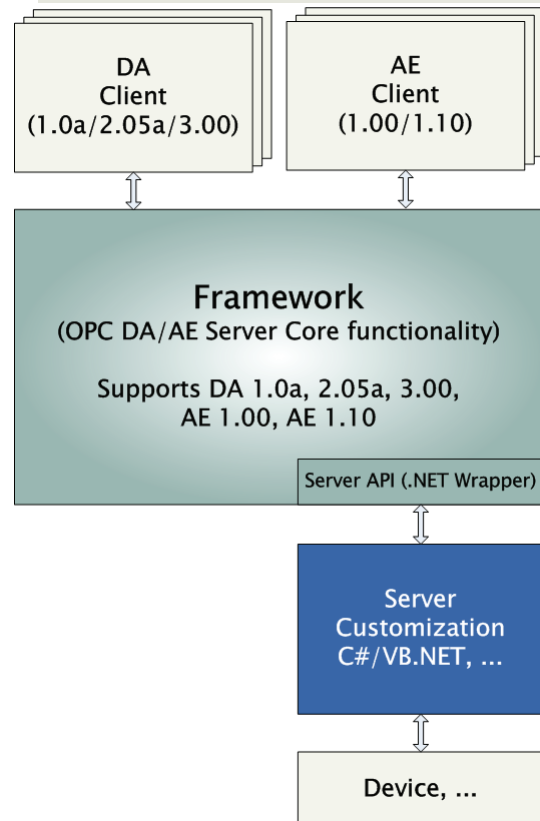
The **"Server API"** defines easy to use interface functions required for developing OPC DA/AE compliant servers. The OPC server is supplied as an EXE file with the application adaptation part in a .NET assembly DLL. This imposes some limitations on the adaptation possibilities but makes the adaptation much easier and quicker. By using this API DA/AE servers can be easily implemented by adapting just a few functions, e.g. there are only 5 functions that have to be implemented for an OPC DA Server. The functions handle the configuration of the server and the read/write functionality of items.

The server supports all mandatory and optional OPC DA 1.0a / DA 2.05a interfaces and all mandatory OPC DA 3.00 interfaces. With the available AE add-on the server supports also OPC AE 1.00 / 1.10. The provided sample servers and sample clients, together with the detailed documentation and online help provides all essential information and tools you need to successfully develop your server application.

The OPC Server Developer Studio .NET offers unique features for performance and functionality improvements of the developed OPC Server like Event Driven Mode for Device Access; Dynamic address space with items added when they are first accessed by a client and removed when they are no longer in use; Item browsing can be implemented to browse the cache or the device/database.

A free evaluation version is available at www.technosoftware.com featuring full functionality including the DLL and source code of all provided sample servers. It can be used for 14 days.

"The OPC Server Developer Studio .NET offers a fast and easy access to the OPC server technology. Develop OPC DA 1.0a, 2.05a, 3.00 and OPC AE 1.00 / 1.10 compliant Servers with C#/VB.NET."



Picture 3: OPC Server Developer Studio .NET Structure



Features Include

The OPC Server Developer Studio .NET includes features like:

- ✓ Short time-to-market for standard compliant products. Develop OPC servers in days instead of weeks or months.
- ✓ Highly utilizable tool for quick and easy development of OPC compliant servers.
- ✓ Powerful possibilities for performance optimizations.
- ✓ Event Driven Mode for Device Access.
- ✓ Dynamic address space handling allowing adding and removing items during run-time.
- ✓ Item browsing can be implemented browsing the cache or device/database.
- ✓ Custom Properties.
- ✓ Can be used as NT Service.
- ✓ Servers are fully compliant to the OPC Specifications DA 1.0a, DA 2.05a, DA 3.00 and AE 1.00 / AE 1.10.
- ✓ Tested with the OPC Foundation Compliance Test Tool.
- ✓ Generic Server is implemented as a C++ based executable.
- ✓ The application specific part can be implemented with C#/VB.NET or any other compiler capable of generating a .NET assembly.
- ✓ Extensive diagnostic and installation support.

Self-Tested for Compliance

The OPC Foundation Self-Tested for Compliance logo indicates that these products have been tested by the manufacturer to be compliant with the following OPC Specifications: Data Access 1.0a, 2.00, 2.05a, 3.00 and Alarms&Events 1.00, 1.10. Additional information about compliance testing, logo program and a summary of test results for the OPC Server Developer Studio .NET can be found at www.opcfoundation.org.

Product Specification

Product Type

- Server Development

OPC Compatibility

- OPC Data Access 1.0a
- OPC Data Access 2.05a
- OPC Data Access 3.00
- OPC Alarms&Events 1.00
- OPC Alarms&Events 1.10



Supported Operating Systems

- Windows 2000, XP
- Windows 2003 Server
- Windows Vista™

Development Environment

- .NET Framework 2.0
- Microsoft Visual Studio 2005 (C#/VB.NET)

Development Interface

- .NET Interfaces

Order Information

Product ID	Description	List Price (EURO)*
TSOPC-SDN	Developer Studio .NET	2495.-
TSOPC-SDN-AE	AE Server add-on	1195.-
TSOPC-SDN-SUPT	1 st year support and updates	795.-

License

- Per-Developer Basis
- One Run-time license included

Delivery

- Downloadable product
- Free Standard Support for a 90 days period included.

