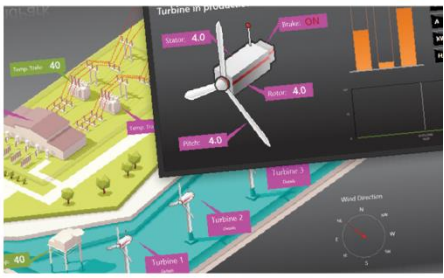
INDUSTRIAL AUTOMATION SOFTWARE

progea

Movicon.next™

Automation Platform.next generation



Movicon NExT

3.2 Tags

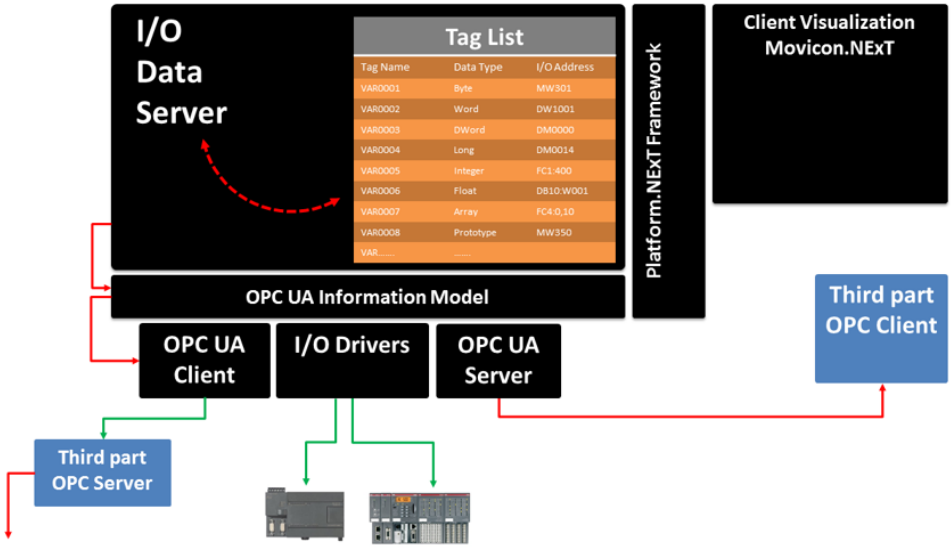
Ver.3.4.268

Table of Contents

1. Tags

1.1. General Concepts

The Platform.NExT I/O Data Server module is the essential data communication and management server for the platform's projects. This Server enables field and third party system communication management by defining and centralizing dynamic information using TAGs to define the project's Address Space variables.



This diagram shows the structure of the system handling data defined using the Tag List while in realtime communication with the field.

The I/O Data Server module enables the platform to communicate with the field devices in real-time, using the included communication drivers or the connection technology to third party OPC UA servers.
All the gathered information is defined in the Server's Address Space, which defines the list of Tags that are visible to all the resources and modules of the entire Automation Platform.NExT platform.

1.2. Variable Definition (Tag)

The project Variables or Tags are defined in the platform's I/O Data Server module and added to the Server's "Address Space (Tag List)" window.

The Variables can be connect to the field through Communication Drivers to exchange information between external devices (such as PLCs, instrumentation, etc.) and all the Platform.NExT modules.

Those variable that are not connected to external devices or systems are considered to be Internal Variables and are therefore not included in the license variable count.

Variables can be grouped into folders at different nested levels. Variables belonging to different folders can also have the same name. One variable can be defined with different types of data, comprising the "Method" and the "Enum". Variables can be retentive and can have initial values.
In order to enter a new variable (Tag) within the project, proceed as follows:

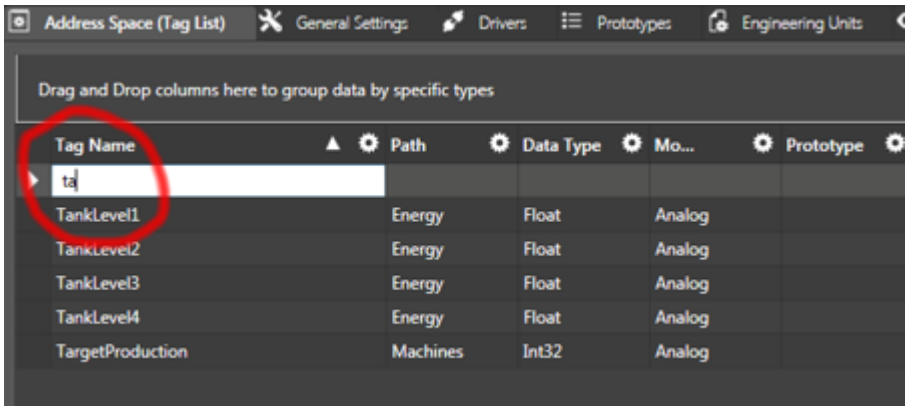
Double click to open the **Address Space** resource from the "IO and Data Server" group in the Project window in order to display the list of project variables in the workspace. By using the "Add New Tag", available in the "UFUAServer-Server Address Space" ribbon, add a variable to the list.
After having added the variable a window will appear to set the variables general settings. This window can then be accessed afterwards by double clicking on the variable inserted.

