EasyBuilder Pro

Recipe Database

This guide walks through basic information and usage of EasyBuilder Pro Recipe Database.

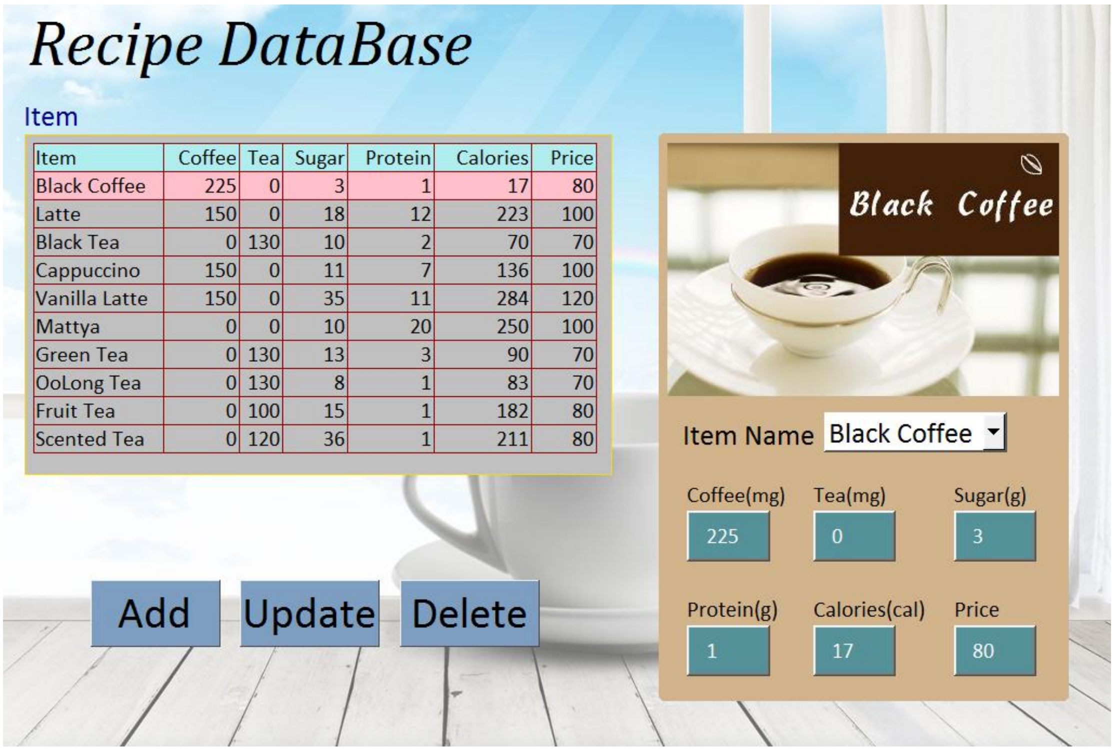
V1.01

Table of Contents

1. Introduction of Recipes

Overview

Recipe DataBase optimized the way of using and editing recipes (RW, RW_A). Recipe DataBase displays the edited recipes in table form, and there's no need to calculate the interval between addresses. Certain Macro functions are provided for searching recipes faster and easier. The process is to build the needed data type in [System Parameter Settings], use [Recipe Records] to enter the values, and then use [Recipe View] object to display the result. The edited recipe data can be operated and adjusted by using other objects.



Item

Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120
Mattya	0	0	10	20	250	100
Green Tea	0	130	13	3	90	70
OoLong Tea	0	130	8	1	83	70
Fruit Tea	0	100	15	1	182	80
Scented Tea	0	120	36	1	211	80

Item Name: Black Coffee

Coffee(mg)	Tea(mg)	Sugar(g)
225	0	3

Protein(g)	Calories(cal)	Price
1	17	80

Features

- Displays recipe data in table form.
- Sorts the records in desired column.
- Searches recipe data by Macro.