* Pricing is subject to change / Call us for updated price



OPC Server Developer Studio Source

The OPC Server Developer Studio Source offers a fast and easy access to the OPC Data Access (DA) and OPC Alarms&Events (AE) technology. Develop OPC DA 1.0a, 2.05a, 3.00 and OPC AE 1.00, 1.10 compliant Servers with C++ or C#/VB.NET or any other compiler capable of generating a .NET assembly.

The developer can concentrate on his application and servers can be developed fast and easily without the need to spend a lot of time learning how to implement the OPC specifications. The server API is easy to use and many OPC specific functions, e.g. creating a group or adding an item to a group are handled by the framework. Even the complex asynchronous read/write handling is handled by the framework.

The **"Framework"** includes all OPC DA 1.0a, 2.05a, 3.00 and OPC AE 1.00, 1.10 handling and ensures the OPC compliance. It is implemented as generic C++ classes.

The **"Server API"** defines easy to use interface functions required for developing OPC DA/AE compliant servers. The OPC server is supplied as C++ source with the application adaptation part clearly separated from the framework.

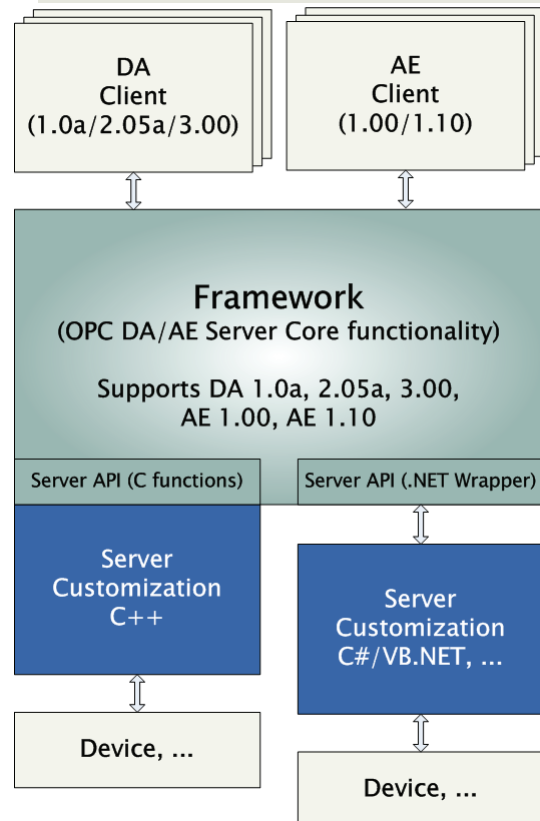
Only a few methods need to be adapted to implement a new server. The code that needs to be modified is clearly separated from the generic OPC code, making updates to new OPC standard versions easy and straightforward.

The availability of the full source code protects your investments and allows you to fully adapt it to your requirements. Whether you want to quickly implement a simple server or create high performance OPC servers with ten thousands of items the OPC Server Developer Studio Source is the right choice.

The OPC Server Developer Studio Source offers unique features for performance and functionality improvements of the developed OPC Server like Event Driven Mode for Device Access; Dynamic address space with items added when they are first accessed by a client; Item browsing can be implemented to browse the cache or the device/database.

The included client executables can be used to test a server implementation and therefore eliminate the effort to create own test clients.

"The OPC Server Developer Studio Source offers a highly utilizable tool for quick and easy development of OPC compliant DA 1.00a / 2.05a / 3.00 and OPC AE 1.00 / 1.10 compliant Servers."



Picture 3: OPC Server Developer Studio Source Structure



Features Include

The OPC Server Developer Studio Source includes features like:

- ✓ Short time-to-market for standard compliant products. Develop OPC servers in days instead of weeks or months.
- ✓ Highly utilizable tool for quick and easy development of OPC compliant servers.
- ✓ Powerful possibilities for performance optimizations.
- ✓ Event Driven Mode for Device Access.
- ✓ Dynamic address space handling allowing adding and removing items during run-time.
- ✓ Item browsing can be implemented browsing the cache or device/database.
- ✓ Custom Properties.
- ✓ Can be used as NT Service.
- ✓ Servers are fully compliant to the OPC Specifications DA 1.0a, DA 2.05a, DA 3.00 and AE 1.00 / AE 1.10.
- ✓ Tested with the OPC Foundation Compliance Test Tool.
- ✓ Complete source code for all server modules included.
- ✓ The application specific part can be implemented with C++ or C#/VB.NET or any other compiler capable of generating a .NET assembly.
- ✓ Diagnostic client and diagnostic tools as executables included.
- ✓ Extensive diagnostic and installation support.

Self-Tested for Compliance

The OPC Foundation Self-Tested for Compliance logo indicates that these products have been tested by the manufacturer to be compliant with the following OPC Specifications: Data Access 1.0a, 2.00, 2.05a, 3.00 and Alarms&Events 1.00, 1.10. Additional information about compliance testing, logo program and a summary of test results for the OPC Server Developer Studio Source can be found at www.opcfoundation.org.