Tag List Display

The Tag List window can be displayed in two different modes as described below. The display type can be selected using the two selection buttons located on the Tag List window's bottom border.

Hierarchy View (Tree Structure)
The Tags are displayed in the classic tree structure using folders and subfolders.

Flat View (Grid)
The Tags are displayed in a tabular grid and therefore with all the operating commands typically used in grids, such as order bys and filters are used instead of folders.
The image below is an example of a grid window set to filter Tag names:



Note that it is not possible to use the Tag association functionality in the Grid display.

1.3. Tag Data Models

The "Model Type" defines the variable type to be set in compliance with the OPC UA standards. The different types are as follows:

Variable

"Variable" tag types represent objects which are commonly referred to using the term "variable". This objects can be defined a "Data Type" for sizing the Tag. The Data Type of these Tags can be any type from "Boolean" to "Byte" with or without sign, "Word" and "DWord" with and without sign, "Float", "Double" and "String". This type of Tag contains a numeric value or string value which can be read or written according to the settings.

Note: a "**Variable**" Tag type does not support Engineering Unit properties: in this case, use the "**Analog**" Tag type.

Digital, Enumerated

The "Digital" and "Enumerated" tag types are similar: both are "UInt32" type but the "Digital" has only two states while the "Enumerated" has "n". The descriptive string containing the association of various states ("Enum Strings") is specified for both. Only two texts are specified for the "Digital" type while the as many texts as necessary are can be specified for the "Enum" type.
The "|" is used in the string as a separator between on state text and the next.

This type of Tag can be used when needing to display a corresponding string at the rising edge of a numeric value written in the variable.

Analog

The "**Analog**" Tag types are similar to Variable Tag types except that a "Analog" type supports the **Engineering Unit**, and therefore the Server uses and published this information in relation to the Tag value.

Method

The **"Method"** tag allows methods or functions to be executed and which are those made available from the Communication Driver libraries. In this way, methods can be used for executing communications tasks in synchronization. However a "Method" is not a variable that contains a value but one that becomes a function that can be called using the "Call Method" command from the Command List.

ObjectType

The **"Object Type"** tags are structured variables. When a tag is defined as an "Object Type" it will be then necessary to select the Prototype structure to be referred to (for further information see "Prototype Definition").

1.4. Tag_Properties

Each individual Tag inserted on the I/O Data Server's Tag list can be configured by using its **Property Window.** This window opens when double clicking the tag selected.

General Properties

The General properties allow you to configure the Tag's fundamental properties.

Name

The name of the variable is inserted in this field. The variable must have its own unique name when in a group.

Names of tags can only contain alphanumeric characters and the underscore character '_', however the underscore and numbers cannot be used as the initial character.

Description

This field is used for giving the variable a description (field not obligatory).

Model Type

This field is used for selecting which type of variable model is being inserted, whether an analogic value, method or an enumerator, etc.

Data Type

this field is used for selecting the variable's data type. This field is only available when selecting certain 'Model Types' such as the "Analog" Model Type. In this case the variable can be entered with data type such as Boolean, Byte, Int32 or Float, etc.

Enable Tag Statistics

This property is used to enable Tag Statistics. When enabled the following stats values will be published by the Server:

Min: minimum value obtained by the variable
Max: maximum value obtained by the variable
Average: average value obtained by the variable
NumUpdate: number of variable value variations.
TotalTimeOn: total time variable maintained a value other than zero.

Variable value statistics are saved in the variable's retentive file, therefore if you enable the Tag Statistics, you will also need to enable the Retentive option.
The Variable Statistics are only supported for scalar tags and not Structure or Array variable types.
The Variable Statistics are also managed in redundant systems.
The EditDisplay object can be used to access Tag value statistics by displaying the Tag's current value or its value statistics.

Array Dimension

To create a Array tag simply select the number of items it is composed of in the "Array Dimension" field. When left set with the zero value, the tag will remain a simply tag type. Array elements start with "0".
Structure arrays cannot be defined but contrary to this array structures can.

To insert an initial value in a array tag you must insert the values of each item separated by the pipe character (|) and enclosed between brackets like this: {1|10|5.5|...}

Enum Strings

This property is available only when "Digital" or "Enumerated" have been selected as "Model Type".
The texts corresponding to the enumerator or boolean values are entered in this field. The "|" character must be used to separate texts of one state and the next. The

string can be composed using the edit window which can be accessed from this property. This window also allows you to associate strings from the project String Table with the 'Texts' Resource.

