



Movicon NExT

3.0 Data Server

Ver.3.4.268

Table of Contents

1.1. GENERAL DATA SERVER CONCEPTS.....	1
2. I/O DATA SERVER.....	3
2.1. DATA SERVER WORKSPACE	3
2.2. HISTORIAN MANAGEMENT	3
2.3. ENGINEERING UNITS.....	7
2.4. GENERAL SERVER SETTINGS.....	8
2.5. ADVANCED SERVER SETTINGS	10
2.6. COMMUNICATING WITH DEVICES	13
2.7. DATA TRANSPORT DEFINITIONS	14
2.8. SERVER STATUS AND LOG.....	17
2.9. CLIENT CONNECTIONS TO SERVER.....	17
2.10. DB CONNECTION SETTINGS WINDOW.....	20
2.11. PC TIME CHANGE	21

1.1. General Data Server Concepts

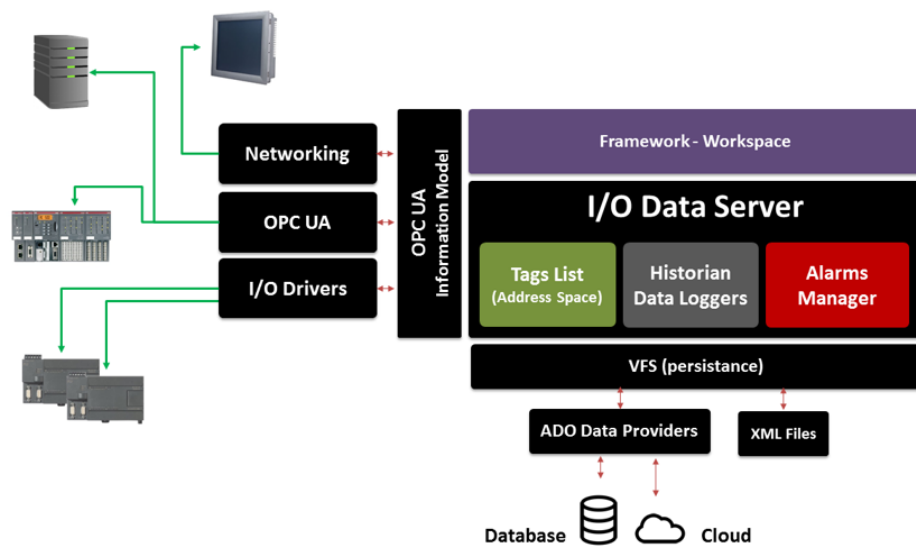
The Platform.NExT I/O Data Server module is an essential feature to the platform's projects as a communication and data manager server. The Data Server manages communication with the field and third party systems by defining and centralizing information dynamically using Tags described in the project's Tag List resource (Address Space).

In addition the Server manages alarms and records events in the Historical Log.

Ultimately, the Data Server can historically data log process data collected in Tags using the Historian and Data Logger archive manager as an option.

As shown below, the Platform.NExT Server module is designed to manage:

- **Data Communication**
- **Alarm Manager**
- **Historical Log management**

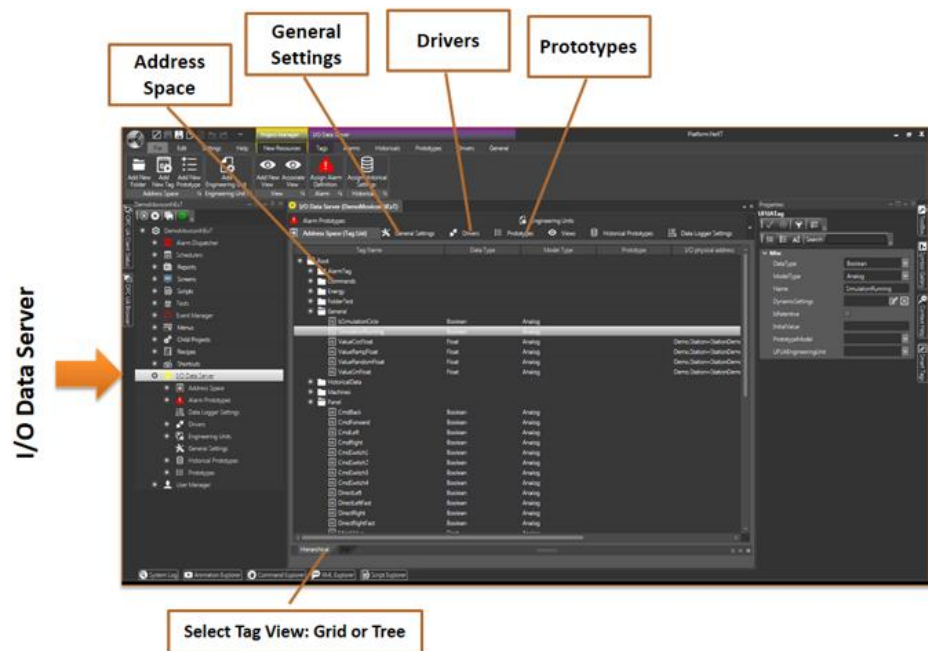


This diagram shows the structure of the Data Server which manages communications, the alarm manger and historicals.

2. I/O Data Server

2.1. Data Server Workspace

The I/O Data Server module is available from the project resource tree structure. When opening this resource, the workspace open up displaying the general settings and providing access to all the Server's functionality configurations that include the Tag List and others provided with the module as described in this guide.



The Tag List can be set as pleased to display in a tree structure containing Tag folders listed in hierarchical order or in a grid to allow operations such as selecting and filtering to be performed.

2.2. Historian Management

Among other things the I/O Data Server module also manages the recording of data in the historical log archive using the modes and configurations described in the relating chapters.