Product Specification

Product Type

- Server Development

OPC Compatibility

- OPC Data Access 1.0a
- OPC Data Access 2.05a
- OPC Data Access 3.00
- OPC Alarms&Events 1.00
- OPC Alarms&Events 1.10

Supported Operating Systems

- Windows 2000, XP
- Windows 2003 Server
- Windows Vista™

Development Environment

- Microsoft Visual Studio .NET 2005 (C++)
- or
- .NET Framework 2.0
 - Microsoft Visual Studio 2005 (C#/VB.NET)

Development Interface

- C++ Classes, full source code included
- or
- .NET Interfaces



Order Information

Product ID	Description	List Price (EURO) *
TSOPC-SDS	Developer Studio Source	5495.-
TSOPC-SDS-SUPT	1 st year support and updates	1795.-

License

- Per-Developer Basis
- One Run-time license included

Delivery

- Downloadable product
- Free Standard Support for a 90 days period included.

* Pricing is subject to change / Call us for updated price



Server API

This chapter includes a short overview of the important API functions for OPC Server Development. It doesn't show the complete API.

Functions called by the Generic Server

Name	Description
OnClientConnect	This method is called when a client connects to the OPC server. If the method returns an error code then the client connect is refused.
OnClientDisconnect	This method is called when a client disconnects from the OPC server.
OnCreateServerItems	This method is called from the generic server at the startup; when the first client connects or the service is started. All items supported by the server need to be defined by calling the AddItem callback method for each item.
OnGetDAServerParameters	This method is called from the generic server at startup; when the first client connects or the service is started. It defines the application specific server parameters and operating modes.
OnGetDAServerRegistryDefinition	This method is called from the generic server at startup for normal operation or for registration. It provides server registry information for this application required.
OnGetPropertyValue	Returns the values of the requested custom properties of the requested item. This method is not called for the OPC standard properties 1..6. These are handled in the generic server.
OnItemRequest	This method is called when the client accesses items that do not yet exist in the server's cache.
OnQueryProperties	Query the properties defined for the specified item
OnRefreshItems	Refresh the items listed in the parameter array in the cache. This method is called when a client executes a read from device. The device read is called with all client requested items.
OnShutdownSignal	This method is called from the generic server when a Shutdown is executed.
OnWriteItems	This method is called when a client executes a 'write' server call. The items specified in the parameter array need to be written to the device.



Functions called by the Application Adaptation

Name	Description
AddAnalogItem	This function is called by the application adaptation part and adds an analog item to the generic server cache.
AddItem	This function is called by the application adaptation part and adds an item to the generic server cache.
AddProperty	Adds a custom specific property to the generic server list of item properties.
RemoveItem	This function is called by the application adaptation part and removes an item from the generic server cache.
SetItemValue	Write an item value into the generic server cache.
SetServerState	Set the OPC server state value which is returned in client GetStatus calls.

Operation and Sequences

The application adaptation part of the server is called by the generic server part at different stages during startup and operation. Therefore complete control over the startup, operation and shutdown of the server is possible. A sample operation include the following steps:

The generic server calls some methods at startup:

OnGetDAServerRegistryDefinition

OnGetDAServerParameters

OnCreateServerItems

and some when a client calls certain functions:

OnRefreshItems

OnWriteItems

Callback functions allow the customization to get status information from the generic server or change data in the generic server. The most important callback methods are:

AddItem

SetItemValue

Additional functions can be called by the generic server, e.g. for dynamic Address Space handling

OnItemRequest

Additional callback functions can be used, e.g.:

RemoveItem



Performance

All available OPC Server Solutions from Technosoftware base on the same C++ source; the OPC DA/AE Server Core framework but they differ in the supported compilers, design and functionality. This results in a different performance available for the three versions.

Performance Test Cases

This chapter compares the performance of the OPC Server Developer Studio DLL, OPC Server Developer Studio .NET and OPC Server Developer Studio Source. To allow the best possible comparison the following test structure was chosen:

- Each server has an address space of 9000 data tags
- The device is just a simulation, all data tags are non-changed values
- With the OPC Server Developer Studio DLL only polled device access is possible, so the OPC Server build with the OPC Developer Studio Source also polls the device and no optimization is made
- The base update rate is set to 200ms
- Client and Server are running on the same PC

For all three servers the following test cases are defined:

1. Synchronous Read from Cache
For all three servers Read from Cache is internally handled by the generic server. No calls are done to the custom specific part of the server. This means the DLLs for the OPC Server Developer Studio DLL and the OPC Server Developer Studio .NET **are not called**.
2. Synchronous Read from Device
For all three servers Read from Device is handled by the custom specific part of the server. Calls are done from the generic server to the custom specific part of the server. This means the DLLs for the OPC Server Developer Studio DLL and the OPC Server Developer Studio .NET **are called**. Device access just means accessing internal data cache and returning the value.
3. Synchronous Write to Device
For all three servers Write to Device is handled by the custom specific part of the server. Calls are done from the generic server to the custom specific part of the server. This means the DLLs for the OPC Server Developer Studio DLL and the OPC Server Developer Studio .NET **are called**. Device access just means accessing internal data cache and writing the value to the custom specific server cache.
4. Asynchronous Read from Device
For all three servers Read from Device is handled by the custom specific part of the server. Calls are done from the generic server to the custom specific part of the server. This means the DLLs for the OPC Server Developer Studio DLL and the OPC Server Developer Studio .NET **are called**. Device access just means accessing internal data cache and returning the value.
5. Asynchronous Write to Device
For all three servers Write to Device is handled by the custom specific part of the server. Calls are done from the generic server to the custom specific part of the server. This means the DLLs for the OPC Server Developer Studio DLL and the OPC Server Developer Studio .NET **are called**. Device access just means accessing internal data cache and writing the value to the custom specific server cache.