Prototype

This field is used for selecting the prototype to be used for defining the variable with. This field is only available when the variable's "Model Type" has been selected as "Object Type". The Prototype to be selected must be previously defined in the appropriated "Prototypes" resource.

Enable Redundancy

When enabled this option excludes the tag from the redundancy management (when provided by project). In this way the Tag will not be synchronized with the other Server on the redundancy list.

When using a Structure variable, each of its members can be enabled or disabled for synchronization (the IsRedundancyEnables property is not available at Structure Variable level but only for each of its members).

Execution Properties

The Tag Execution properties are important because they are used to configure the tag's role within the project.

I/O Physical Address

The physical address of the device's memory area to connect to (e.g. a PLC). Before entering this address you must define a least one communication driver in the project beforehand: for further information on how to insert the link please refer to the section on "Connect Variable to Communications Drivers".
When this property is not set, the tag will be considered as an internal tag and therefore not counted by the license manager.
By using the command on the right a configuration window will appear to set the physical address towards the field. This is done by selecting one of the driver previously inserted in the project and specifying the device's memory area to communicate with.

For specific details on physical addresses for tags, please consult the communication driver's protocol guide.

As from the Movicon.NExT 3.1 version it will be possible to associated different drivers to one single variable. For further information please see the chapter on "Tag I/O Physical Address".

Retentive

This check-box is used for defining the variable as retentive, meaning that its last value must be saved on file and maintained for subsequent project startups.

Initial Value

This field is used for specifying a value with which the variable will be initialized at every project startup (field not obligatory). In cases where the Tag is an array type, the values of each element are to be inserted and separated with the "pipe" (|) character and enclosed between brackets: (eg: {1|10|5.5|...})

Engineering Unit

This property is only available when "Analog" has been selected as "Model Type".

This field is used for selecting an Engineering Unit which was previously inserted in the 'relevant I/O Data Server window in order to associate it to the variable.
The Engineering Unit will be applied to the Tag value calculating the resulting scale value from the parameters set with the Min. and Max raw value and the Min. and Max. value of the set engineering conversion rate.

Historian Settings

This property is used for selecting a Historian type to associate to the tag for recording data. Historians must be previously entered and configured in order to do this. Once a historian has been selected, the tag will be recorded in the historical data archives accordingly.
For further information please see "Historicals".

User Access

The Tag's 'User Access' property allows you to configure its security properties.

A tag's access rights are applied the moment the tag is subscribed to the Server. This means, for example, to enable the opening a Screen that connects to a specific Tag, the user must have already logged in with the appropriate credentials needed to access that Tag. Otherwise the Tag will be subscribed but not accessible. At that point it will not be sufficient to log in again with the appropriate credentials because the Tag will have already been subscribed to the Server. Therefore you will need to close the screen and wait for the time needed to remove the Tag from the Server and then log in correctly in order to re-open the Screen and subscribe the Tag again.

The management to read/write access Tags associated to objects can be assigned to the Server project. To activate this functionality you will need to enable the "User Management" -> Access Level from Tag' property in the desired objects.

User Read Access Mask

This is used to select up to 32 access areas from a selection box to give or deny the logged in user access to read data. The user in question must however have a 'Access Level' that is equal to or higher than the one requested by the Tag.

User Write Access Mask

This is used to select up to 32 access areas from a selection box to give or deny the logged in user access to write data. The user in question must however have a 'Access Level' that is equal to or higher than the one requested by the Tag.

User Access Level

This is used to define a number corresponding to the hierarchical access level, to allow those users who Log On with an access value equal to or higher than this number to have access to and operativity with the Tag value. The access modes available to the authorized user are those defined by the "Access Mode" property.
Once the first User authentication phase has concluded successfully by logging in with the appropriate Access Level, the user can still be prohibited from performing certain operations on the Tag by using the Access Area in read and write.

Access Mode

This setting is used to define the access mode to use by the Server to publish the Tags. When authorizing a user access to the Tag, it is also possible to establish which operations they will be allowed to do using this property.

The possibilities are:

- None: User will not be allowed to write or read data independently from their access rights used to connect to the Tag.
- Read: User will not be allowed to write but only read data independently from their access rights used to connect to the Tag.
- Write: User will not be allowed to read ut only write data independently from their access rights used to connect to the Tag.
- Read/Write: User will be allowed to read and write data according to their access rights used to connect to the Tag.

Audit

Enable Audit Trace

This enables the Audit function. This option is also available for structure prototype members but not for methods. The Audit traces each tag change made by the user while operating with the supervisor: display, check box and selectors. The trace and relative comment (if Enter Comment On Audit) has been enabled), will be recorded in the table already in use by the Historian. The comment in the historian will be saved in the new 'Reason' field.

Enter Comment On Audit

Used to enable the comment on the variables' Audit. When activating this property, the 'Force Password On Audit' and 'Access Level Required to Confirm' properties will automatically show.