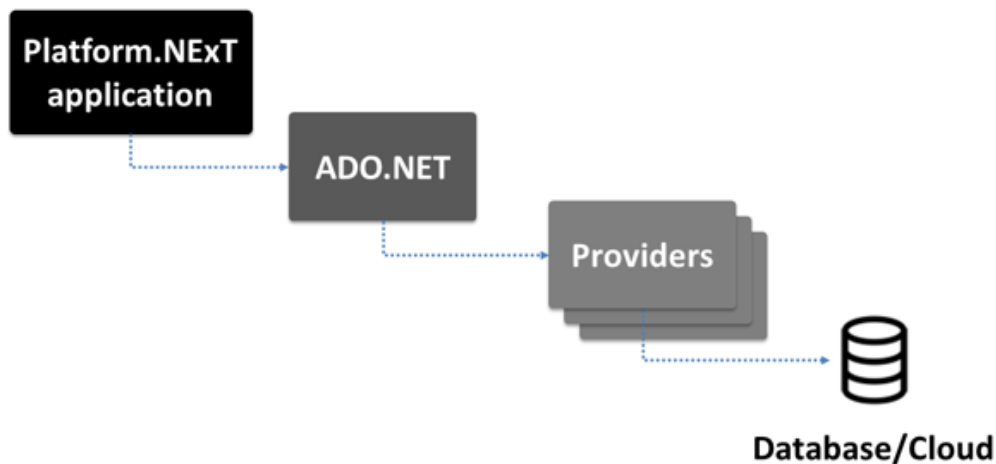
The Server manages data recording by using the following functional modules:

- **Event and Alarm recording**
- **Historical Process Data recording**
- **Data Logger recording**
- **Recipe Management**
- **DB Connectable Toolbox objects (grids, dataloggers etc...)**

Each functional module independently records data and accesses its database using the Platform.NExT platform infrastructure.

Connecting to Databases

The Platform.NExT I/O Data Server module uses the ADO.NET Data Provider technology to connect to the databases. By means of using this technology each project and each data recording resource is independent from the data format that you have chosen to use.



The Connection Strings

By using **Data Providers**, the Data Server can be totally independent from the database type used within the architecture. The connection string can be customized using the Platform.NExT Data Server's configuration properties in order to use the preferred database.

If not specified otherwise, the Data Server comes with a connection string for default to ensure that data is recorded in the SQL Server database automatically.

However, this string can be customised to define a different data Provider in order to access the database of your choice.

The Connection String should therefore contain the information that the Provider will need in order to establish the connection to the database file and for managing the relevant commands in a way that is totally transparent and independent from the project.




Since there are different Data Providers to choose from, the user should check how to set the connection string in function with the specific Provider they wish to use.


This may differ according to which Provider the user wishes to connect to for Oracle, MySQL or others.

As explained previously, use the button on the side to select the desired "Sql Server", "SQL Azure", "MS Access", "XML file" data source. .

It is also possible to modify the connection string manually to expand the connection towards other DataBases.

Database Engine	Version(s)	XPO Data Store Adapter	Connection Strings
Advantage	Server v11.1	AdvantageConnectionProvider	XpoProvider=Advantage;Data Source=\\myserver\\myvolume\\mypat\\mydd.add;ServerType=local;

			UserID=ASSSYS;TrimTrailingSpaces=true
ASA	SQL Anywhere 16, SQL Anywhere 17	AsaConnectionProvider	XpoProvider=Asa;Uid=MyUsername;PWD=MyPassword;DBF=c:\mydatabase.db;Persist Security Info=true
ASE	Sybase Adaptive Server 12, Sybase Adaptive Server 15.5, Sybase Adaptive Server 16	AseConnectionProvider	XpoProvider=Ase;Port=5000;Data Source=MyAseServer;User ID=MyUserName;Password=MyPassword;Initial Catalog=MyDatabase;Persist Security Info=true
DB2	DB2 9.7.4	DB2ConnectionProvider	XpoProvider=DB2;Server=MyAddress:MyPortNumber;User ID=MyUserName;Password=MyPassword;Database=MyDatabase;Persist Security Info=true
Firebird	Firebird 1.5, Firebird 2.5.7	FirebirdConnectionProvider	XpoProvider=Firebird;DataSource=localhost;User=SYSDBA;Password=masterkey;Database=MyDatabase.fdb;ServerType=0;Charset=NONE
MSAccess (32bit Systems)	Access Database Engine	AccessConnectionProvider	<p>XpoProvider=MSAccess;Provider=Microsoft.ACE.OLEDB.12.0;Mode=Share Deny None;data source=C:\Temp\TestDB1.accdb;userid=Admin;password=;</p> <p> You must install 32 bit access drivers versions: https://www.microsoft.com/en-us/download/details.aspx?id=1 https://www.microsoft.com/en-us/download/details.aspx?id=13255 (Access 2016)</p> <p> To set the data logger connection strings successfully:</p> <p>DataProvider=System.Data.OleDb;Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Temp\TestDB1.accdb;</p>
MSAccess (64bit Systems)	Access Database Engine	AccessConnectionProvider	<p>XpoProvider=MSAccess;Provider=Microsoft.ACE.OLEDB.12.0;Mode=Share Deny None;data source=C:\Temp\TestDB1.accdb;userid=Admin;password=;</p> <p> You must install 64 bit access drivers (AccessDatabaseEngine_X64.exe) for versions: https://www.microsoft.com/en-us/download/details.aspx?id=13255 (Access 2010)</p>