For each test case the following sub test cases are defined:

1. Using Read/Write with 100 items in one group and accessing all items at once.
2. Using Read/Write with 300 items in one group and accessing all items at once.
3. Using Read/Write with 500 items in one group and accessing all items at once.
4. Using Read/Write with 1000 items in one group and accessing all items at once.
5. Using Read/Write with 2000 items in one group and accessing all items at once.

We compared the following products:

1. TSOPC-SDS
The sample server provided with the OPC Server Developer Studio Source was modified in a few points to be comparable with the other products; custom property handling and dynamic item handling removed as well as implementation of static address space of 9000 items
2. TSOPC-SDD
The sample server provided with the OPC Server Developer Studio DLL was used as it is, just a static address space of 9000 items was implemented
3. TSOPC-SDN
Because of the test result of previous versions the new version introduces event driven mode for device access resulting in a better performance. Also write calls were optimized. The sample server provided with the OPC Server Developer Studio .NET was used as it is, just a static address space of 9000 items was implemented.
4. .NET Competitor
To be more comparable with other vendor products we have chosen a .NET based sample server of a competitor and implemented a static address space of 9000 items. This sample used also event driven mode for device access.

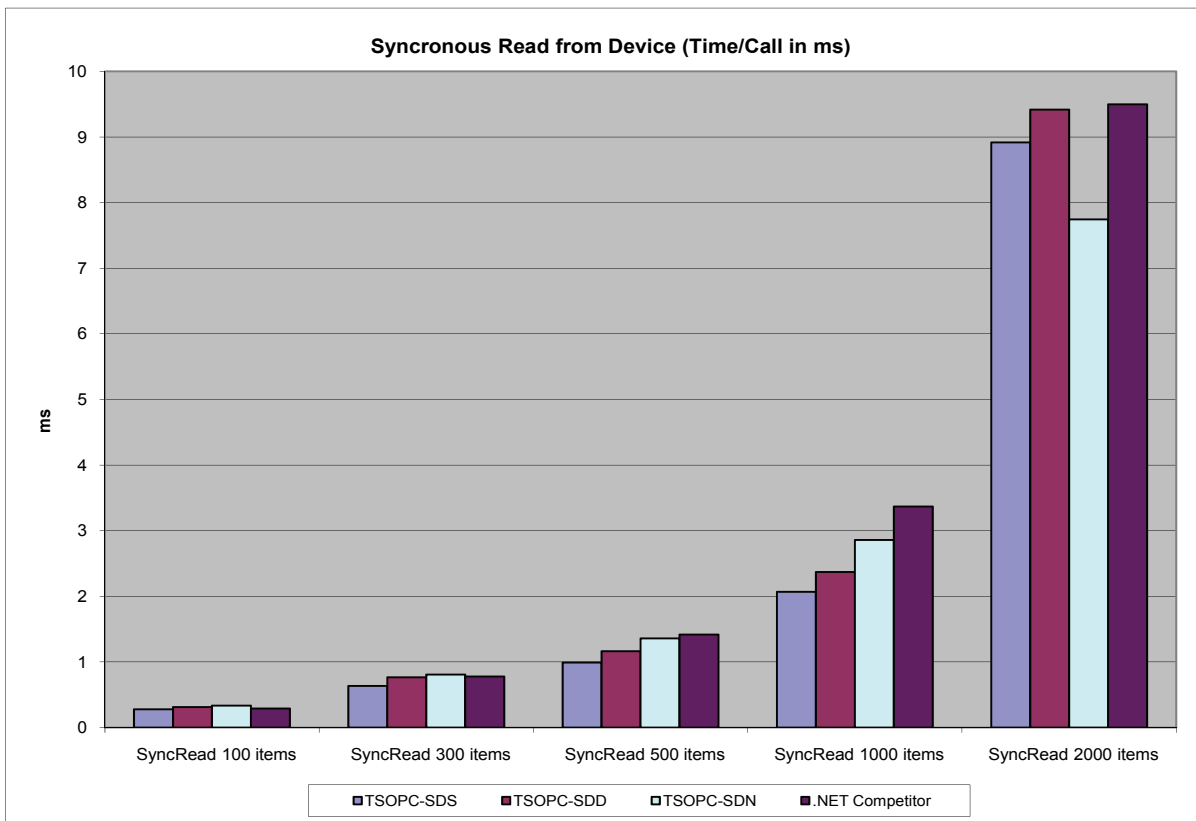
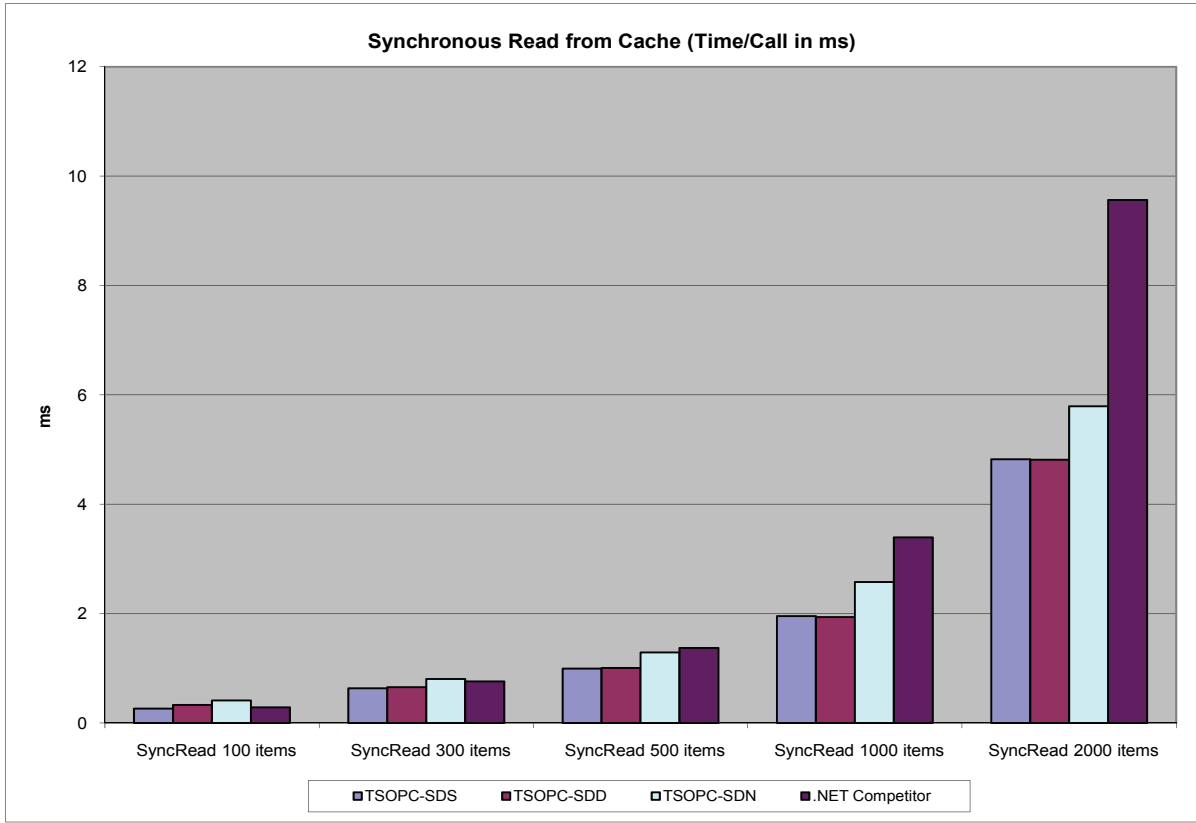
The done performance tests include only the COM and OPC specific part of an OPC Client <-> Server connection. It doesn't consider the device access. Therefore the shown results can only give an indication of the performance of the different versions. It also doesn't consider the optimization possibilities of the OPC Server Developer Studio Source.