Force Password On Audit

Even though the user has already logged-on, this forces the user to re-enter their password again to confirm the comment they have entered in the appropriate audit window which appears on the client.

Access Level Required to Confirm

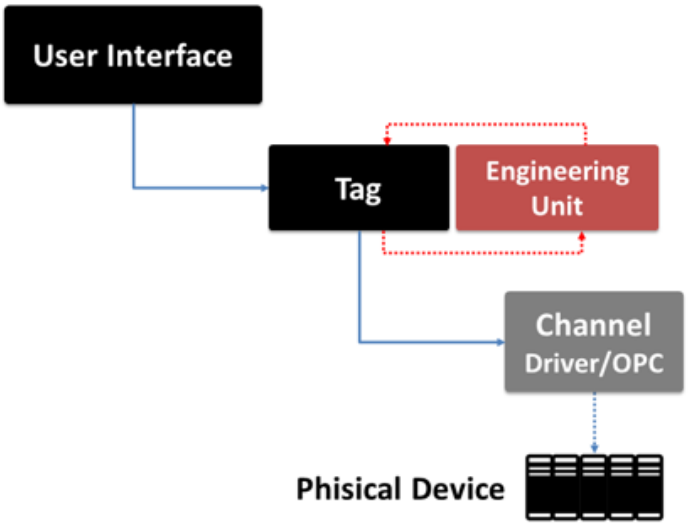
Minimum level required for double checking Audit. Therefore, if the user who entered the comment has lower level access, a double check will be made by requiring another user with this level to confirm.

Max Audit Age

The maximum age for Audit Trace information storage. Rows will automatically be deleted once exceeding this age.

1.5. Engineering Units

The 'I/O Data Server's Engineering Units window enables to insert and configure models for converting data from an input value to a calculated and scaled output value.



The Engineering Units are also important because they enable the automatic configuration of the user interface's multi object scale values. For instance, when associating Tag that is already associated to Gauge object (viewer object with dial indicator needles), this tag will adapt the Min. and Max. scale values according to the those set in the Engineering Unit's EU Low Range and EU Max Range fields.

Name
This is used for defining the name to be assigned to the Engineering Unit.

Unit Name
This is used for defining a text value (or an abbreviation) to assign to the engineering unit.

EU Range Low / EU Low Range
This is used for defining the minimum output value, calculated in ratio of the raw input value.

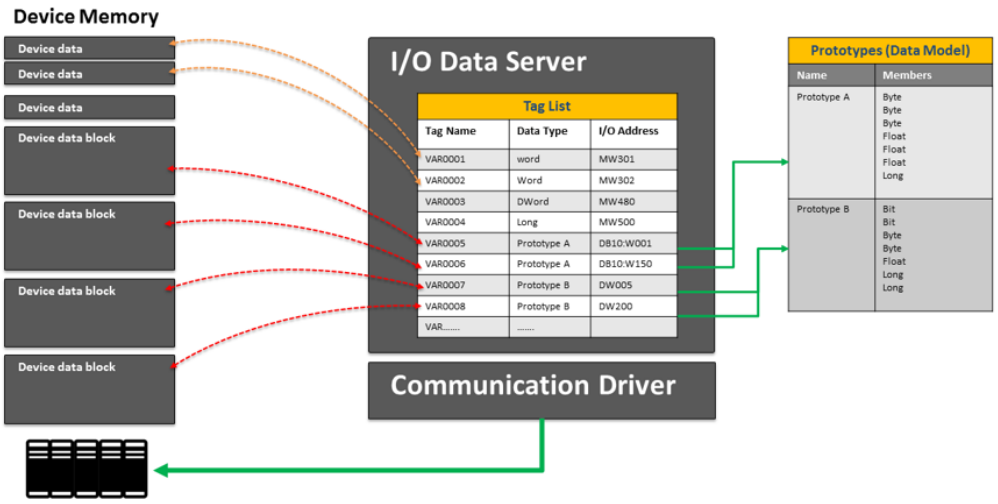
EU Range High/ EU Hight Range
Defines the maximum output value, calculated in ratio of the raw input value.

Instrument Range Low /Low Range Instrument
Defines the minimum raw input value with which the engineering output value will be calculated.

Instrument Range High / high Range Instrument
Defines the minimum raw input value with which the engineering output value will be calculated.

1.6. Prototype Definition

The I/O Data **Server Prototype** window enables you to insert and configure the data models as desired. These data models are also defined as Tag "structures" or "prototypes".
The prototypes are data structures used for defining complex tag types such as those using data model types composed of a series of data types and quantity defined by the user and called "**members**".



The structure prototypes are inserted in the "Prototype List" within the I/O Data Server's window and unlike tags/variables cannot be inserted into folders.