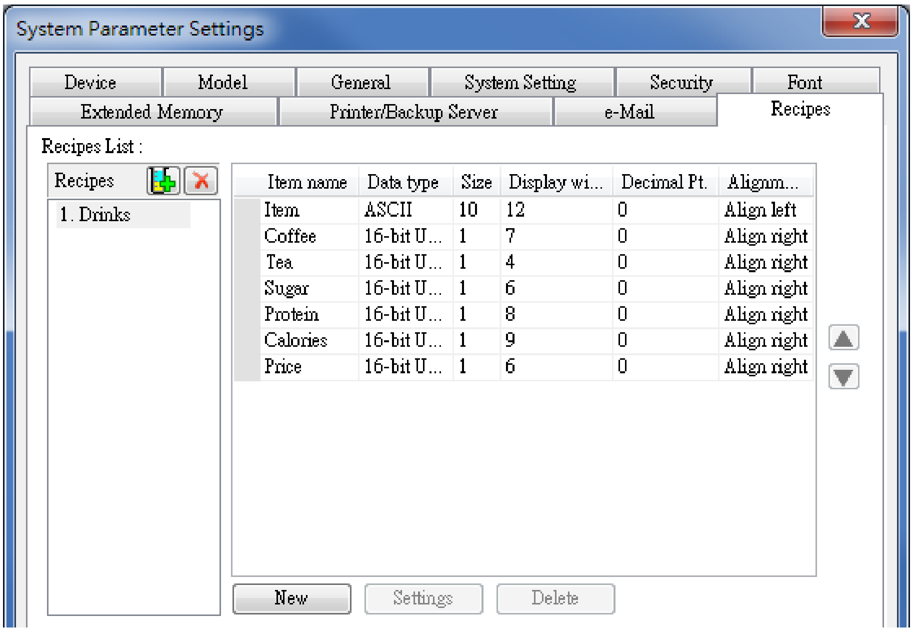
Demo Project

You can refer to Recipe_Database for the samples presented in this document.

Features and Objects

- Recipes Tab

In [System Parameter Settings], [Recipes] tab, add recipes in [Recipe List], and specify item names to define the columns of database.