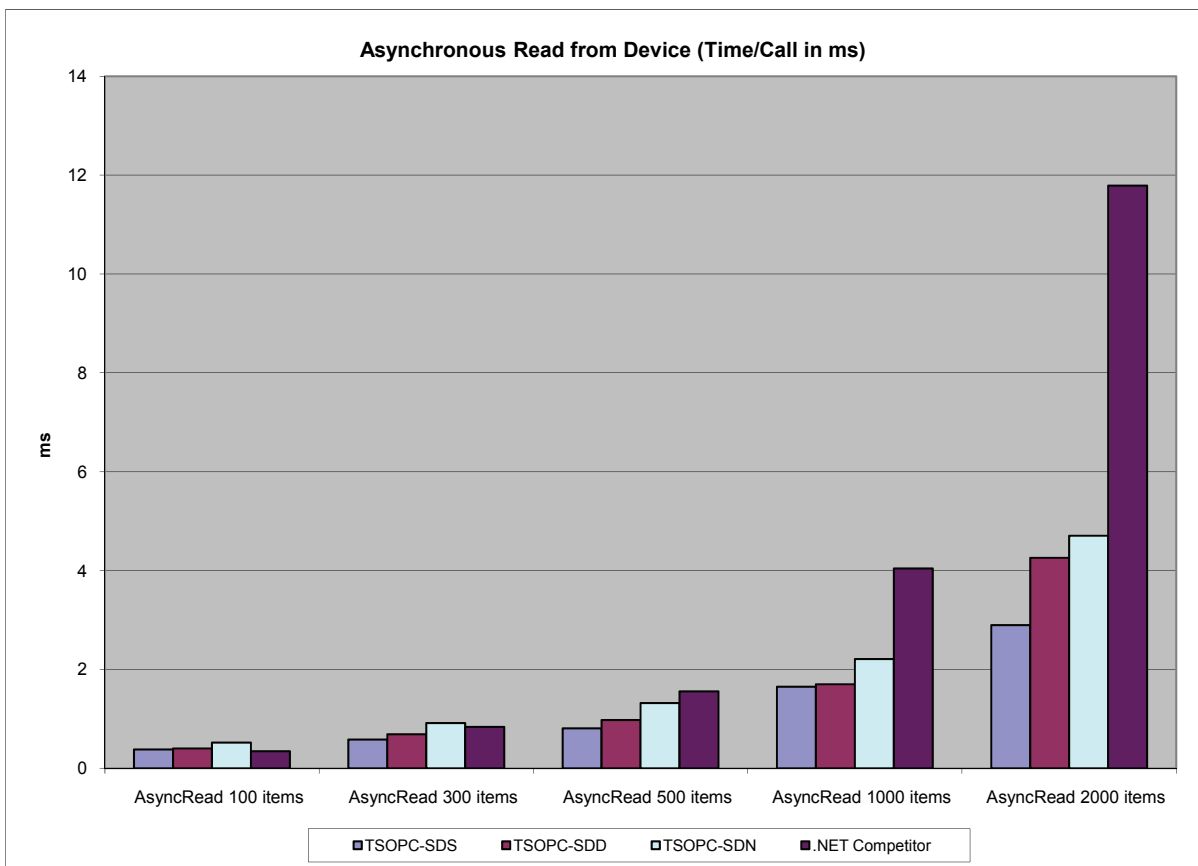
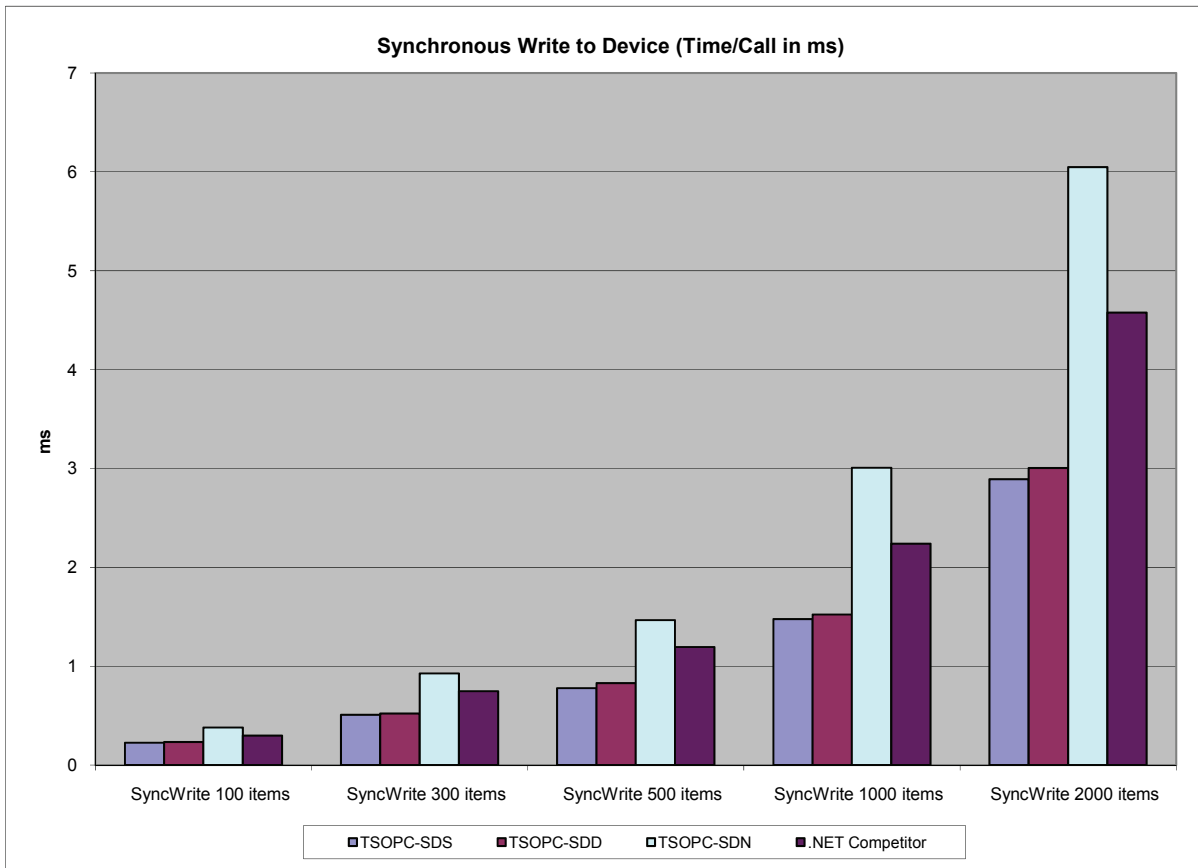
The performance differences mainly comes from the calls to DLLs, especially for the OPC Server Developer Studio .NET. The switch from Unmanaged C++/Managed C++ to a .NET assembly takes most of the time difference.