Folders and members can both be added and nested at different levels within the prototype data model. This is also handy for organizing the prototype members better. The structure defined variable will then be published by the Server towards the Client.
The prototype members can in turn be defined with Prototypes, with the aim to obtain structures within structures. However structure members can not be defined with prototypes that would result recursive. For example, a prototype member cannot be defined and then selected with a prototype that contains a member defined from the same initial prototype (i.e. Prototype1:Member1 where Member1 is defined as Prototype2 and Prototype2:Member2 where Member2 is defined as Prototype1).
A member can also be defined as a "**Method**" type. For example, by using the "Demo" Communication Driver and defining a member as the Driver's Method, this method will then be applied only to those variables belonging to that prototype and not to all the other Driver variables. In this way, a prototype can be created with a simulation variable and two Driver Start and Stop simulation methods and then afterwards each variable defined with this prototype will be able to manage the stop and start simulation commands independently.

Prototype members have a "**MemberOrderID**" property which is used for defining the order with which the members are to be exchanged by the communication driver. The order is established using IDs which are inserted using an incremental number sequence starting with 0 for default. By numbering all the prototype members starting from 0, as well as those nested in folders, will enable you to establish the desired sequence with which members will be exchanged with the Communication Driver. Two commands are available from the Ribbon for increasing or decreasing member ID numbers.

To enter a new Prototype in the project, please proceed as follows:

1. Double click to open the **Prototype** Resource from within the Project window to display the list of the project's Prototypes within the workspace area. Add a Prototype to the list by using the "Add New Prototype" command available in the Ribbon "I/O Data Server Address Space" Ribbon.

2. After having executed the command for inserting a new Prototype, a window will show for entering the prototype's name. This window can also be accessed afterwards by double clicking on the Prototype when inserted.

3. To add new variable members to the Prototype, select the Prototype and then activate the "Add New Tag" command, which is available from the "I/O Data Server-Address Space" Ribbon. At this pint a window will appear, which is very much like the one used for inserting variables, for defining the member's properties. Prototype member properties are infact a subset of variables.

Prototype Member Properties

Each Member of a Prototype can be configured through the Property Window in the same way as simple variables. The only difference is that a few properties such as the 'I/O Physical Address' and 'Retentive' properties are not available for Prototype members but which can be applied to Structure Variables defined with the Prototype. The Structure Prototype can be used to create several Structure Variables once it has been defined and each one of its Members have been configured. It is also possible to intervene to customize the Structure Variable' members to make it different from the other Structure Variables even though defined for the same Prototype. A Structure Variable can be exploded to view its structure and access the properties of each individual member. All the Members have a “UseShared” property which is left checked so that they inherit the configuration defined in the Prototype. When this property is unchecked, it will be possible to edit some of the Member's properties which will then have priority over the Prototype's settings. The following modifications can be done when unchecking a Structure Variable's Member's “UseShared” property:

- Insert the I/O Physical Address in each Member
- Insert a Description
- Insert an initial value
- Assign an Engineering Unit
- Assign a Historical
- Define User Access settings: User Level. Access Area in Read and Write, Access mode
- Assign one or more Alarms
- Assign one or more Views
- Assign script code to the Member

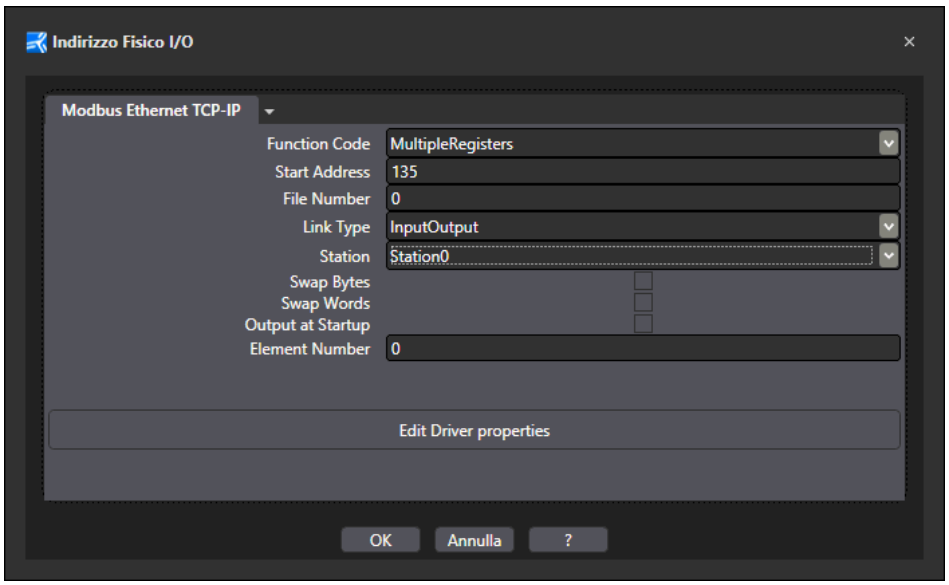
A project that uses the properties of Members defined at Structure Variable level (by disabling the “UseShared” property) will not be compatible if opened with the 3.0 or previous versions of Movicon.NExT.