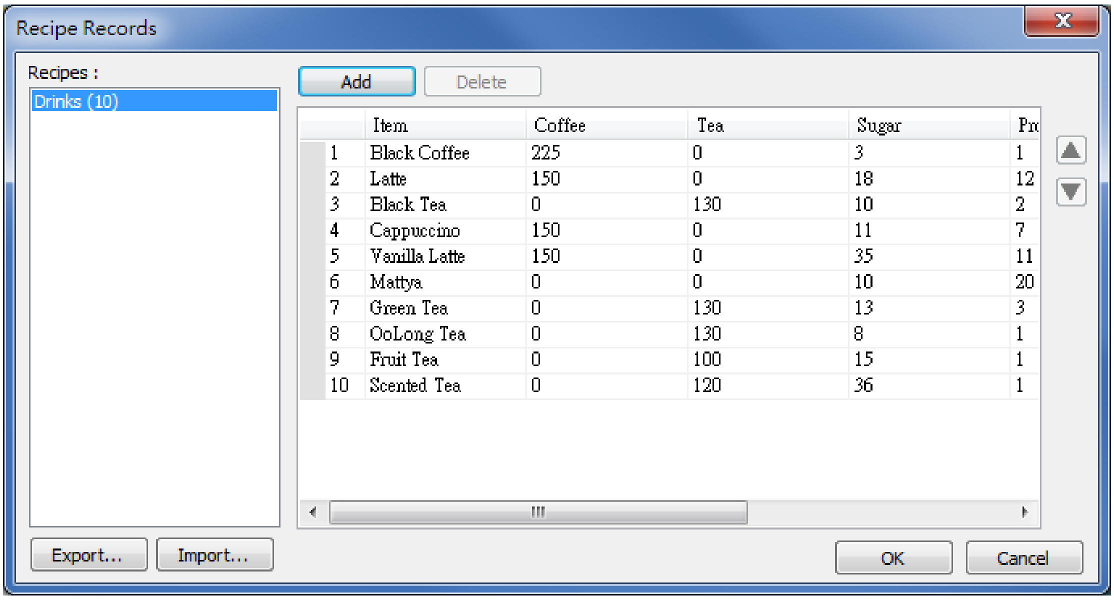


Item name	Data type	Size	Display wi...	Decimal Pt.	Alignm...
Item	ASCII	10	12	0	Align left
Coffee	16-bit U...	1	7	0	Align right
Tea	16-bit U...	1	4	0	Align right
Sugar	16-bit U...	1	6	0	Align right
Protein	16-bit U...	1	8	0	Align right
Calories	16-bit U...	1	9	0	Align right
Price	16-bit U...	1	6	0	Align right

- Recipe Records

Use Recipe Records object to edit recipe contents.

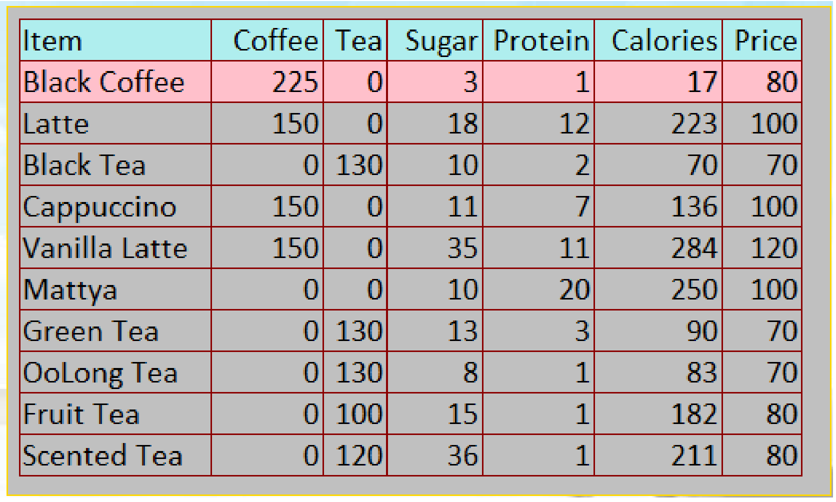
After setting [Recipes] tab in [System Parameter Settings], use [Recipe Records] to add or edit recipe data.



	Item	Coffee	Tea	Sugar	Pr
1	Black Coffee	225	0	3	1
2	Latte	150	0	18	12
3	Black Tea	0	130	10	2
4	Cappuccino	150	0	11	7
5	Vanilla Latte	150	0	35	11
6	Mattya	0	0	10	20
7	Green Tea	0	130	13	3
8	OoLong Tea	0	130	8	1
9	Fruit Tea	0	100	15	1
10	Scented Tea	0	120	36	1