If you are interested in more detailed performance tests and analyses please contact us at sales@technosoftware.com. On a project specific base we can arrange such tests considering your specific environment.

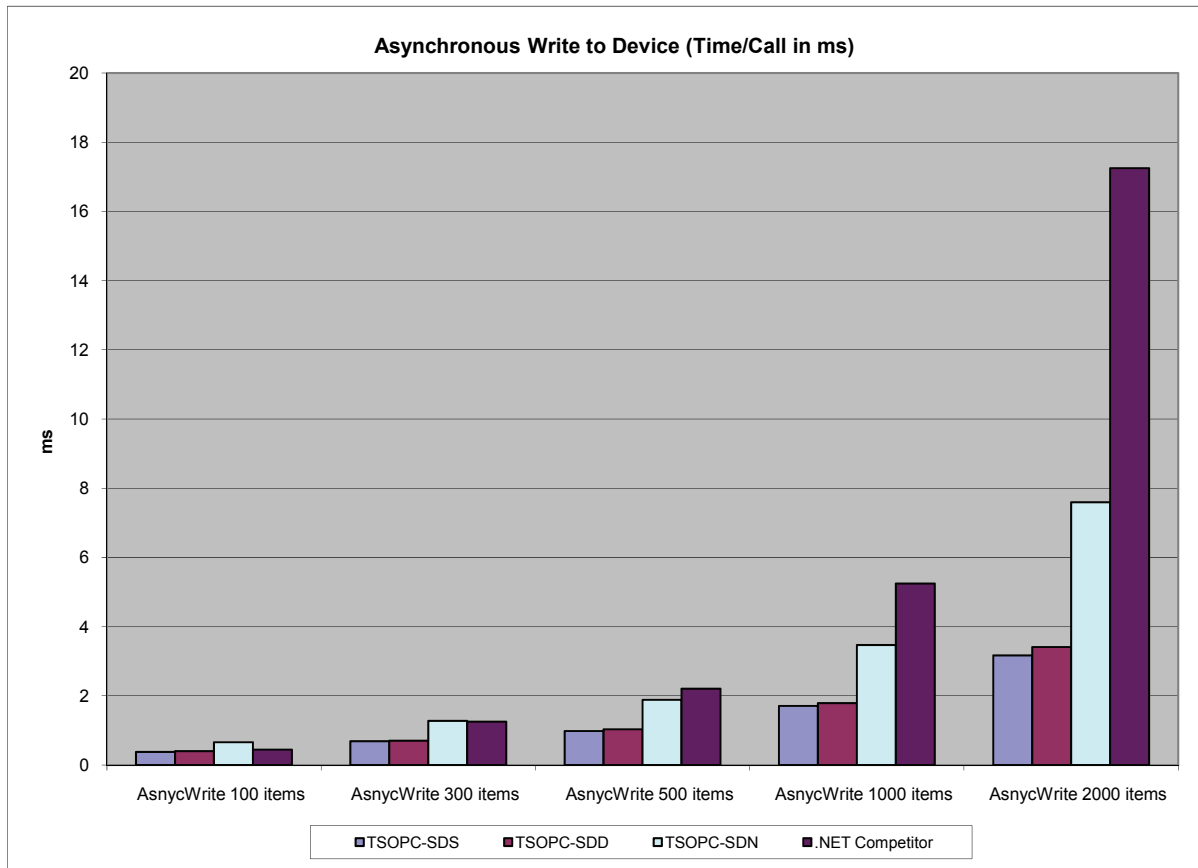


Performance Test Results





Flexibility | Efficiency | Quality



Performance Test Analyses

High performance OPC DA Servers can be build with the OPC Server Developer Studio Source (C++ based), even without any additional optimizations made. The optimization possibilities available with the OPC Server Developer Studio Source can be used to build even OPC DA Servers with more then 150'000 data tags.

The OPC Server Developer Studio DLL doesn't allow any optimizations and therefore can be used for OPC DA Servers up to a few thousand items only.

The OPC Server Developer Studio .NET has the lowest performance. This is a result from the required wrapper transforming unmanaged C++/managed C++ to the .NET world. Every call from the C++ based part to the .NET assembly and vice versa takes a significant amount of time. Even with the implementation of optimizations, like event driven device access, and reducing the numbers of calls between the C++ part and the .NET part, OPC Servers based on .NET has a lower performance then pure C++ based OPC Servers. But for many OPC Servers the .NET based solution can still be the right solution, e.g. in a preparation of a future OPC UA server implementation, for server applications with only a few thousand data tags or for server applications where fast access to data tags is not so important.

Technosoft allows you to choose the solution which best fit your requirements. And because all toolkits base on the same source code an upgrade or switch is easy. With the OPC Server Developer Studio Source you can even decide which part you implement in C++ and which part you implement in C#/VB.NET.



OPC XML Server Developer Studio .NET

The OPC XML Server Developer Studio .NET offers a fast and easy access to the OPC XML-DA (XML-DA) technology. Develop OPC XML-DA 1.01 compliant Servers with C#/VB.NET or any other compiler capable of generating a .NET assembly.

The developer can concentrate on his application and servers can be developed fast and easily without the need to spend a lot of time learning how to implement the OPC specification. The server API is easy to use and many OPC specific functions, e.g. creating a subscription or browsing the address space are handled by the framework. Even the complex polled refresh handling is handled by the framework.

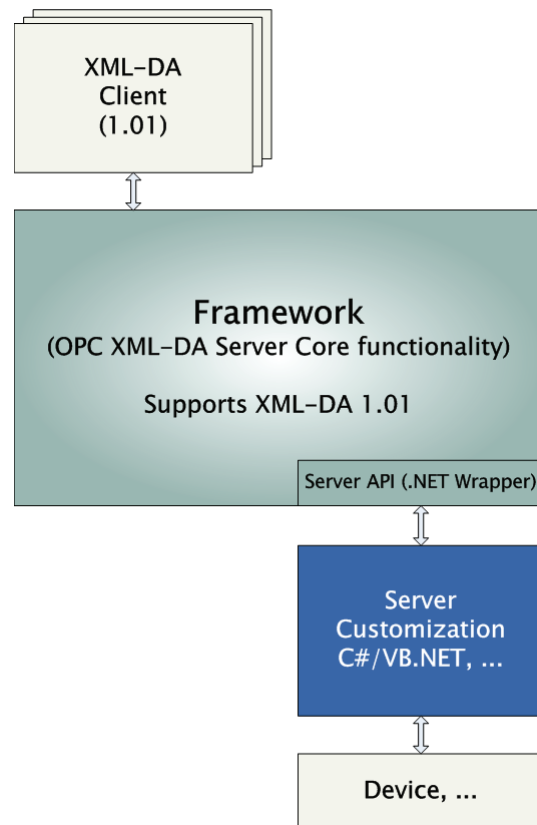
The **“Framework”** includes all OPC XML-DA 1.01 web service handling and ensures the OPC compliance. It is implemented as a generic .NET based assembly.