			<p>https://www.microsoft.com/en-us/download/details.aspx?id=54920 (Access 2016)</p> <p> To set data logger connections, the setup window can be used to obtain configuration strings such as this one:</p> <p>DataProvider=System.Data.OleDb; Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Temp\TestDB1.accdb</p>
MSSqlServer	Microsoft SqlServer 7.0, Microsoft SqlServer 2000, MS SQL Server 2000 Desktop Engine (MSDE 2000), Microsoft SQL Server 2005, SQL Server 2005 Express Edition, SQL Server 2008, SQL Azure™ Database, SQL Server 2008 R2, SQL Server 2008 R2 Express, SQL Server 2012, SQL Server 2012 Express (including LocalDB), SQL Server 2014, SQL Server 2014 Express (including LocalDB), SQL Server 2016, SQL Server 2016 Express (including LocalDB)	MSSqlConnectionProvider	<p>XpoProvider=MSSqlServer;Data Source=(local);UserID=username;Password=password;Initial Catalog=database;Persist Security Info=true</p>
MSSqlServerCE	Microsoft SqlServer 2005 Mobile, SQL Server 2005 Compact Edition (Microsoft SqlServer	MSSqlCEConnectionProvider	<p>XpoProvider=MSSqlServerCE;Data Source=MyDatabase.sdf;Password=MyPassword</p>

	2005 Everywhere Edition CTP), SQL Server Compact 3.5, SQL Server Compact 4.0		
MySql	MySQL Server 4.1, MySQL Server 5.0, MySQL Server 5.1, MySQL Server 5.7	MySqlConnectionProvider	XpoProvider= MySql;Server=MyServerAddress;User ID=MyUserName;Password=MyPassword; Database=MyDatabase;Persist Security Info= true;Charset=utf8
Oracle	Oracle 9i, Oracle 10g, Oracle 11g, Oracle 12c	OracleConnectionProvider ODPConnectionProvider ODPManagedConnection Provider	XpoProvider= Oracle;Data Source=TORCL;User ID=MyUserName;Password=MyPassword
Pervasive	Pervasive PSQ 9, Pervasive PSQ 10, Pervasive PSQ 11	PervasiveSqlConnectionProvider	XpoProvider= Pervasive;Server=MyServerAddress; UID=MyUserName;PWD=MyPassword;ServerDSN=MyDatabase
PostgreSQL	PostgreSQL 7, PostgreSQL 8, PostgreSQL 9, PostgreSQL 10	PostgreSqlConnectionProvider	XpoProvider= Postgres;Server= 127.0.0.1;User ID=MyUserName;Password=MyPassword; Database=MyDatabase;Encoding=UNICODE
SQLite	SQLite 3	SQLiteConnectionProvider	XpoProvider= SQLite;Data Source=filename
VistaDB	VistaDB 4, VistaDB 5	VistaDBConnectionProvider	XpoProvider= VistaDB;Data Source=C:\mydatabase.vdb4 XpoProvider= VistaDB5;Data Source=C:\mydatabase.vdb5



You should install the Connector/NET 6.9.9 version when using MySQL.



The 'default Historian Connection' string will also be used by the Historical, Data Logger, Reports and all other screen objects displaying data retrieved from the DB unless another customized connection string has been set in their properties.

2.3. Engineering Units

The Data Server's **Engineering Units** window allows you to insert and configure data conversion models, from an input value to a calculated and scaled output value.



The Engineering Units are also important for the automatic configuring of scale values used in many of the user interface objects. For instance if the project has a tag that has been associated to a gauge object (a display object with a needle), it will automatically adapt to the Min. and Max. scale values as set in the Engineering Unit's Low Range and High Range fields.

General

Name

This is used to define the name to be assigned to the Engineering Unit

Unit Measure Label

This is used to define a text value (or an acronym) to assign to the engineering unit.

Scales

Engineering Unit Low Value

This is used to define the low output value, calculated on a ratio corresponding to the raw input value.

Engineering Unit High Value

This is used to define the high output value, calculated on a ratio corresponding to the raw input value.

Low Raw Input Value

This is used to define the low raw input value with which the engineering unit output value will be calculated.

High Raw Input Value

This is used to define the high raw input value with which the engineering unit output value will be calculated.

2.4. General Server Settings

The Server's **General Settings** window is used to configure the general settings that determine how the Data Server functions.

General Settings

The I/O Data Server's General Settings window contains the modules general settings. Each individual setting parameter has been set with a default value that is optimal for most cases but can be changed as required.

Application Name:

The Name of the Server application. This name can be customized and will be displayed in the selection list when browsed by Clients (eg. third party OPC UA Clients).

In addition to the Application Name other information such as "Manufacturer Name", "Product Name", "Product URI", "Software Version" and "Build Number" can also be entered. This information will then be published by the Server as additional information or for managing Security Certificates.

Manufacturer Name

Allows you to enter the manufacturer's name (published by the server)

Product Name

Allows you to enter the Product name (published by the server)

Product URI link

Allows you to set an URI for the product/manufacture (published by the server)

Software Version

Allows you to enter the software Version (published by the server)

Build Number

Allow you to enter product build number (published by the server)

Historian Default Connection:

If not specified otherwise, this is the connection string that will be used for default by the Historian and DataLogger Prototypes for recording data by means of using the Server's Historian module.

When left with the default setting, the Server will use the Microsoft SQL Server technology. If you wish to connect to another database, you will need to use an adequate ADO.NET DB Provider according to the information relating to the connection strings towards the desired DB.



See concepts on connecting to Databases using ADO.NET Data Providers in this manual.

Events Default Connection:

If not specified otherwise, this is the connection string that will be used for default by the Events and Alarms (Historical Log) recording engine.

When left set with the default setting, the server will use the Microsoft SQL Server technology.

If you wish to connect to another database, you will need to use an adequate ADO.NET DB Provider according to the information relating to the connection strings towards the desired DB.



See concepts on connecting to Databases using ADO.NET Data Providers in this manual.

Events Max Age:

Maximum storage life of data recorded in the Events and Alarms (Historical Log) Database. This can be set to the length of time you wish to store data in the archive. The default value is set to 365 days. Therefore, once this time has been reached, the system will start recycling by overwriting the oldest data with the most recent data.