- Recipe View

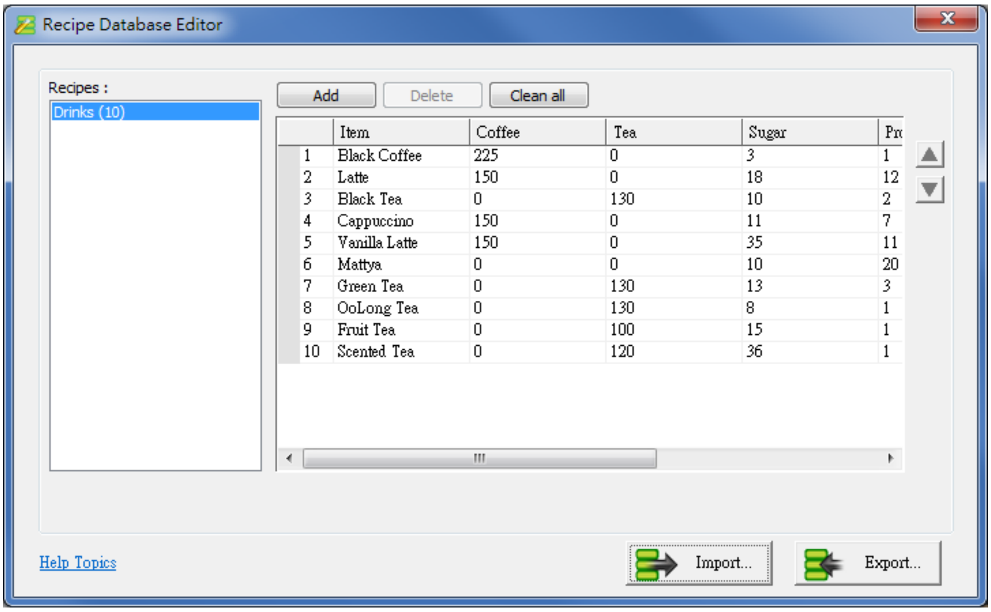
Recipe View object is used for displaying a specific recipe. Users can view all items and values of a recipe by this object, and use relevant registers to monitor or modify recipe records.



Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120
Mattya	0	0	10	20	250	100
Green Tea	0	130	13	3	90	70
OoLong Tea	0	130	8	1	83	70
Fruit Tea	0	100	15	1	182	80
Scented Tea	0	120	36	1	211	80

- Recipe Database Editor

Click the icon to open the [Recipe Database Editor] dialog box. Import a *.db file here and start editing. This tool allows editing recipe without opening EasyBuilder Pro.



	Item	Coffee	Tea	Sugar	Pr
1	Black Coffee	225	0	3	1
2	Latte	150	0	18	12
3	Black Tea	0	130	10	2
4	Cappuccino	150	0	11	7
5	Vanilla Latte	150	0	35	11
6	Mattya	0	0	10	20
7	Green Tea	0	130	13	3
8	OoLong Tea	0	130	8	1
9	Fruit Tea	0	100	15	1
10	Scented Tea	0	120	36	1

