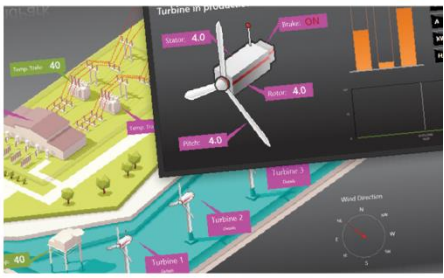
INDUSTRIAL AUTOMATION SOFTWARE
progea

Movicon.next™
Automation Platform.next generation



Movicon NExT

7.0 Recipes

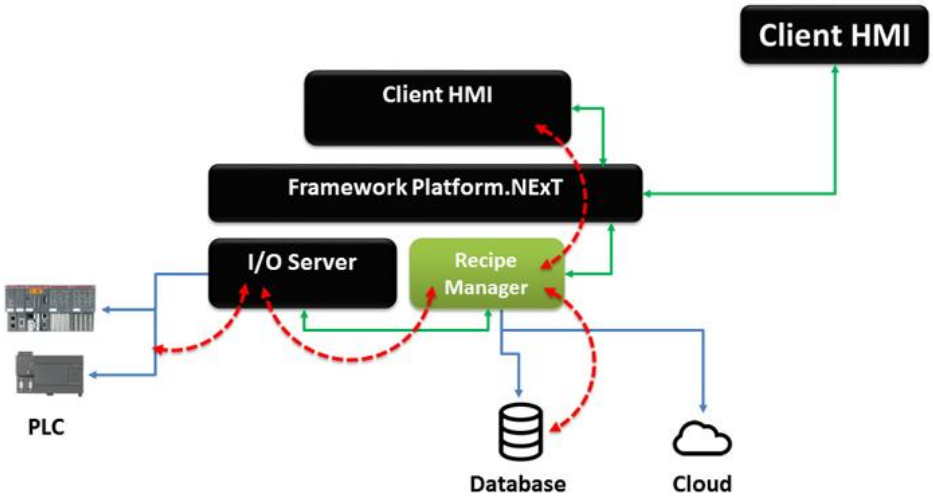
Ver.3.4.268

Table of Contents

1. Recipes

1.1. Recipe Manager

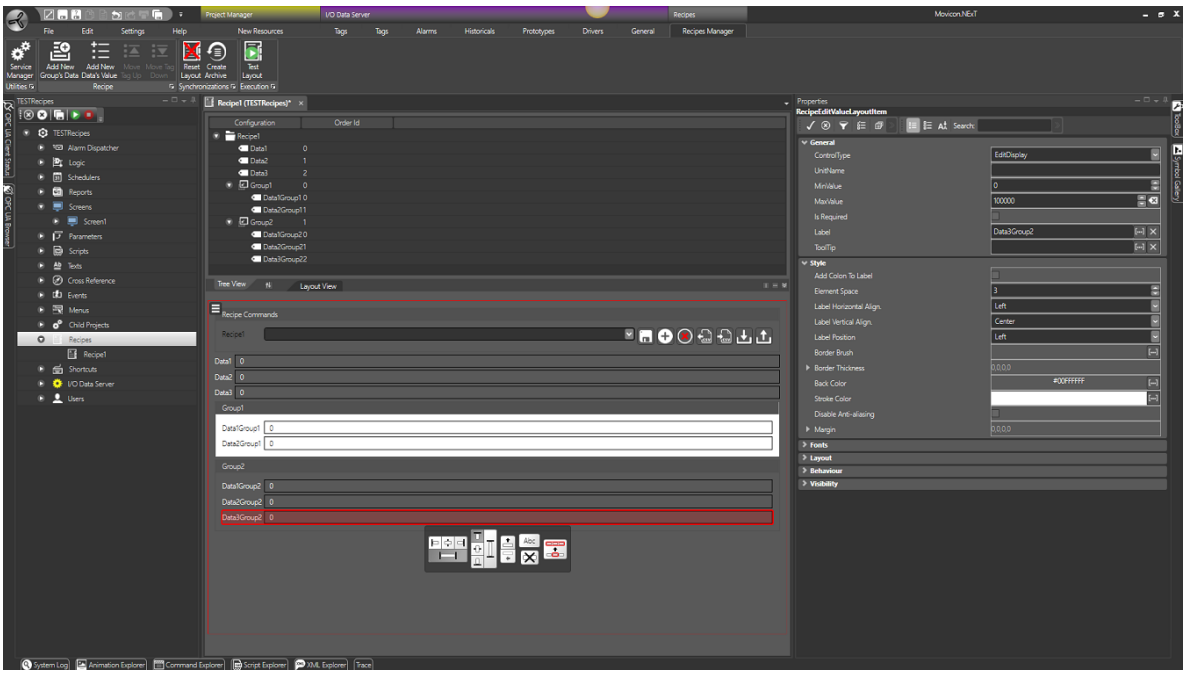
Often manufacturing plants need to record parameters and set points of different product types on file and store in an archive system so that they can then be modified when and as needed to activate production runs of different products. This is what is commonly known as production recipe management.
The Platform.NExT Recipe management allows you to define system or process data groups,values and set points that must be recorded and managed in database archives. They can then be selected by name (recipe index or title) when needed and activated by transferring them to the PLC or field devices.



This diagram shows a function block scheme of the Platform.NexT Recipe Manage. This module manages connections towards the database and display objects along with direct connectivity with field devices.

For instance, production processes whose end product is the result of a mixture composed of different components, each with a specific measure, necessitate a solution where such components and measures can be established in production recipes. Once the recipe has been established with the right ingredients needed to obtain the desired end product, it can then be activated and processed accordingly.
This concept can be applied whenever it is necessary to enter, save, print and activate data whenever required.