Diagnostic Enabled:

This enables the publishing and managing of diagnostic information on the server to which clients can access.



Attention! The activation of this option may have an impact on server performances. Therefore, it should be used with caution and only when absolutely needed in addition to debugging. For example, when enabling this option, the server will take longer to startup with projects containing big volumes of Tags that have been defined in the Server's Address Space.

2.5. Advanced Server Settings

There is a series of Advanced Server Settings that can only be accessed by using the Property Window. This window permits expert users to change certain settings that determine how the Data Server functions.

General

For further information please see the chapter on "General Settings".

Database Settings

These settings are used to establish a connection towards the DB for recording historical data.

Historian Default Connection

Sets the predefined connection string to be used by all the resources based on the historical log DB.

Event Default Connection

Sets the predefined connection to be used by all the resources based on the historical log DB.

Events Mas. Age

Sets the max. age that data can remain in the Even DB.

Historical Wait Retry

Sets the waiting time in seconds between one attempt to connect to the DB and the next.

Historical Max. Retries

This is used to set the max. number of attempts to access the Historical DB before starting to unload data from the cache file to a local text file.

Advanced Database Settings

When selecting the Data Server and accessing its Property Window you will notice that it has additional settings called 'Advanced'. These settings are used to configure the I/O Data Server. Each single setting parameter has been preset with a default value which in most cases is optimal. However these parameters can be changed according to specific user needs.

Max. Concurrent Historical Access

This is used to set the maximum number of concurrent accesses to the Data Server's historical. Any accesses exceeding this limit will be put on hold until one of the ongoing accesses has concluded.

Historical Cache Size

This is used to set the maximum cache size containing failed data entry attempts. Once this value has been exceeded, all data pending a new attempt to be entered on the database will be flushed onto a text file in XML format in the local disk where the Server is running.

Min. Historical Pending Entries

This is used to set the minimum number of data in the Cache pending Database entry. When the number of data is less than the value set here, data flushed to a predisposed local XML text file can re-enter into the Cache.

Max. Historical Pending Entries

This is used to set the maximum number of data in the Cache pending Database entry. Once this limit has been reached, the additional data will be flushed to a predisposed local XML text file. If this predisposed text file also reaches its set limit, data will be irretrievably lost.

Max. Historical Deleting Entries

This is used to set the maximum number of records to be deleted from the database each time a maximum age database control is performed.

Max. History Flush File Size

Maximum size in MBytes of all Flush files. Once this limit has been reached, the oldest files exceeding this value will be permanently removed. This will be traced in the log with this message: "The maximum size of safe files allowed has been exceeded causing the permanent removal of some files from path '{0}' !"

Max. History Alarm Branches

Sets the maximum number of alarm events to be loaded in the buffer.

Max Session Timeout

This sets the maximum time that a Client session can stay open even when not communicating with the Server. It is advised to leave this set with the default value.

Min. Session Timeout

This sets the minimum time that a Client session can stay open even when not communicating with the Server. It is advised to leave this set with the default value.

Max. Browse Continuation Points

This is the maximum number of continuation points used for browsing operations by clients towards Server.

The continuation point is necessary for performing subsequent requests to get remaining information. This is useful when the amount of information received by the client may be incomplete due to a limited maximum amount of information the client can receive in one individual interrogation.

Max. History Continuation Points

Maximum number of continuation points (as above) used for HistoryRead operations.

Max. Request Age

Maximum duration of incoming requests. Take note that the oldest requests are deleted.

Execution

This group of properties concern those settings for the diagnostics and speech functions.

Diagnostic Enabled

This is used to enable the publish and manage information on diagnostics in the Server to which clients can access. Activating this option may have an impact on server

performances therefore it should be used with great care only when really necessary and for debugging purposes.

Speech Enabled

Enables the speech function for system message notifications.

Speech Voice Name

Sets the name of the speech engine.

Max. Session Count

Used to set the maximum number of concurrent open sessions.

Default Data I/O Sampling Interval

Used to set the default minimum refresh rate for Tag value notifications from the server to the client. If no refresh time has been set in the client, the server will use this value set in milliseconds.

OPC UA Server Option

Parameter settings for sessions to be managed by the server in client-server communications.

Min. Publishing Interval

The minimum time in minutes to publish data supported by the server.

The server resets subscriptions that have a lower interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

Max. Publishing Interval

Maximum time in minutes to publish data supported by the server.

The server resets subscriptions that have a higher interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

Publishing Resolution

This parameter indicates the minimum difference in minutes between the supported publishing intervals.

This is always used for calculating the subscription interval that is returned by the server in respect to the one requested by the client.

Max. Subscription Lifetime

Defines how long subscriptions can remain open without being published by client.

Max. Message Queue Size

Defines the maximum number of messages saved in the queue for each subscription.

Max. Notification Queue Size

Defines the maximum number of notifications saved in the queue for each monitored item.

Max. Notification Per Publish

Defines the maximum number of notifications saved in the queue for each publication.

Min. Metadata Sampling Interval

Defines the minimum metadata sampling interval.

Redundancy

This section contains various configurable redundancy properties and options. For further information on editable properties in this section please refer to the chapter on "Redundancy Settings".

Audit

Audit Trace Default DB Connection

This property field is used to set a different connection string as an alternative to the one used by the project for default and represented by the Historian Default connection string in the IO Data Server's general settings. When set it will be defined with an alternative database within which a UFUAAuditDataItem table will be created to trace recordings of those tags being audited.