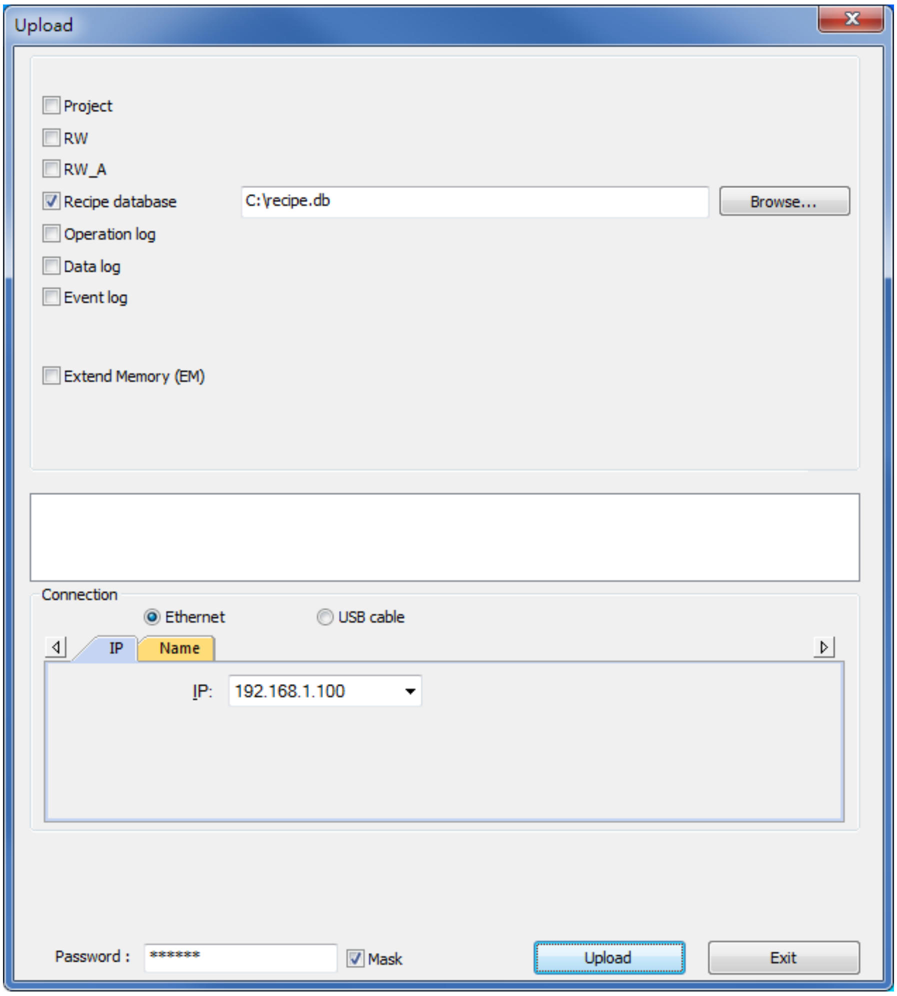
- Recipe Query Functions

  Four Macro functions can be used to query recipe data:

  1. RecipeGet Data: Get recipe data.
  2. RecipeQuery: Query recipe data to obtain the number of records that meets the specified condition.
  3. RecipeQueryGetData: From the result gained by RecipeQuery, get the data of the specific item.
  4. RecipeQueryGetRecordID: From the result gained by RecipeQuery, get the specific record ID.

Upload/Download Recipe Database

Utility Manager offers an option to upload / download recipes. The way is the same as uploading or downloading project files. To upload, click [Utility Manager] and [Upload], select HMI, and then select [Recipe database] check box.



Upload

☐ Project
☐ RW
☐ RW_A
☑ Recipe database C:\recipe.db
☐ Operation log
☐ Data log
☐ Event log

☐ Extend Memory (EM)

Connection
● Ethernet ○ USB cable

IP: 192.168.1.100

Password : ****** ☑ Mask

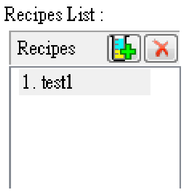
2. How to Build Recipes

Recipe Settings

In [System Parameter Settings], [Recipes] tab, add recipes in [Recipe List], and specify item names to define the columns of database.

Recipe List

Add or delete recipes in this field.



Recipes List :

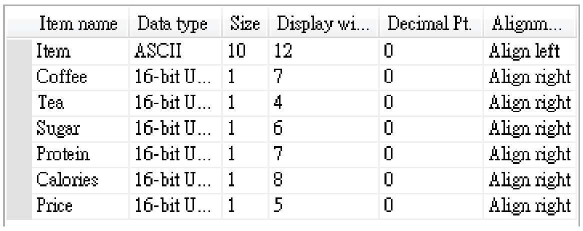
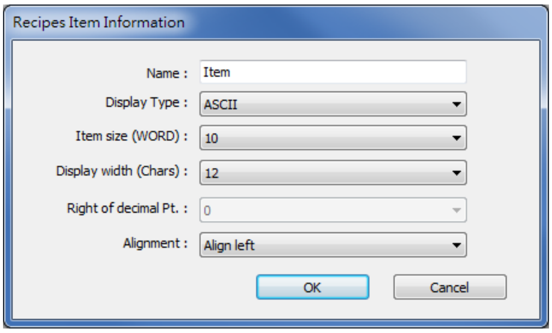
Recipes

1. test1

- Recipe name can’t be repeated.
- Only support alphanumeric names.
- Up to 100 recipes allowed.

Recipe Items

Set the data type, size, display width, decimal point, and alignment to display the recipe items. Up to 1000 recipe items allowed.

Setting	Description
New	Point to an item and click [New], an identical item will be added. 
Settings	Point to an item and click [Settings], the detailed item information is shown and allows users to modify the contents. 
Delete	Delete an existing recipe item.