1.7. I/O Addressing

Each Tag can be associated with a physical I/O address towards Communication Drivers by using their properties. Before assigning the physical address of the device to which the tag must be associated, you will need to insert and configure at least one communication driver to enable this.

By using the Tag's "I/O physical address" you will be able access the configuration window used for assigning the address.

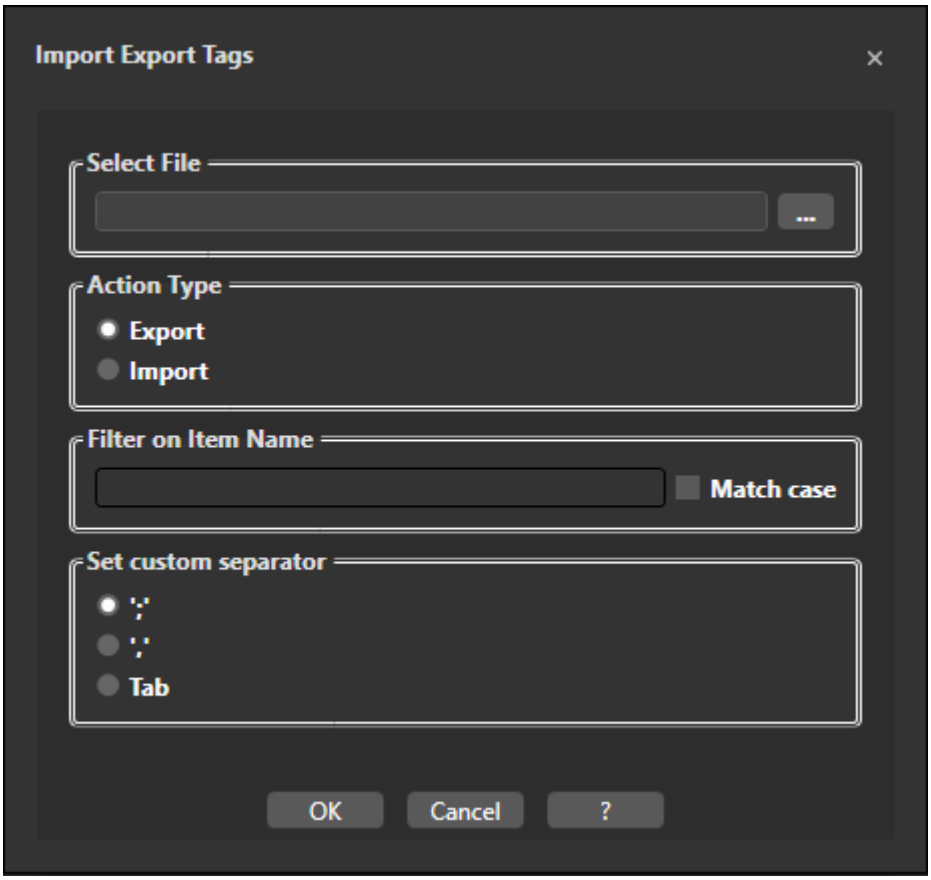
The I/O physical address configuration window refers to the previously inserted communication driver or drivers. Based on the driver type and relating communication protocol, you will be able to define the physical address in the memory of the device to which the Tag will be associated. This same window can be used for defining the connection mode as well.



Note: Tags that have not been associated with a physical I/O address will not be considered as internal tags and therefore they will not be counted by the License manager.

1.8. Import/Export

The I/O Data Server permits use of commands to Import or Export data in .CSV text files that are usually managed by external editors such as Microsoft Excel.
Data can be referred to the Tag List or Engineering Units.
The Import and Export commands are available from one of the predisposed ribbons.



Import/Export Tags

Tags can be exported or imported to your project by using the appropriate commands from the I/O Data Server > Address Space Ribbon.
The commands provide different method level type options to define the tags to be exported or exported:

Select File
Use this setting to select the destination file to import or export data.

Action Type
Use this to select the tag import or export.

Filter on Item name
Use this setting to filter data to import or export. This command can be set to upper or lowercase case sensitive.

Set Custom Separator
Select the separator type to use between one variable and the next.

Import/Export Engineering Units

Engineering Units can be exported or imported in the same way used for exporting and importing Tags in your project.
The operations for executing the command are practically the same except for the Base, Medium and Advanced level methods of the Tag properties.