Platform.NExT completely manages the use of the Recipes in automatic using a functional model which comes included with the system and is called the 'Recipe Manager'. This module can be configured by means of using the 'Recipe' resource for each recipe in the project tree structure.

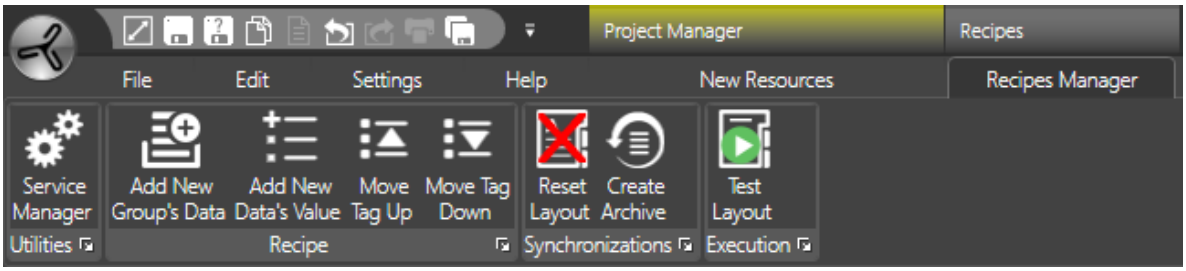


When you open the **"Recipe"** resource from the project's tree structure window, as shown above, the editor will display and by means of which new recipes can be created or existing ones can be modified by using structure or layout tools.

- The first tool is indispensable for creating the recipe structures such as defining the recipe's name, groups and value field.
- The second tool is used afterwards to defined the recipe's "**Layout**" eg. how the value fields are to be represented in the graphical interface. This field is Optional as it is not indispensable to use this graphical interface to manage our recipes.

After having added the various data to our recipe, we can then go ahead and associated it to a specific comando to manage it or configure it directly using the recipe's Dati Tag properties.

The 'Recipes' Ribbon Tab and other "Recipe Manager" Tabs, that activate when the recipe opens in the workspace, contain the following commands (see figure below):



Service Manager

This Service Manager command is used to activated the recipe process as service.

Add New Group's Data

This command is used to insert a new group in the recipe that is open.

Add New Data's Value

This command is used to insert a new variable in the open recipe.

Move Tag Up

This command is used to move a variable up one place in the recipe tree structure.

Move Tag Down

This command is used to move a variable down one place in the recipe tree structure.

Reset Layout

This command is used to resets the recipe's layout.

Create Archive

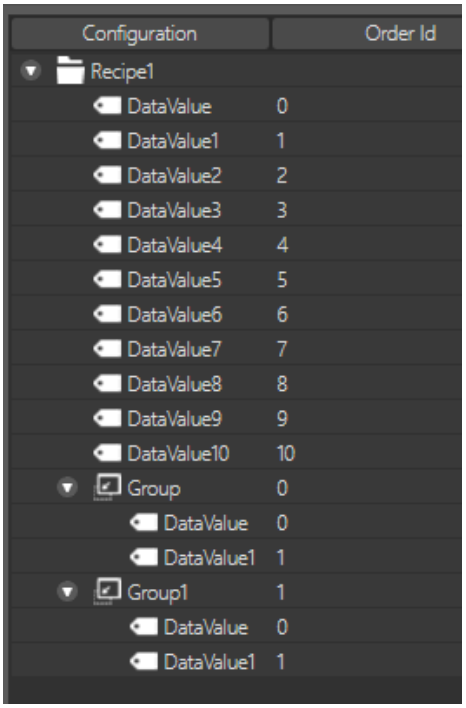
This ocmmand created the recipe's data structure in the associated database by cancelling the existing one, if any, along with its data contents.

Test Layout

This command displays the system window with the recipe's graphical interface as it will appear to the user in runtime using the system Pop Up window.

1.2. Recipe Structure

A Platform.NExT Recipe is composed of one or several data values that can be divided into different Groups while maintaining reference to the Recipe Index.
The Recipe data values are saved in the database and connect to the field devices by means of the I/O Data Server's Communication Drivers. This enables recipe values to be loaded from the database and transferred to the PLC or to connected field devices or vice-versa whereby values can be read from the device and saved in the database with the set Recipe name (index).



The recipe structure is represented by a tree structure within its own Editor as shown above. The recipe is comprise of various as described below:

Groups in the Recipe Structure

Using groups in the recipe structure enables users to create different groups to divide and organize lots of recipe fields as well as define specific physical start addresses for each individual group created.