The **“Server API”** defines easy to use interface functions required for developing OPC XML-DA compliant servers. The provided sample XML-DA servers make development quick and simple. By using this API XML-DA servers can be easily implemented by adapting just a few functions. Nevertheless the API doesn't restrict the developer and is designed for flexibility and speed. Small XML-DA Servers with a few items as well as large XML-DA Servers with thousands of items are supported.

The OPC XML Server Developer Studio .NET offers unique features for performance and functionality improvements of the developed OPC Server like Event Driven Mode for Device Access.

A free evaluation version is available at www.technosoftware.com featuring full functionality including the DLL and source code of all provided sample servers. It can be used for 14 days.

“The OPC XML Server Framework offers an object-oriented .NET VB/C# and Delphi application framework for simple development of XML-DA compliant servers.”



Picture 3: OPC XML Server Developer Studio .NET Structure



Features Include

The OPC XML Server Developer Studio .NET includes features like:

- ✓ Short time-to-market for standard compliant products. Develop OPC servers in days instead of weeks or months.
- ✓ Highly utilizable tool for quick and easy development of OPC compliant servers.
- ✓ Powerful possibilities for performance optimizations.
- ✓ Event Driven Mode for Device Access.
- ✓ Custom Properties.
- ✓ Server is implemented as an ASP.NET Web Service.
- ✓ Servers are fully compliant to the OPC Specification XML-DA 1.01.
- ✓ Tested with the OPC Foundation Compliance Test Tool.
- ✓ The application specific part can be implemented with C#/VB.NET or any other compiler capable of generating a .NET assembly.
- ✓ Extensive diagnostic and installation support.

Self-Tested for Compliance

The OPC Foundation Self-Tested for Compliance logo indicates that these products have been tested by the manufacturer to be compliant with the following OPC Specifications: XML-DA 1.01. Additional information about compliance testing, logo program and a summary of test results for the OPC Server Developer Studio .NET can be found at www.opcfoundation.org.

Product Specification

Product Type

- Server Development

OPC Compatibility

- OPC XML-DA 1.01

Supported Operating Systems

- Windows 2000, XP
- Windows 2003 Server
- Windows Vista™

Development Environment

- IIS
- .NET Framework 2.0
- Microsoft Visual Studio 2005 (C#/VB.NET)

Development Interface

- .NET Interfaces



Order Information

Product ID	Description	List Price (EURO) *
TSOPC-XSDN	XML Developer Studio	2495.-
TSOPC-XSDN-SUPT	1 st year support and updates	795.-

License

- Per-Developer Basis
- One Run-time license included

Delivery

- Downloadable product
- Free Standard Support for a 90 days period included.

* Pricing is subject to change / Call us for updated price



License Model ...

Technosoft's license model is extremely simple and flexible enabling you to quickly start your work. Technosoft license highlights:

- No royalties, run-times fees, or server deployment charges.
- Simple, per-developer licensing model – each product is licensed to a single user.

We recognize that you may use more than one machine for development, and our license permits our products to be installed on multiple PCs. Office computer, home computer, laptop – one price covers all locations.

Some software companies price their OPC server and client development components at a lower market price, but we encourage you to examine the total cost. Hidden license fees and unexpected deployment costs can quickly push up the cost of working with OPC server and client development components. In contrast, Technosoft's straight-forward license model provides a platform with the flexibility you need to get the job done - with no hassle and within your budget.

A license is needed for each developer who will be working with our products, and volume discounts are available. If you would like to see a custom quotation that includes volume pricing, please contact us at sales@technosoft.com.

Subscription Coverage ...

Each package includes a ninety days or one year Subscription for technical support and access to new releases.



Why Technosoftware ...

- ❖ You aim at a successful partnership, to which you make the demands and we take over the realization.
- ❖ You want to introduce new technologies, e.g. OPC, with the support of an experienced specialist in your team.
- ❖ You want to optimize costs, by using our already existing know-how, instead of starting from the ground up.

For Technosoftware AG the satisfaction of the customer stands in the center. We reach this by a small, competent and dynamic team of coworkers, which apart from the satisfaction of the customer; take care of a high quality of work. We concern the steps necessary for it together, under installation course of the customer.

**Use our know-how
to shared success!**

Technosoftware AG

Farmweg 4

5702 Niederlenz

Schweiz

Tel.: +41 62 888 40 40

Fax: +41 62 888 40 45

sales@technosoftware.com

www.technosoftware.com



10 years experience with Server and Client Software Developer tools for OPC gave us an understanding for the needs of our customers.

Flexibility | Efficiency | Quality