Enable Data File Protection

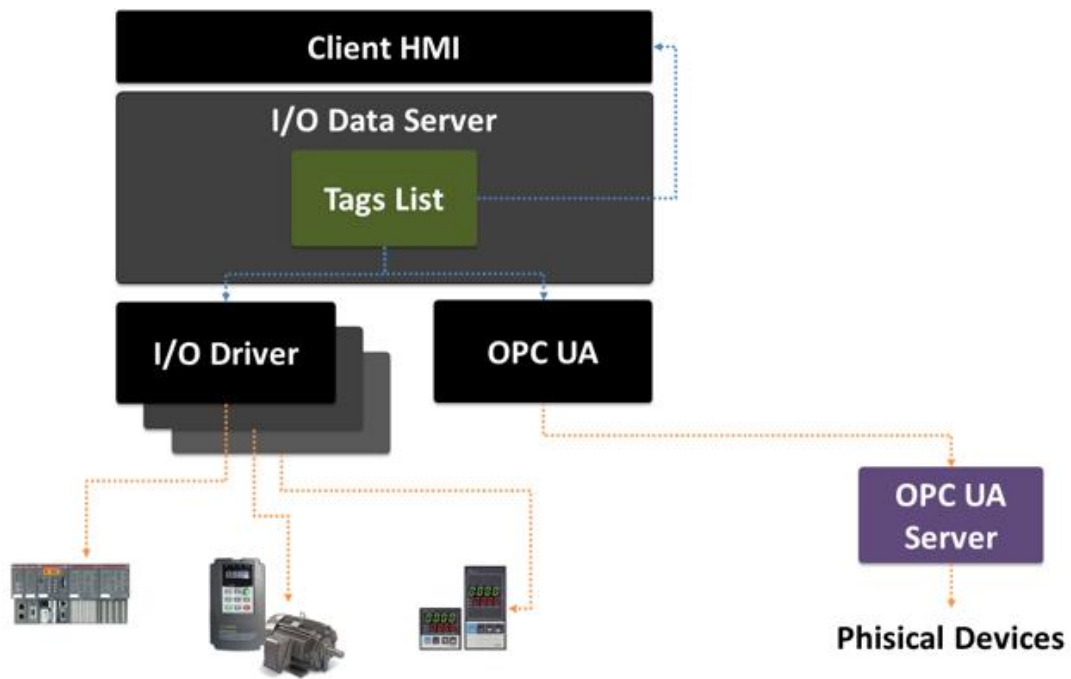
This property field is used when wishing to protect audit recordings from tampering, unintentional or intentional modifications of any kind. This is done by inserting a column with encrypted data in the UFUAAuditDataItem table. The information contained in this column will prevent the mishandling of sensitive data recordings without compromising data integrity.

2.6. Communicating with devices

One of the main tasks of the I/O Data Server module is to manage communications with the field devices using different communication channels that manage protocols towards the physical I/Os of external peripherals.

The platform is therefore capable of communicating in real-time with field devices using communication drivers that are included with the product or technology that connects to third party OPC UA Servers. Whatever mode is used all the information collected from the field physical addresses is associated to Tags which are found in the Tag List resource used for the Server's Address Space.

The Tag List renders all real-time information visible to all the resources and modules of the entire Automation Platform.NExT platform whether Server or Client.



This diagram shows an example of an architecture using I/O Drivers or the OPC UA technology to communicate.

The Driver configuration specifications are described in the relating sections of this documentation according to each driver's specific communication protocol type (e.g. Siemens S7, Ethernet-IP, Modbus, Omron and others) along with the OPC UA standard communication technology.



The I/O Data Server Communication Drivers are installed with the actual Server. However you can install additional drivers by downloading them from the Progea website directly. Those created by third parties can also be used.



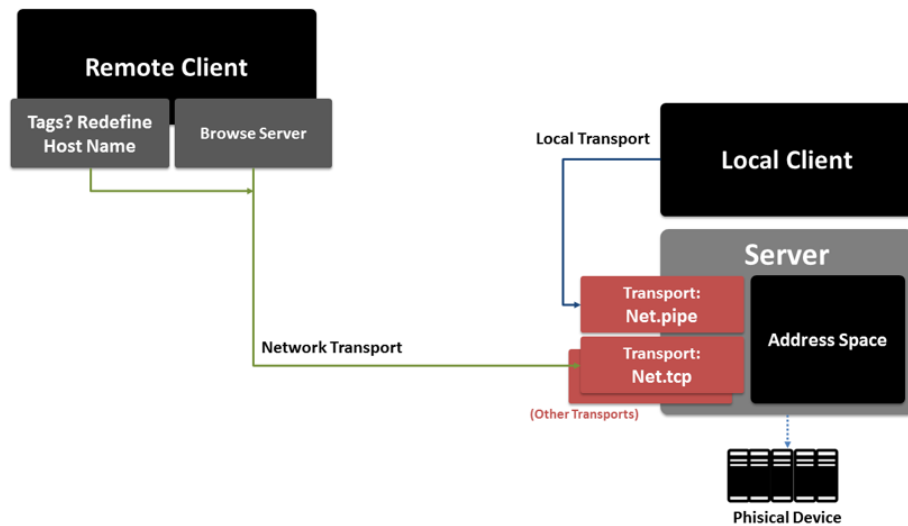
Platform.NExT has been designed to enable expert users to create their own specific communication drivers and integrate them within the platform. In order to help users to do this Progea offers a SDK specifically dedicated to developing custom protocols. They also provide support and advice when needed.

2.7. Data Transport Definitions

In I/O Data Server settings you will find a window containing a list of Transports which are available for Server project data communications.