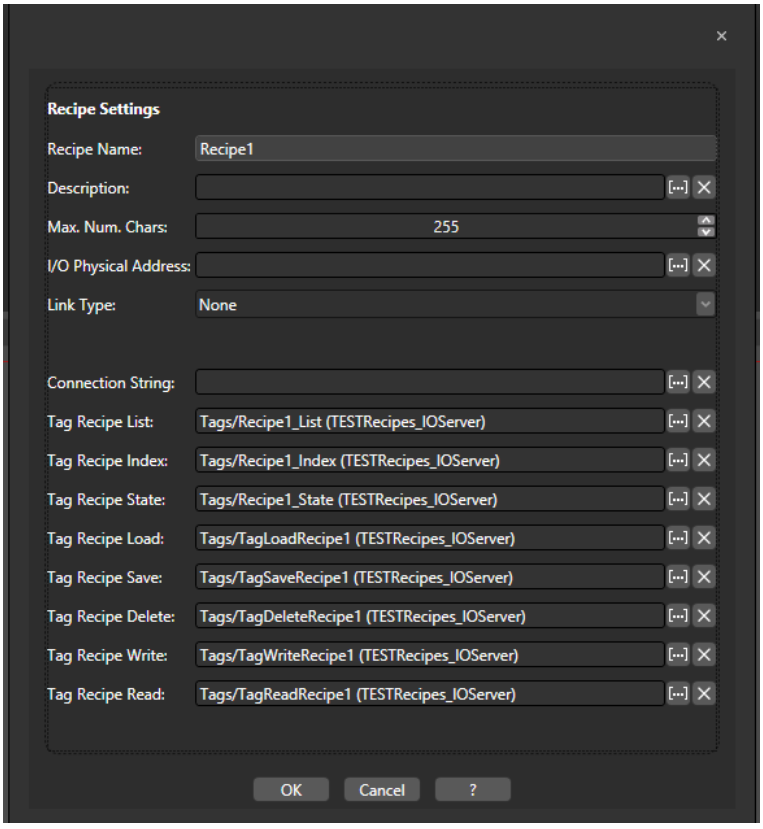
Data Values in the Recipe Structure

The recipe fields are represented by data values which are typical set points that are used to compose the recipe values. Each data value is recorded in the recipe based on their index.

The Recipe's structure must be defined before configuring the Recipe's layout. The Recipe Layout is the recipe's graphical interface which is configured using the recipe's "Layout View" Editor.
The data layout, as describe in the relating topic, will determine how you wish the Recipe to be displayed along with the its relating operativity using the appropriate commands provided.

1.3. Recipe Properties

The window used for setting the Recipe properties is opened with a double click or pressing the F4 key after having selected name of the Recipe opened in the workspace (see figure below). The same parameters are also available in the Properties Window once the Recipe has been selected.



Connection Settings

The Connection properties determine some modes relating to the recipe connections and regrouping of the following settings:

Session Name

This property is used to indicate a private session within which the recipe will be handled. If no session is indicated, the default session belonging to the whole project will be used.

Publishing Interval

Parameter can only be used when the recipe's "Session Name" has been set. This value is calculated on the "Remove Item Delay" property to indicate after how many seconds the recipe is to be unloaded and put its existing variables out of use.
If the recipe is not set a with session, the one set for the project will be used instead and as a consequence so will the value set in the project's "Close Screen Delay" property be used.

Disable When Not Used

This parameter is only handled when the recipe's "Session Name" has been set. It is used to indicate after how many seconds the not-in-use items must be disabled.

If the recipe is not set a with session, the one set for the project will be used instead and as a consequence so will the value set in the project's "Remove Item Delay" property.

Remove Item Delay

Sets the delay time in seconds with which the inactive ITEM's OPC UA subscriptions will be removed.

Remove Item Num

Sets the number of inactive OPC Items to be removed at a time.

Only Secure Connections

When enabled only those server connections declared secure by the appropriate certificates will be used.

Fast Sampling Interval

This parameter can only be used when the recipe's "Session Name" property has been set. This property is used to set at what frequency rate an in-use and existing variable is to be updated. This parameter is passed to the Server and then the Driver when the recipe is loaded and the variables go into use.
If the recipe is not set a with session, the one set for the project will be used instead and as a consequence so will the value set in the project's "Fast Sampling Interval" property be used.

Slow Sampling Interval

This parameter is only handled when the recipe's "Session Name" has been set. This is used to set at what frequency rate an existing variable, that is going out of use, is to be updated. This parameter is passed to the Server and any eventual Driver, when the recipe is loaded.
If the recipe is not set a with session, the one set for the project will be used instead and as a consequence so will the value set in the project's "Slow Sampling Interval" property be used.

Database Settings

Connection String
This field is used to specify the connection string to be used for recording Recipe data in the database. When this field is left empty, Movicon will use the project's default connection for the historicals defined in the project's "Historian Default Connection" property.

Max. VarChar Precision
This field is used to specify the maximum number of characters can be used in a recipe string. This parameter effects both on the recipe name and the set string values.

General

Recipe Name
Indicates the name of the recipe. This field is read only. The recipe's name can be changed directly in the Project Explorer Window.

Description
Indicates the Recipe's description. Text can be entered as pleased and will be shown in the Recipe's "Layout View" window in the “ToolTip” of the "Recipe Index" control.

I/O Physical Address Start value
This field is used to specify the start address of the area where data is to be exchanged with the connected device. In order for this to function you will need to configure a Communication Driver beforehand. The connection settings are the same ones used in the project Tags for connecting them to the Driver.

Link Type
Indicates the data exchange task type between the Recipe and the device. The Recipe can actually be read or written to/from the device, or read only or write only. In this case the link type is determined by the task type defined in the "Start Address" parameter. This field however is read only and gets updated automatically by Movicon based on the settings defined in the "Start Address". When no settings have been defined in the "Start Address" property, this field will be set to "None".

Data Tags

Recipe List Tag
This field is used to select a string type Tag in which the list of recipes saved in the DataBase will be shown in runtime. The “pipe” (|) character is used to separate each recipe name in the string.

Recipe Index Tag
This field is used to select a string type Tag in which to set the name of the recipe to be loaded from the Database in runtime. The name of the recipe defined in this Tag will also be the one used by the "Recipe Commands" for read, write value operations.

