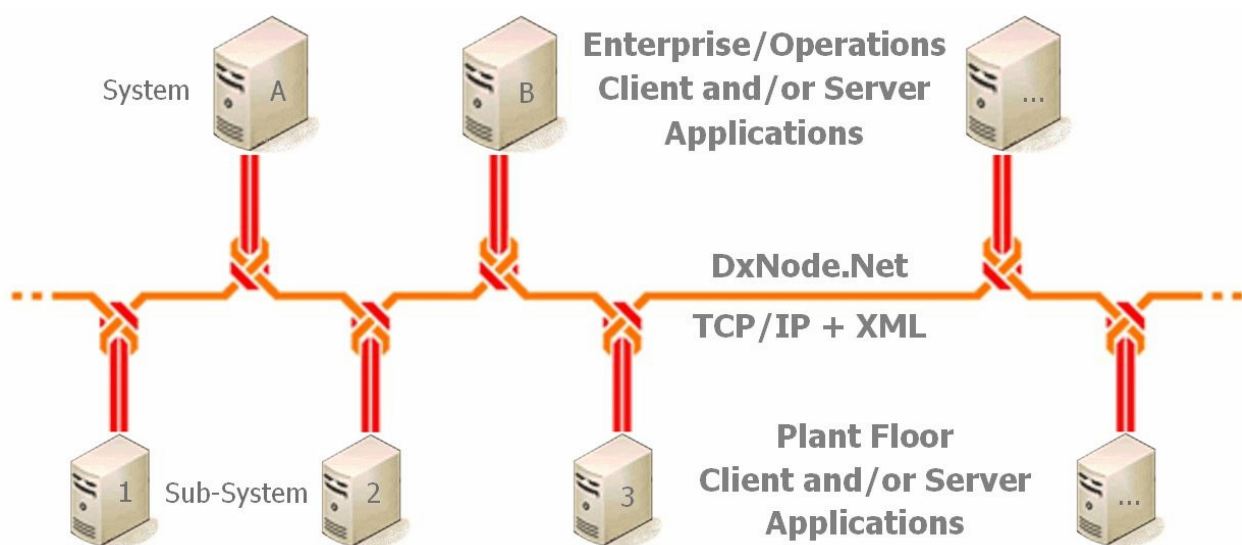


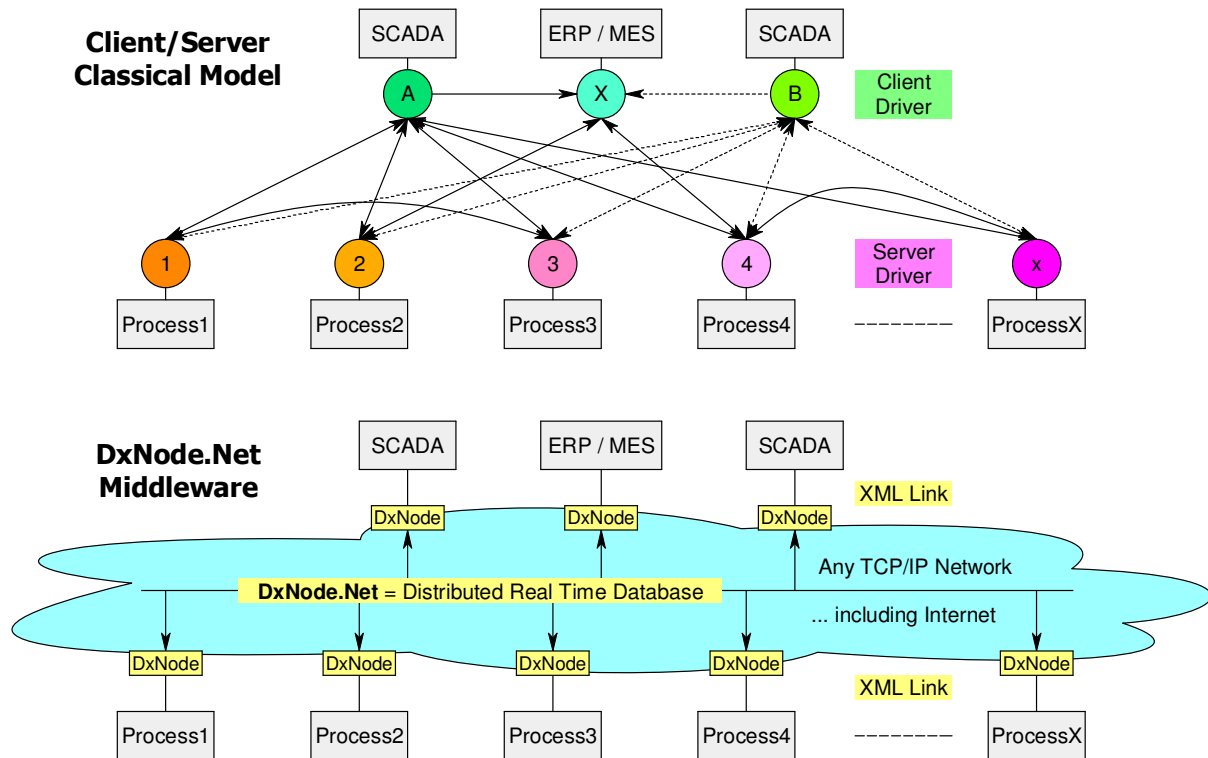
DxNode[®] Net Whitepaper

Data exchange **Node Network**



What is DxNode.Net ?

DxNode.Net stands for "Data eXchange Node Network", it is middleware that was developed as an independent open real time database using international standards only. In contrast to the classical client/server model used for data exchange between multiple heterogeneous systems, DxNode.Net unifies all interfaces and features many advantages that are explained below. For the classical model, every partner is either a data client or a data server that requires an individual driver interface:



Consider DxNode.Net an independent open distributed real time database where multiple DxNodes are interconnected and can play the role of a data client and/or a data server as desired. This simplifies the applications in that only one standardized XML interface is required to allow for reading and/or writing data from and to the database. Any partner can exchange data with any other partner without the need to know who is connected to read and/or write data. DxNodes can exchange data through the Internet using Web Services and, every single DxNode can simultaneously be used as a local real time database to interconnect multiple local processes.

DxNode.Net is independent on third party software and – apart from the operating system – it is based only on international standards as **TCP/IP** (Data Transmission Control Protocol, Internet Protocol), **XML** (eXtensible Mark-up Language) and **Web Services** (Internet Access).

DxNode.Net is self-sufficient that is, its operation is not dependent on the external processes or devices. Whenever two systems are connected, then they can exchange data without the need of a common server. DxNode.Net is not only a data transport layer but it supports a number of standardized functions that help to simplify the interface to applications:

- Any DxNode supports redundant systems and connections in that you simply can connect multiple parallel running clients and/or servers.