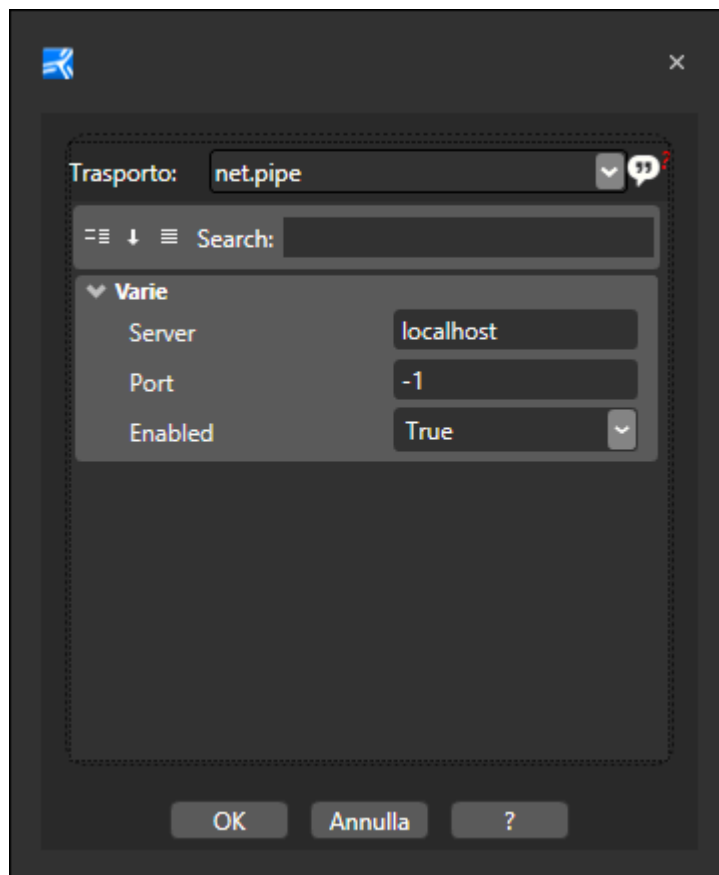
This window can then be used for inserting the transport criteria to use for Server and Client data communication. Once the Server has been started up, it will provide the end points according to the transports which have been enabled in the project.

The Clients (any Client) can then use the preferred transport, or one that is the most suitable for the application type being used, independently from the data model being used.



The Platform.NExT Data Server has been specifically designed to make data independent from the Transport type selected to enable distributed Client-Server architectures to use the one most suited to the application type being created.

By using the Data Server settings window you can enter, delete or edit the transports available to the Data Server so that they can be used by clients wishing to connect. This operations are done using the "Add Transport" or "Edit Transport" window:



This window allows you to choose and configure the Transport you wish to add to the Data Server:

- Transport Type
- Server Name
- Port
- Enabled/Disabled

The local net.pipe transport

Projects are designed with the "net.pipe" transport for default. This transport is the one used in the **local application** for connecting the Client to the Server if not specified otherwise. This means that the visualization Clients assigned with Tags defined in graphical screen objects in the Server will connect using the local net.pipe transport. This transport is one of the those provided by Microsoft WCF (Windows Communication Foundation) and uses local memory to ensure maximum performances.

This transport is for 'local' use only and therefore does not consent to network connectivity. If you need to use network connectivity you must enter one of the other transports provided.



Attention! "net.pipe" is a local transport that entails the Client and the Server run within the same session. This transport cannot be used if the Client project and the Server project run on different machines.

When using a Terminal Server or a Remote Desktop, the net.pipe transport may require user login with an appropriate access level.

Network Transport

To enable remote Clients to connect to the Server a network transport must be configured in the Server.

There are several network transports each one of which differs in aspect according to how they performance or provide security. It is therefore the project engineer to decide which transport to use for their architecture as required or in function with client needs.

The order with which they are used, for instance when a tag connects to a display, is as follows:

- Net.Tcp
- Opc.Tcp
- Https
- Http
- NoSecurityHttp

If only a few of the transports have been enabled, the first available in the project will be used. If none of the transport have been configured, the first one in the Server's configuration file will be used.

2.8. Server Status and Log

The I/O Data Server has a specific log window that allows users to view the Server project's data communication status for diagnostics purposes.

This window activates when starting up the Server manually by using the "Start Server" command and displays Server's status and any error messages.

2.9. Client Connections to Server

The Platform.NExT platform uses an architecture based on Client and Server with a data information model based on the OPC UA standard. This model ensures users maximum data exchange interoperativity allowing Clients to connect to Data Servers as required by being totally independent from whichever transport they wish to use, whether it be a data exchange model, Client NExT connectivity or third party Clients.



Attention! The Client to Server connection must be enabled on the Server License. Except for special licenses, the Networking feature is usually enabled and consents Movicon.NExT Client access to Platform.NExT Servers. Third party OPC UA Client access requires the enabling of the OPC UA Server option on the purchased license.



Please be reminded that the Local Client connection is always allowed and is provided for default. However a Client-Server license is needed for local Client-Server applications along with local connection based on the "net.pipe" transport which is available in the default project configurations.

Transport definition

In order to obtain a Remote Client connection the first thing to do, other than enabling this option on the license, is to define the desired Transport on the Server so that it becomes available in the Data Server's General Settings.

Typically, a connection between a Movicon.NExT Client and a Platform.NExT project's Data Server can be based on the "Net.tcp" transport being the most popular. However the Server is also programmed to use other Transports, as indicated in the corresponding paragraph for security reasons or specific needs.



Attention! A least one internet transport must be defined in the Server to enable the Client to connect (e.g. 'net.tcp').