Recipe State Tag

A double word type Tag can be selected in this field in which the Recipe status will be shown in runtime. The double word is managed in bits and each bit has a significant meaning as follows:

Bit 0: Recipe Manager initialization
Bit 1: Stops Recipe Manager
Bit 2: Loading values from DB now in progress
Bit 3: Saves Recipe Values in the DB being used
Bit 4: Removes Recipe from the DB being used
Bit 5: Reads Values from the Device being used
Bit 6: Writes Values in the Device being used
Bit 7: Importing Recipe values from File now in progress
Bit 8: Exporting Recipe values from File now in progress
Bit 12: Error while updating Recipe Tag List
Bit 13: Error while updating State Tag
Bit 14: Error while updating Recipe Tag values
Bit 15: Error while Saving Recipe in DB
Bit 16: Error while removing Recipe from DB
Bit 17: Error while reading values from Device
Bit 18: Error while values in Device
Bit 19: Error while importing values from File
Bit 20: Error while exporting values from File
Bit 21: Connection TimeOut of Recipe Command Tags
Bit 22: Connection TimeOut of Recipe Tag Values
Bit 24: Not used
Bit 25: Saves Recipe values in the DB running correctly
Bit 26: Removes Recipe for the DB running correctly
Bit 27: Reads Values from the Device running correctly
Bit 28: Writes Values in the Device running correctly
Bit 29: Recipe Values imported correctly
Bit 30: Recipe Values exported correctly
Bit 31: Recipe Values read from DB executed correctly

Load Recipe Tag

This field is used to select a Tag to load the values of the recipe currently selected in the Recipe Index Tag (currently active recipe) from the DB the moment upon which a value changes from zero (then automatically zeroed by the system after operation has completed).

Save Recipe Tag

This field is used to select a Tag which will save the recipe currently selected in the Recipe Index Tag with its current associated variable values.

This is performed that moment upon which this Tag assumes a value different to zero (then automatically zeroed by the system after operation has completed).

Delete Recipe Tag

This field is used to select a Tag which will delete the recipe currently selected in the Recipe Index Tag with its current associated variable values.

This is performed that moment upon which this Tag assumes a value different to zero (then automatically zeroed by the system after operation has completed).

Write Recipe Tag

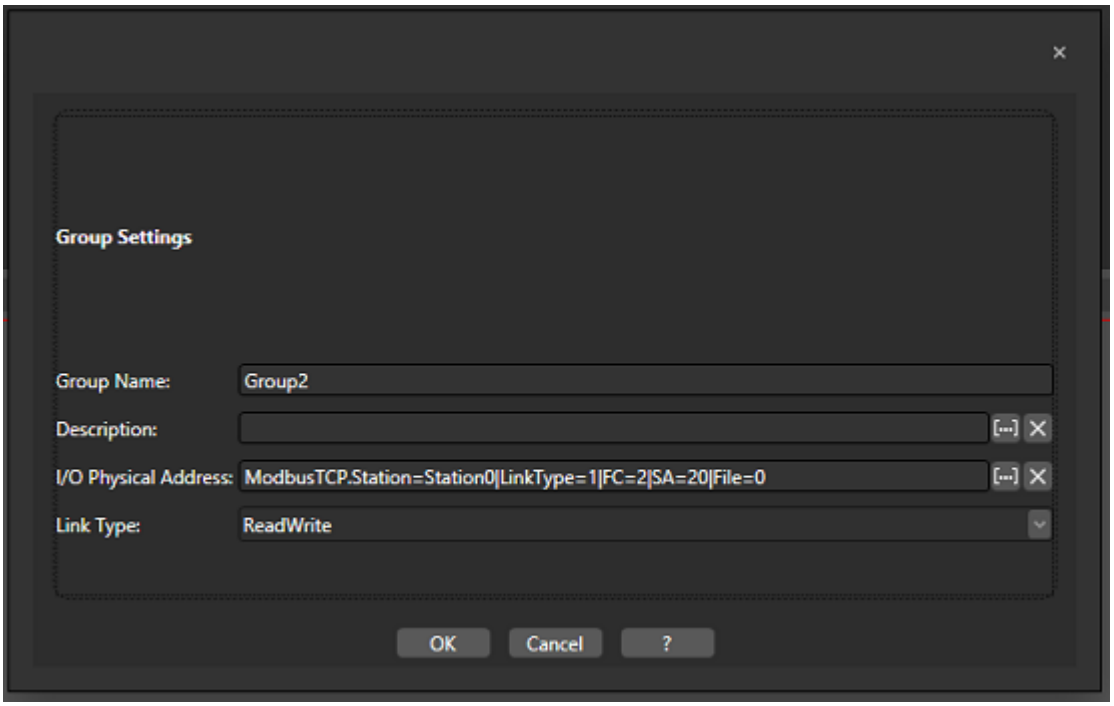
This field is used to select a Tag to send the values of the recipe currently selected in the Recipe Index Tag to an external device (eg. PLC) This is performed the moment upon which this Tag assumes a value different to zero (then automatically zeroed by the system after operation has completed).

Read Recipe Tag

This field is used to select a Tag to read the values from an external device (eg. PLC) which will then be assigned to the recipe currently selected in the Recipe Index Tag. This is performed that moment upon which this Tag assumes a value different to zero (then automatically zeroed by the system after operation has completed).

Group Property Settings

In addition to organizing data into coherent groups, these Recipe Group properties are also used to specify a "Start Address" which will refer to that group only. In this way different date exchange areas can be defined toward the device for each Group. The window used for setting properties of groups belonging to a Recipe opens with a double click or the F4 key after having selected the name of the group belonging to the recipe in the workspace (see figure below). The same parameters are also available in the Properties window once the Recipe has been selected.



Group Name

Indicates the name of the Group

Description

Indicates the Group's description. Text can be entered as pleased and will be shown in the Recipe's "Layout View" window when the Group is inserted.