1.9. System Tags and Local Tags

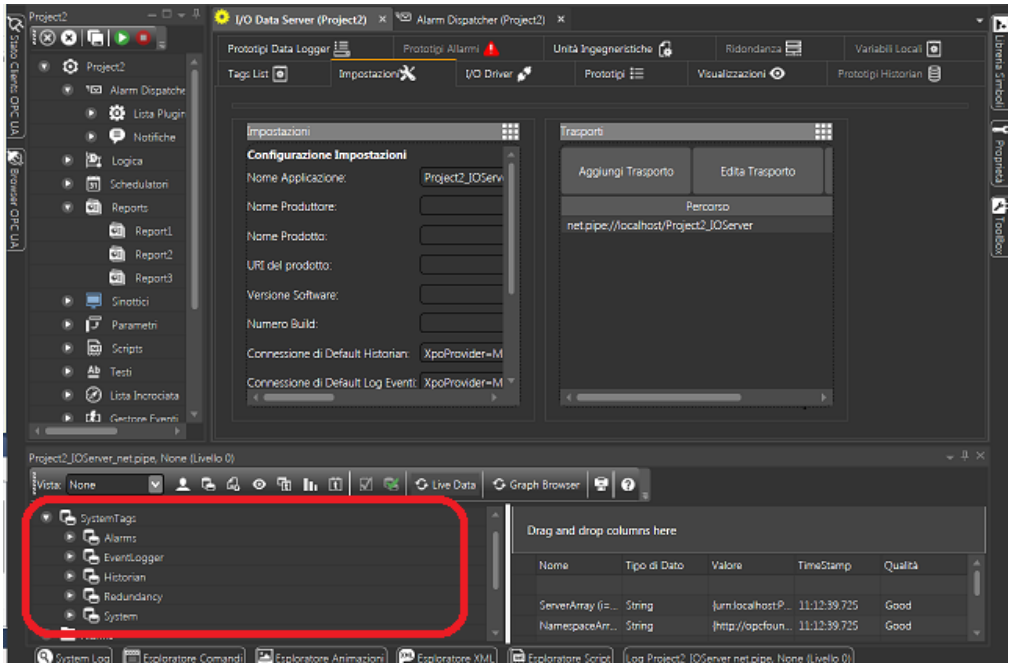
The Platform.NexT Data Server defines a list of tags which are available for entire project use on both client and server. However, there are Tags that are only available for use on the Client side. These tags are very handy to use in certain occasions when not using the Data Server.
The Tags that can be used on only when selected on the Client side are:

- **System Tags**
- **Local Tags**

Both types are described in the chapter on Visualization Client.

System Tags

Tags exposed by the Server report information relating to the functioning of the project's server side. Contextually, when the project server is started up, a client integrated in the NExt Editor is opened as the "OPC UA Browser" window. This window appears at the bottom of the screen and shows the variables exposed by the Server on the left hand side.



The SystemTags group provides information relating to the system divided in categories.

Tag Name	Data Type	Group	Description
SoundState	Boolean	Alarms	Tag for reading or setting the alarm sound state to activated or deactivated
NumActiveOff	Int64	Alarms	

			Number of alarm active with OFF status.
NumActiveOn	Int64	Alarms	Number of alarm active with ON status.
NumActiveOnOff	Int64	Alarms	Number of alarm active with ON and OFF status.
NumEnabled	Int64	Alarms	Number of enabled alarms.
NumNotAck	Int64	Alarms	Number of acknowledged alarms.
NumShelved	Int64	Alarms	Number of shelved alarms.
DeleteEntriesPending	Integer	Eventlogger	Number of pending delete entries to be performed.
DeleteEntriesRunning	Integer	Eventlogger	Number of Delete entries being run.
DischargingEntries	Boolean	Eventlogger	“True” when discharging data entries due to reaching internal memory cache limit (Max Pending Entries).
FailsEntriesPending	Integer	Eventlogger	Number of failed entries pending database recording.
FailsEntriesRunning	Integer	Eventlogger	Number of previously failed entries being recorded on database.
FlushEntriesPending	Integer	Eventlogger	Number of entries pending flush to local XML file after the maximum number of attempts failed to record on Database.
FlushEntriesRunning	Integer	Eventlogger	Number of entries being flushed to the local XML file to then be retrieved by the system when database re-entered from error state.
RecordEntriesPending	Integer	Eventlogger	Number of entries pending database recording.
RecordEntriesRunning	Integer	Eventlogger	Number of entries being recorded on database.

DeleteEntriesPending	Integer	Historian	Number of delete operations pending execution.
DeleteEntriesRunning	Integer	Historian	Number of delete operations being run.
DischargingEntries	Boolean	Historian	“True” when discarding data entries due to reaching the internal memory cache limit (Max Pending Entries).
FailsEntriesPending	Integer	Historian	Number of entries waiting to be recorded on database and which failed the previous attempt to do so.
FailsEntriesRunning	Integer	Historian	Number of entries being recorded on database and which failed the precious attempt to do so.
FlushEntriesPending	Integer	Historian	Number of entries waiting to be flushed to the local XML file due to exceeding the maximum number of attempts after the last attempt to be recorded on database failed.
FlushEntriesRunning	Integer	Historian	Number of entries being flushed to the local XML file to then be retrieved by the system when database returns back from error state.
RecordEntriesPending	Integer	Historian	Number of entries waiting to be recorded on database.
RecordEntriesRunning	Integer	Historian	Number entries being recorded on database.
Redundancy ActiveServerState	Boolean	Redundancy	Indicates whether the local server ia active or inactive. Each server indicates its on status.
DynamicTags	Integer	System	Returns the number of TAGs counted for license purposes.  Each 'simple' tag from 1 to 64 bit is counted as a licensed TAG while each