Associating data from a Client towards a Remote Server

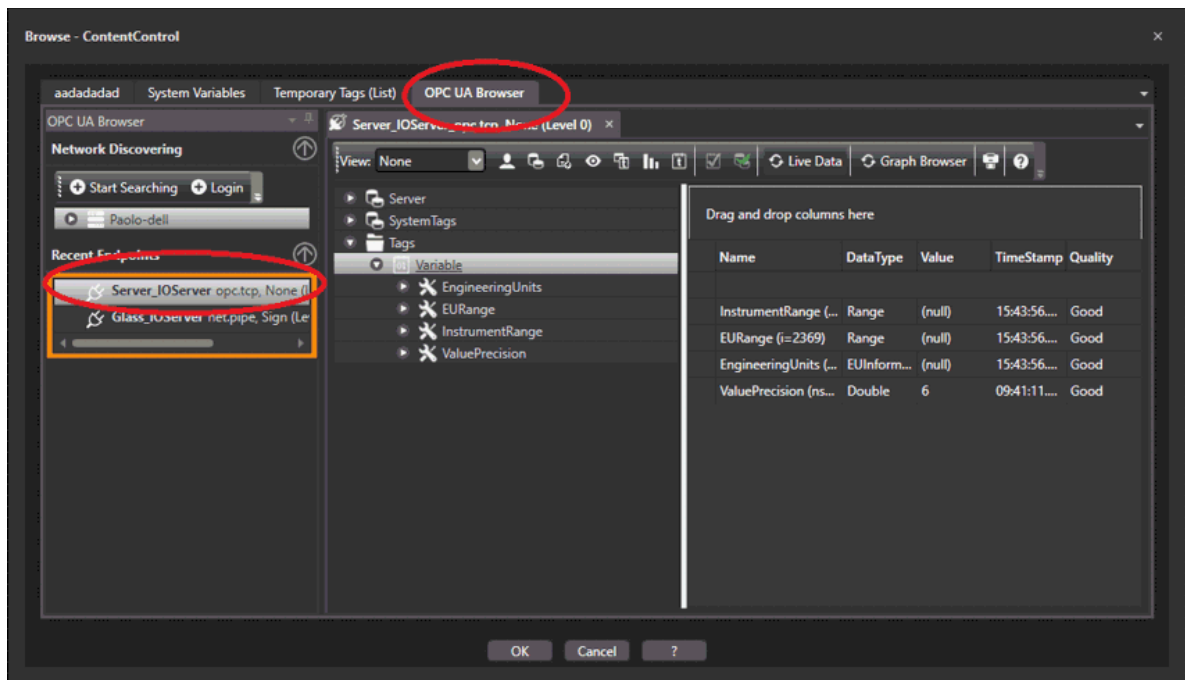
The Data Server exposes its Tags using the preferred internet transport installed on the Server according to the OPC UA standard information model.

All the information available in the Server Tags can therefore be accessed by any Client, either from Movicon.NExT screens or from third party OPC UA Clients. The connection standard is the same even though different types of transports can be used.

If you wish to create a Client visualization project using the Movicon.NExT screens, you can associate the variables to the graphical objects using the Tag Browse window. Each graphical symbol or object can be associated with tags from Local Servers or by using the connection standard. The "OPC UA Browser" can also be selected from the same

Browser window and the Data Server can be selected from those available in remote. After which, the item to connect to (corresponding to the Server Tag) can then be selected.

- **In this case, the Client project will not be defined with any data locally but will be connected directly to the Server for those items which have been selected.**
- **However, if you have a Client that is a copy of a project defined with data of tags previously connected to a local server, you will need to refer to the relating paragraph for details.**



This image shows the Browse window for associating Tags to objects using the OPC UA Browser.

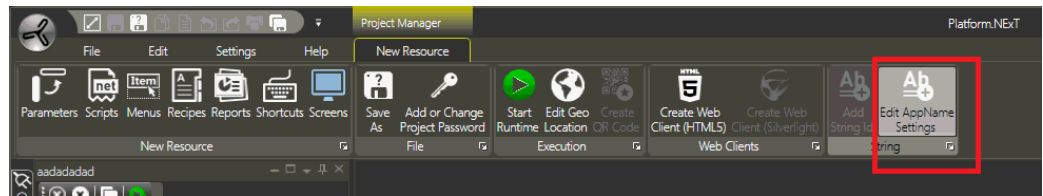
Redefining local Client Tags towards another Server

When you need to redefine the address of a remote Server to point to from an already exiting project that is therefore already connected to Tags defined locally in the address space, you can use its reassign tag function.

In fact, all the tags of screens connected to the Local Server's Tag List use the "net.pipe" for default and are connected to the project's referenced Local Server. When the Client is moved to another station, this Remote Client station will then need to point to the Server station in which the data resides.

However the data definition consists of a series of parameters based on the OPC UA specifications. The correct and most simple way to inform the Client project where to point to in order to access the server is to use the "Edit AppName Settings" tool. This tool directs the connections to the server's new address to be pointed to instead of the address of the server's original position.

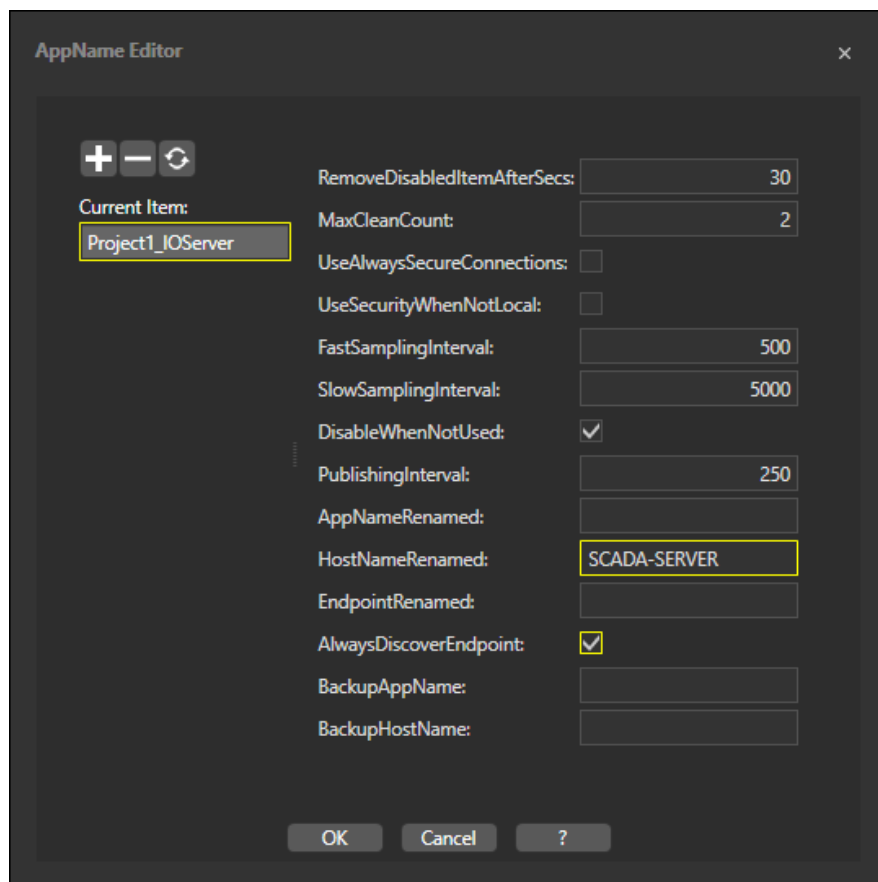
This command is available from the appropriate Project Ribbon as shown below.