I/O Physical Address

This field is used to specify the start address of the area where data is to be exchanged with the connected device. In order for this to function you will need to configure a Communication Driver beforehand. The connection settings are the same ones used in

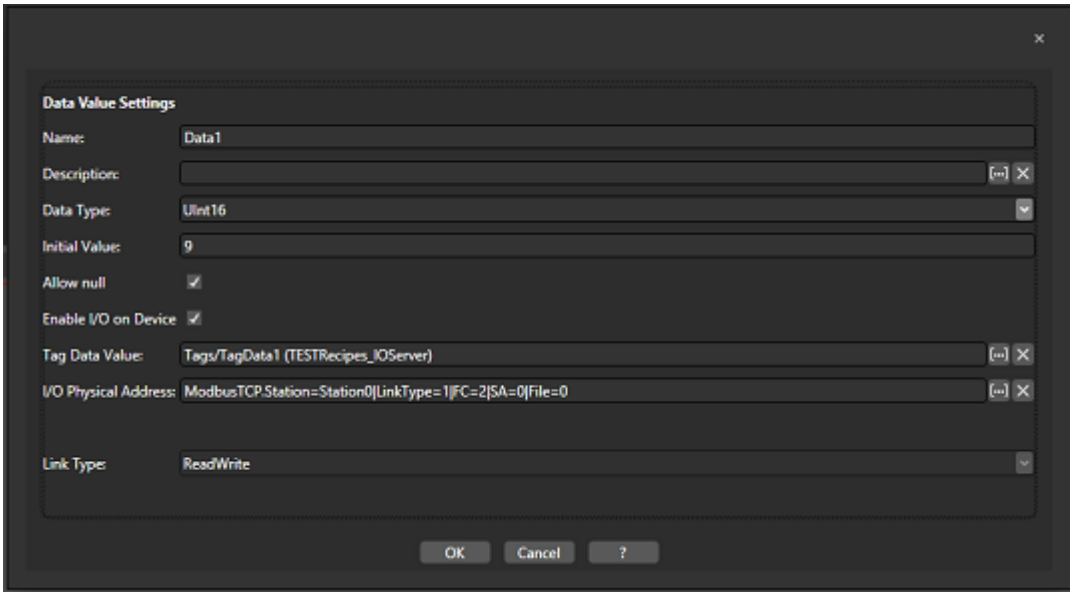
the project Tags for connecting them to the Driver. If you do not enter a start address in this field, the settings from the "Start Address" analog field set for the Recipe will be used instead.

Link Type

Indicates the data exchange task type between the Group and the device. The Recipe can actually be read or written to/from the device, or read only or written only. In this case the link type is determined by the task type defined in the Group's "Start Address" parameter. This field however is read only and gets updated automatically by Movicon based on the settings defined in the "Start Address". When no settings have been defined in the "Start Address" property, this field will be set to "None".

Recipe Data Value property Settings

The Recipe data value settings represent the recipe's true values which are exchanged with the device. They are configured in nearly the same way as those of a normal Tag. The window used for setting properties of a field (or value) belonging to a Recipe opens with a double click or the F4 key after having selected the name of the group belonging to the recipe in the workspace (see figure below). The same parameters are also available in the Properties window once the Recipe has been selected.



Name

Name of the recipe data value.

Description

Indicates the Recipe data value description. Text can be entered as pleased and it will be shown in the Recipe's ""Layout View" wind when inserted in the data value is inserted.

Data Type

This property defines the Recipe's data value type. The options are:

- Boolean
- SByte
- Byte
- Int16
- UInt16
- Int32

- UInt32
- Int64
- UInt64
- Float
- Double
- String

Initial Value

This is used to enter a default value to be assigned to the Recipe data value when a new Recipe is created. If left unassigned, the numeric data value will be initialized with zero and the string data values will be initialized with null.

Allow null

When enabled, a null value can be assigned to the Recipe's data value.

Enable I/O on Device

When enabled the Recipe data value will be exchanged with the device using the Communication Driver. When disabled the vlaue will be recorded on database only and will not be exchanged with the device even when a when a valid Recipe or Group "Start Address" has been defined.

Tag Data Value

This is used to select a Tag (the same type defined for the Data Value) in which the corresponding Recipe value will be inserted during runtime according to which command has been executed (Load Recipe from DB, Read values from device, etc.).

Si sconsiglia l'inserimento dell' "indirizzo I/O Fisico" , ad esempio del Plc con cui andremo a comunicare, direttamente sulla variabile associata alla Ricetta.

I/O Physical Address

This field is used to specify the start address of the area where data is to be exchanged with the connected device. In order for this to function you will need to configure a Communication Driver beforehand. The connection settings are the same ones used in the project Tags for connecting them to the Driver. If this field is not defined with a starting address, the Group data value starting address will be used. Otherwise the one set in the Recipe's if it exists.

Encoding Type

This is used to select the type of encoding to use when saving string data on database. The options are:

- ASCII
- Unicode
- UTF32
- UTF7
- UTF8

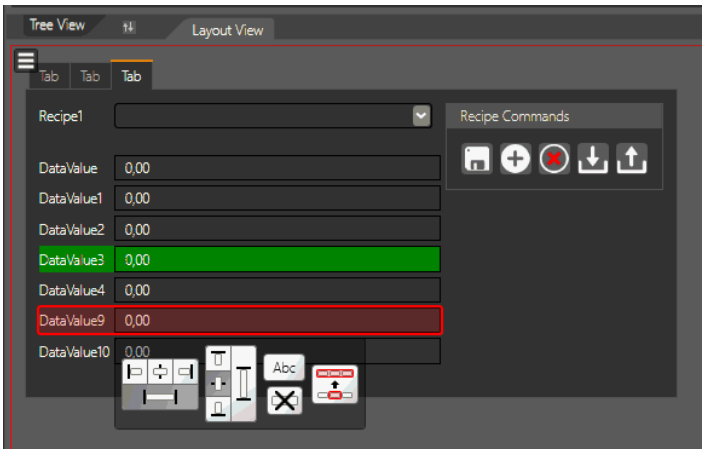
Link Type

This is used to indicate the data exchange task type between the Recipe and device. The Recipe can actually be read or written to/from the device, or read only or written only. In this case the link type is determined by the task type defined in the Group's "Start Address" parameter. This field however is read only and gets

updated automatically by Movicon based on the settings defined in the "Start Address". When no settings have been defined in the "Start Address" property, this field will be set to "None".