- ✘ DxNode.Net is totally event driven (no loss of events) and supports automatic time stamping that includes a time correction value, if the clock of a partner system is not synchronized.
- ✘ DxNode.Net supports so called Store&Forward messages by buffering data on hard disk to guarantee no loss of events. Whenever a connection was lost, a DxNode will automatically send all relevant historical data and re-synchronize the partner node and all its connected systems.
- ✘ DxNode.Net supports a method for bi-directional data exchange which allows writing commands without locking the partner system. The Method is also used for data synchronization in that every node can play the role of a data server or a data client for each data point.
- ✘ Every connection to a DxNode automatically creates internal data points for supervision and monitoring the link. These data points can be used for control and statistical information.
- ✘ The entire configuration of a DxNode including all interfaces is specified by a standardized XML file thus, no additional configuration file is required for the external application. Configuration includes signal marshalling, filtering and two address spaces for mapping user specified local and network names. DxNode.Net also supports online configuration and downloading of configuration data.
- ✘ DxNode.Net is based on TCP/IP and XML that is used for all: node configuration, communication protocol (local and network telegrams) and data storage (hard disk). That is, all information is transparent and can be monitored by XML editors.
- ✘ DxNode data is compatible to OPC (value/timestamp/quality). DxNode.Net supports both, OPC Client and OPC Server interfaces. Other than OPC, DxNode.Net works with no loss of events and, it supports store&forward messages as well as redundant systems and connections including data encryption if specified.
- ✘ Each DxNode is started automatically as a service or manually as any other executable program. Only two files are required to run: the program (.exe) and the configuration (.xml). XML configuration and XML telegrams can be validated for compliance by an XML Schema (.xsd). The XML Schema may even be customized to meet user specified standards.