When using this command a window will show and which is used to search for the Server when in run mode. If you already know the name of the server you can enter it using the [+] command.

You will need to define the following parameters in this window:

- **Current Item:** use the Application Name of the project in use.
- **HostName Renamed :** define the name of the network of the Server to point to.
- **Always Discover EndPoint:** select this setting to enable Client to automatically detect and replace previous endpoints with the new ones to point to.



This figure shows the window used for assigning the new name of the server for client projects to point to.

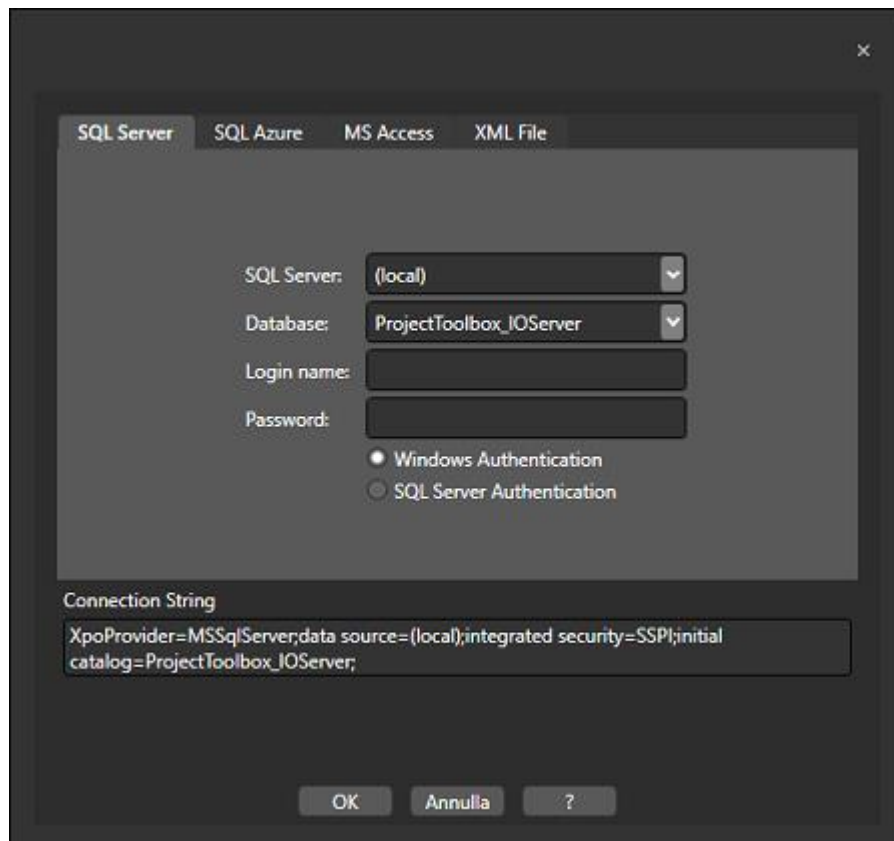
Using Child Projects in distributed Client-Server architectures

The Child and Parent technology is available in Platform.NEXT Client-Server project network architectures. This technology allows you to create projects distributed in a network that automatically connect to a referenced Server.

Please consult the chapter on Platform.NEXT Child Projects for further information.

2.10. DB Connection Settings Window

The DB Connection Settings window is used to define connection strings towards the Data Base selected from those available. This window can be displayed from various project elements that provide the possibility to manage which data source to connect to such as the I/O Data Server or Client viewer objects such as the Data Analysis and Historical Log Viewers.



The provider can be configured using the options shown in the above screenshot.

2.11. PC Time Change

When changing the time on the PC, the Server disconnects the tags and then reconnects them. This behaviour is intentional otherwise if the connection were to continue while clock is put forward/back, any connected security items would go into time-out or expire altogether. Therefore in order to prevent this from happening the connection must be aborted and then re-established.

In normal situations this rarely happens especially when changing legal daylight time. However, it is always best to avoid errors being made by those resources trying to access tags at the same time disconnection occurs. This can be done by manipulating the scripts in such away to prevent this from happening.

Progea Srl
Via D'Annunzio, 295
I-41123 Modena
info@progea.com
Tel +39 059 451060

Progea International SA
via Sottobisio, 28
6828 Balerna (CH)
international@progea.com
Tel +41 91 96 76 610

Progea Deutschland GmbH
Marie-Curie Str., 12
D-78048 VS Villingen
info@progea.de
Tel +49 (0)7721 99838 0

Progea North America Corp.
2380 State Road 44, Suite C
Oshkosh, WI 54904
info@progea.us
Tel. +1 (888) 305-2999