1.4. Recipe Layout

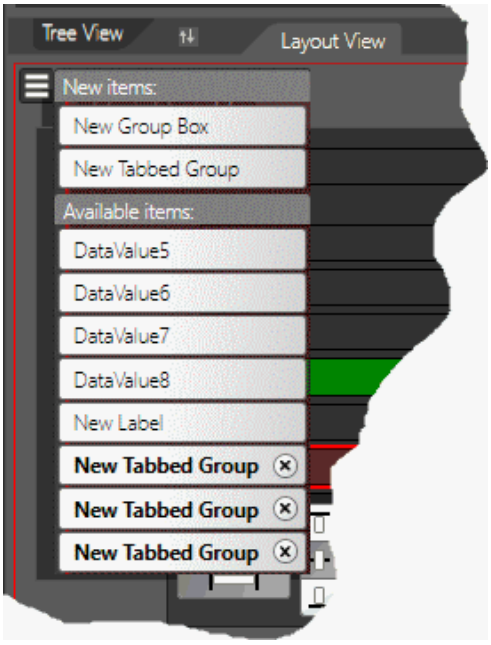
After the Recipe's structure has been created with Groups and Data Values for example, you will then be able to create how the recipe will graphically display in runtime using the Recipe Layout editor.



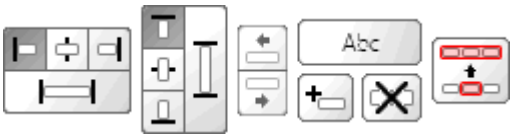
This image shows a Platform.NExT Recipe layout being edited.
As mentioned above, before going ahead with editing the Recipe's Layout you must define the groups and data values using the Configuration Editor.

Inserting data values in the Layout

All of the data values defined in the Recipe's configuration can be inserted in the Layout using the command icon placed a the top left of the Layout Editor window. This icon opens a list of items that can be inserted and positioned in the Layout.



The command list of the fields provides, initially, the availability of all the fields and the structure groups inserted in the graphic layout. then selecting the single field of the list, this will be placed in the Layout Chart and disappear
from the list. If the fields have been included in a group then so it will see only the Group name. Dragging a group we will also be added all defined inside fields.
The items will be inserted in the Layout in their default positions. Once inserted they can be dragged to position as pleased. The position commands from the toolbox shown below can also be used. Each item can be dragged and sized by width and height with the mouse as well. Other style properties, such as Fonts and Colors, can be set using the Property Window.



Each Recipe item can therefore be positioned and sized within the Recipe Layout as pleased by using the appropriate Layout editor commands.
The Layout command from the toolbox allow you to quickly:

- Align and position the selected item
- Enter and modify texts
- Remove items from the layout
- Add or remove TAB items

The TAB items

TAB item can be inserted from the Item List that can be used to divide the Recipe graphical items or Groups in the Layout. Once a TAB item has been positioned in the layout, the other recipe items, such as groups and data values, can be positioned on the TAB as desired.
The TABs can be added and removed using the above mentioned toolbox.

Recipe commands

By using the list of elements you can add a command bar to the recipe layout. This bar is an object containing command buttons that the user can use for performing standard recipe operations. The commands are (from left to right):

- Save Recipe
- Add Recipe
- Delete Recipe
- Import from CSV File
- Export to CSV File
- Transfer to PLC
- Transfer from PLC

The Recipe Command Lock item can be positioned in the Layout as pleased. The command buttons are represented by system icon by default to identify the command.

Nevertheless they can be defined with tooltips containing text by means of using their properties or replace with text strings.

Recipe Controls

When a Recipe data value is inserted in the layout it is assigned a graphical control for default. These controls come in the form of a Check-box for Boolean type data values and a Display for numeric or string data values. These controls can be edited by means of using their properties according to type:

• **CheckBox**: this is a classic check-box that is used for setting the value 0-1 graphically.
• **RadioButton**: by using this control's EnumOptions properties it is possible to select the number of options that can be selected by associating an item list starting with the value 0.
• **EditDisplay**: this is a classical editable display.
• **ComboBox**: by using this control's EnumOptions properties it is possible to insert an item list that can be selected in runtime. The list index starts from the value 0.
• **ListView**: by using this control's EnumOptions properties it is possible to insert an item list that can be selected in runtime. The list index starts from the value 0.

According to the object type selected, it will be possible to define the RecipeEditValueLayoutItem properties:

• **UnitName**: displays the unit name at the side of the value
• **MinValue**: minimum value that can be inserted in the field
• **MaxValue**: maximum value that can be inserted in the field

By using the list of items that can be inserted in the layout and which appears when clicking the icon on the top left of the Editor window, you can also insert following objects:

• **New Group Box**: inserts an empty Group Box in the layout.
• **New Tabbed Group**: inserts a TAB control in the layout as described above.
• **New Label**: inserts a text only label in the layout.

All of the controls which can be inserted in a recipe's layout have "Label" or "Header" properties which represent the name displayed in the layout. They also have a ""ToolTip" property which appears when passing the mouse over the control. These properties are subject to change language. Therefore if ID Strings exit in the string table with their names, they will be managed with the change text language in runtime.

1.5. Recipe Connectivity with PLC and Database

A Platform.NExT Recipe is provided with a database connection to manage historical log filed where the recipe data values are recorded.

All the recipe's data values are also connected directly to the PLC so that data transferral can be managed autonomously. This direct connectivity method reduces the time to transfer data because there is no need to pass through any I/O Data Server polling management to synchronize data in order to complete the transfer.