Where can DxNode.Net be applied ?

DxNode.Net is available for Microsoft Windows and Linux based systems. The program DxNode.exe incl. libraries require approx. 500 KB of memory plus 200 Bytes per Datapoint at runtime. A single node can support 100'000 data points and may connect up to 100 other DxNodes or processes. Thus, you can use DxNode.Net for a wide range of applications, starting at small embedded systems up to large control systems that are linked through Internet. DxNode.Net may be regarded a "world wide distributed real time database" that can be used for any purpose in any area, for example:

- ✘ Enterprise Resource Planning (ERP)
- ✘ Management Execution Systems (MES)
- ✘ Supervisory Control and Data Acquisition Systems (SCADA)
- ✘ Production and Process Control Systems
- ✘ Building Control Systems
- ✘ Traffic Control Systems
- ✘ Process Data Collection and Distribution
- ✘ Any system, where real time data transport from/to other systems is required

How does DxNode.Net compare to other products ?

A DxNode is a complete re-loadable and self-sufficient application (service or program) that uses a single executable for all functions - it is not just a specification. The entire configuration, data storage and data exchange protocol are based on open standards XML and TCP/IP – this is unknown to other products that are based on OPC specifications, BACnet, PROFINET, etc. DxNode.Net was designed to support all properties and functions of OPC including the planned OPC Unified Architecture. However, DxNode.Net supports some more features:

- ✘ DxNode.Net is not an interface specification but a ready to install and run product
- ✘ DxNode.Net supports x-fold redundant or parallel running systems and connections through Internet including store & forwarding messages to/from hard disk.
- ✘ DxNode.Net supports a method for commands (bi-directional data exchange) that includes automatic transition indication in order to provide most comfortable operation even at very slow transmission rate. The method allows to chain server nodes in a complex network (similar to the World Wide Web) without the loss of operator comfort. In fact, nodes are loosely coupled and do not require record locking as known for other systems.
- ✘ DxNode.Net supports two address spaces that allow customizing Datapoint addresses throughout the network while keeping vendor specific local addresses as required.
- ✘ Any data transaction is fully event driven, though it supports polling or sampling of data by any application. Because there is no loss of events, DxNode.Net can replace OPC DA, DX, DA/XML and A&E in one, that is, DxNode.Net does all without the need of multiple specifications by a simple unified XML protocol.
- ✘ Data can be encrypted and filtered online for any connection, filters include timestamps (less and greater than) and value status (user specified quality flags).
- ✘ Compliance tests can be performed by verifying the XML configuration and XML telegrams on the fly by a single XML Schema. The schema can be customized to meet user specified standards.
- ✘ Any DxNode can be used as a local real time database with any number of connections (tasks, processes) and local or common data points. And, a DxNode can simultaneously communicate through Internet using standard Web Services on the hosting computer.
- ✘ DxNode.Net is independent on specific DLL's or Microsoft's DCOM thus, it can be ported to other operating systems. Data access is performed by user specified ports, not by an automatic port mapper as known for DCOM, which makes it difficult for firewall protection.