			member of an array or structure tag counts as 1 TAG.
Redundancy ActiveServerHostName	String	Redundancy	This tag is redundant in all the Servers that have been started up and reports the name of the Active Server which is the one in control.
Redundancy ArrayAliveServerHostName	String	Redundancy	This tag is redundant in all Alive Servers and reports the list of names of those Servers which are currently alive.
Redundancy SwitchActiveServer	Method	Redundancy	This method, which can be executed both in the Active Server and inactive Server, switches the Active Server when invoked. This method is enabled when there are two Servers on the list only and if the 'Keep Server Active' redundancy property has been enabled.
NumActiveOn	Int64	Messages	Number of alarm messages with ON Status.
NumEnabled	Int64	Messages	Number of enabled alarm messages.
NumShelved	Int64	Messages	Number of shelved alarm messages.

Local Tags

The local project tags are tags that the programmer can create within the project and used in screen controls or script. The Local Tags are not deployed in the Server's address space and are only visible at Client level.

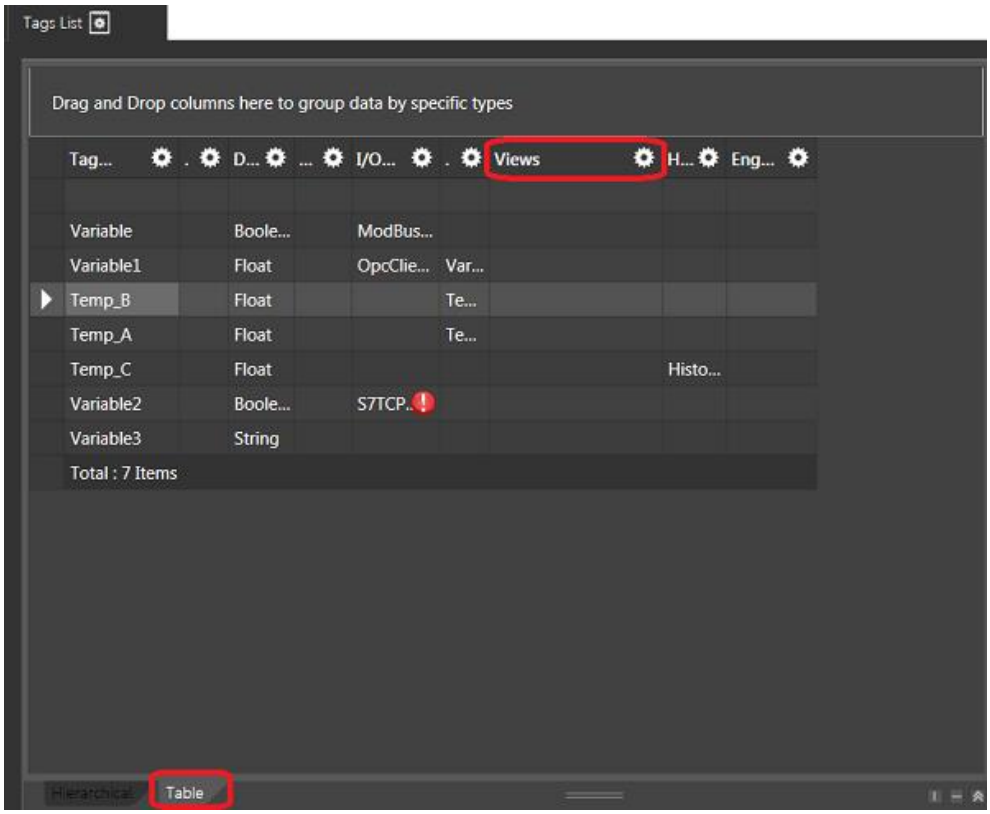
For further information please see topic on: "Local Client Tags".

1.10. Views

The I/O Data Server's View window is used for inserting and configuring tag group "Views" when selecting tags from the Address Space using the tag browser window.
All tags in the platform are selected with the "browser" window which has been set for default to browse all defined variables.

It is also possible to create one or more "Views" that can then be associated to each individual Tag or Tag Group. This will enable you to display the desired "View" from those inserted in the server by using the "browser" window.

A filter is provided in the 'Table' display type which can be used to display all the variables that have been associated to a specific 'view'. This mode type is accessible from the "address space" - "Tag List".



INDUSTRIAL AUTOMATION SOFTWARE

progea

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