Database Connectivity

The recipe is provided with the option to connect to a database file through the properties which the recipe index can be set with. The recipe will use the Microsoft SQL Server as set for default in the project's general settings if not specified otherwise. However the recipe's properties can also be used to customize the connection to another database different to the one used for default if desired.

A recipe is saved in the database in one or more tables associated to each other. Usually these tables consist of a main table and a number of secondary tables, one for each data group. Each recipe instance has a data record in each table.

In the figure below there are two tables containing data from the Recipe1 recipe composed of 3 data fields (Data1, Data2, Data3) and a group composed of another 2 data fields (Data1Group1, Data2Group2): in this figure we can see that 3 different recipe instances have been recorded and have been called formula1, formula2 and formula3 respectively.

Risultati | Messaggi

	Recipe1_ID	Recipe1	Recipe1_CreationDateTime	Recipe1_ActivationDateTime	Data1	Data2	Data3
1	30D7517D-5782-433C-A11D-19F535EB7074	formula 3	2016-04-01 14:04:41.2500000	2016-04-01 14:07:34.6600000	3	3	3
2	1CF69842-5D01-4A5C-9A81-2C91A33597F2	formula 2	2016-04-01 14:04:21.8430000	2016-04-01 14:07:48.1030000	2	2	2
3	92119677-C9ED-48EE-B027-5AD96161085C	formula 1	2016-04-01 14:04:06.8200000	2016-04-01 14:07:56.4100000	1	1	1

Risultati | Messaggi

	Recipe1_ID	Data1Group1	Data2Group1
1	30D7517D-5782-433C-A11D-19F535EB7074	3	3
2	1CF69842-5D01-4A5C-9A81-2C91A33597F2	2	2
3	92119677-C9ED-48EE-B027-5AD96161085C	1	1

PLC Connectivity

The recipe can be set with a physical I/O address of a PLC or field device through its index and element properties (group, data value). This will ensure that data is transferred from the database to the PLC and vice-versa in one unique read or write operation towards at least one communication driver set in the project.
This type of connection management ensures the **safe transfer of one unique data block**, that is synchronized and autonomous. This is because It does not pass through the I/O Data Server and therefore does not carry the risk of being fragmented when polled by priority.
Substantially this maximizes security of read and write operations towards the devices.

The "Start Address" can be set at individual "DataValue" level and "Group" level as well as in the actual recipe. The hierarchical order with which the connection address towards the device is used is as follows:
1. Address defined at "datavalue" level
2. Address defined at "group" level
3. Address defined at "recipe" level

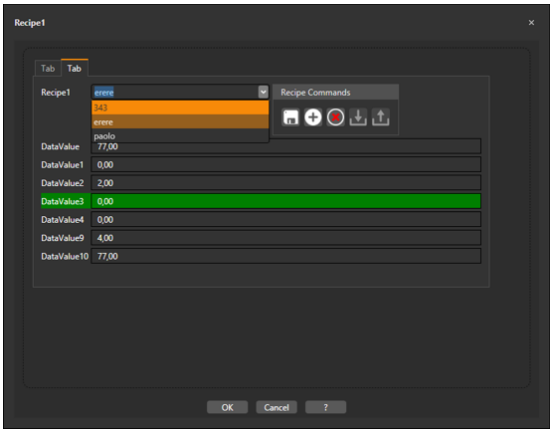
Please also consider that use of the address at Group or Recipe levels is only possible in certain conditions and according to the Communication Driver being used. For example, if using a Communication Driver where the address is set in the variables' properties, each data value must also must be set with the same address. Conversely, if the driver can only be used with absolute addresses and there are several data exchange areas being used (eg. Modbus which have Coil, Holding Register areas, etc.),

then setting the address at Group level requires that the group data be homogenous and with consecutive addresses.

It is the programmer's job to use Platform.NExT correctly to set addresses at each field, data group and entire recipe level by taking into consideration the communication protocol specifications used.

1.6. Recipe Operativity in Runtime

During project runtime it will possible to manage recipes that have been configured in edit mode using the recipe structure and graphical layout settings.
When in runtime mode the recipe can be managed in the Movicon.NExT HMI Client visualization module by using a native recipe pop-up window or the 'Recipe Viewer' object which can be placed in any screen from the object Toolbox.



This image shows a recipe displayed in runtime using a native recipe manager Pop-up window.

Attention: recipe graphics can be shown on the visualization client in two different modes: in a native pop-up window or in a Recipe Viewer object inserted on screen from the toolbox.
Even though graphically there is no difference between the two, only the Recipe Viewer object is supported by Web Client and not the native pop-up window.

Recipe Operativity

To enable correct recipe operativity you will need to make sure that the recipe graphical layout has been configured with the right elements according to the end result you wish to obtain. This entails configuring the recipe data values, tabs, commands and index combo boxes as required.

Recipe Index

The element that shows the Name of the recipe, which can be inserted in any position desired within the recipe layout, is represented by a combo box object. This combo box constitutes the Recipe's index which enables the user to assign names for recipes to be recorded in the database by the system during runtime.
By means of using this element the operator will be able to have:

1. An element for editing names in order to identity each new recipe to be recorded.
2. An element to select and activate a recipe from the comb-box list containing names of those recipes previously entered and recorded.

Save Button

This command saves the selected recipe in the combo box, that was previously added/modified, in the database. This command therefore writes the edited data in the database by recording a new record or modifying an existing one.

New Button

This button is used to add a new Recipe. The Recipe will be added along its default data values but it will not be saved on Database until the Save command is used to do so.

Remove Button

This command is used to remove the currently loaded and displayed Recipe from the database. Recipe removal requires the user to confirm the operation beforehand.

Import Button

This command imports a recipe from a ".csv" file . Naturally the data structure in the file must be coherent with the recipe's structure.

Export Button

This command exports the currently load and displayed recipe to a ".csv" file.

Read Button

This command reads data from the device (PLC, etc.) and updates them in the fields of the Recipe that is currently active at that moment. Therefore it will be possible to save a new Recipe with a different index or overwrite the one which is currently active by using the 'Save' command .

Write Button

The command is used to write the recipe data to the device (PLC, etc.).

1.7. Using Tags to manage Recipes

The Recipe value fields can also be associated to Server Tags. This is very handy for managing Recipes by using Tags and not just by using the "Recipe Window" object's graphical interface and the Pop-up opened by the "Show Recipe" command.

A Tag can be assigned to each recipe value field and other tags to the Recipe Object as listed below (for more details please see the paragraph on the Recipe Properties):

- Recipe List Tag
- Recipe Index Tag
- Recipe State Tag
- Load Recipe Tag
- Save Recipe Tag
- Delete Recipe Tag
- Write Recipe Tag
- Read Recipe Tag

The Recipe Tag List contains the list of recipes existing in the Database. If the name of the recipe entered in the Recipe Index Tag exists on Database, its values will be loaded in the "Value Tags". If the recipe has not yet been saved on database, the "Value Tags" will be initialized with the field's 'initial values'.

From this point onwards, the user will be able to load, save or delete the recipe from the Database or read and write valued from and to field devices using the 'Recipe Commands' from the Platform.NExT's 'Command List. To see whether the operations were performed correctly or if any errors have occurred, the user can run a check on the "Recipe State Tag' bit.

The use of tags will allow you to obtain the maximum flexibility possible in managing recipes. Remember that, for example:

- Maximum recipe layout graphics customization: the tags can in fact be represented on screen by using all the graphics tools provided in the Platform.NExT environment.
- The possibility to manage recipes in background using, for example, script code, without a user being present to perform operations through the client interface.
- The possibility to create embedded recipes. A recipe is defined embedded when its Index Tag is associated to the field of another recipe. In this way, a rather complex and articulated recipe can be subdivided in a number of sub-recipes that are all connected to each other.

Below is a complete list of "Recipe Commands" from the Platform.NExT's 'Command List':

SHOW: activates the recipe's native pop-up window.

LOAD: loads all the recipe's values, that are currently selected in the Recipe Index Tag, from the DB.

SAVE: saves all the recipe's values, that are currently selected in the Recipe Index Tag, in the DB.

REMOVE: Removes the recipe, that is currently selected in the Recipe Index Tag, from the DB.

ACTIVATE: sends the recipe's values, that are currently selected in the Recipe Index Tag, to an external device (eg. PLC).

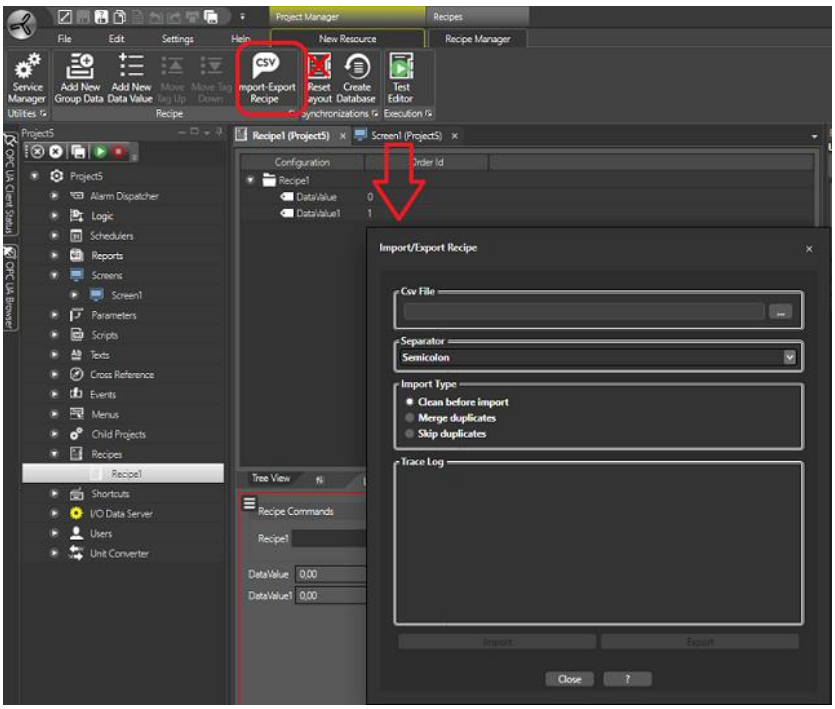
READ: Reads the values that will be assigned to the recipe currently selected in the Recipe Index Tag from the external device (eg. PLC).

EXPORT: Exports the currently selected values in the Recipe Index in a <file.csv>.

IMPORT: Imports the the currently selected values in the Recipe Index from a <file.csv>.

The tags associated to a recipe's value fields can reside both on the NExT server and on other OPC UA servers connected in net.

1.8. Import & Export Recipes



This tool is used to import and export Movicon.NExT recipe structures in CSV. file format.

This file can also be edited using an external editor and then be reimported within the workspace afterwards.

Properties

CSV File: Indicates the name and path of the CSV file which will be used during the import/export procedures.
Separator: This is used to select the column separator used in the CSV. file. The options are:

- Colon
- Comma
- Semicolon
- Tab
- Custom (this lets you specify a custom character to use as the separator)

Import Type: The import modes are:

- **Clean before import:** Eliminates all existing contents within Movicon.NExT before importing the new contents.
- **Merge Duplicates:** Merges duplicates that are detected while being imported.
- **Skip Duplicates:** This allows you to specifiy which duplicated fields to be skipped when found in the CSV file.

Trace Log: logs all what happens during file import/export procedure.

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