Item name	Data type	Size	Display wi...	Decimal Pt.	Alignm...
Item	ASCII	10	12	0	Align left
Coffee	16-bit U...	1	7	0	Align right
Tea	16-bit U...	1	4	0	Align right
Sugar	16-bit U...	1	6	0	Align right
Protein	16-bit U...	1	7	0	Align right
Calories	16-bit U...	1	8	0	Align right
Price	16-bit U...	1	5	0	Align right

Recipes Item Information

Name : Item

Display Type : ASCII

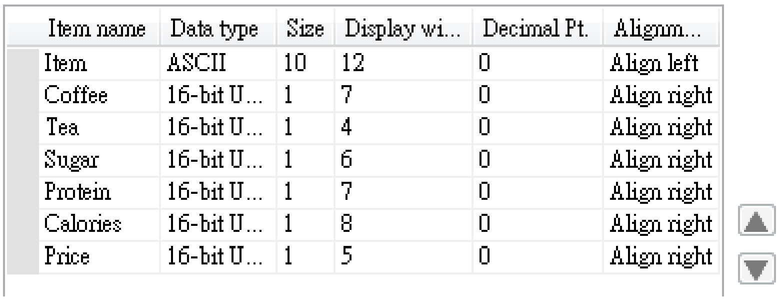
Item size (WORD) : 10

Display width (Chars) : 12

Right of decimal Pt. : 0

Alignment : Align left

Item Settings

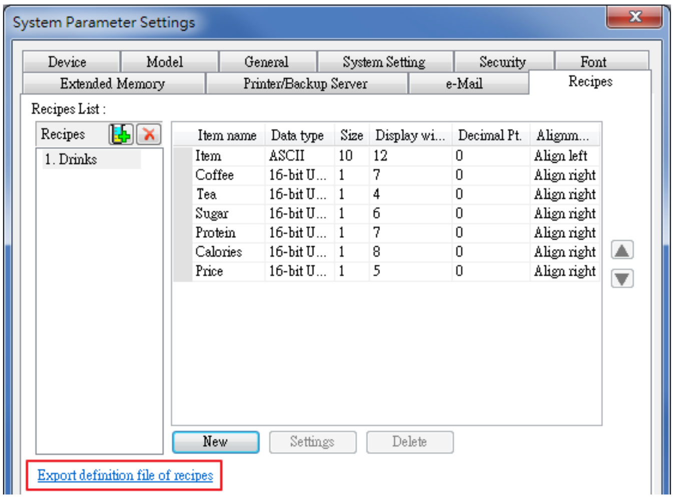


Item name	Data type	Size	Display wi...	Decimal Pt.	Alignm...
Item	ASCII	10	12	0	Align left
Coffee	16-bit U...	1	7	0	Align right
Tea	16-bit U...	1	4	0	Align right
Sugar	16-bit U...	1	6	0	Align right
Protein	16-bit U...	1	7	0	Align right
Calories	16-bit U...	1	8	0	Align right
Price	16-bit U...	1	5	0	Align right

Setting	Description
Item Name	Specifies the item name. Only allows alphanumeric names and "_" symbol.
Data type	The supported data types: 16-bit BCD, 32-bit BCD, 16-bit Hex, 32-bit Hex, 16-bit Binary, 32-bit Binary, 16-bit Unsigned, 32-bit Unsigned, 16-bit Signed, 32-bit Signed, 32-bit Float, ASCII, Unicode, High/Low Reversed, 14 types in total.
Size	Specifies the data length. The data length can only be specified in ASCII, Unicode, High/Low Reversed formats. The limit is 255 words.
Display Width	Specifies the column width in [Recipe View] object.
Decimal Points	Adjusts the number of digits after the decimal point.
Alignment	Aligns recipe data when display them in [Recipe View] object.

Exporting definition file of recipes

Export the item names in the recipe into *.rdef file by using this feature. The exported file will not include the content in Recipe Records.



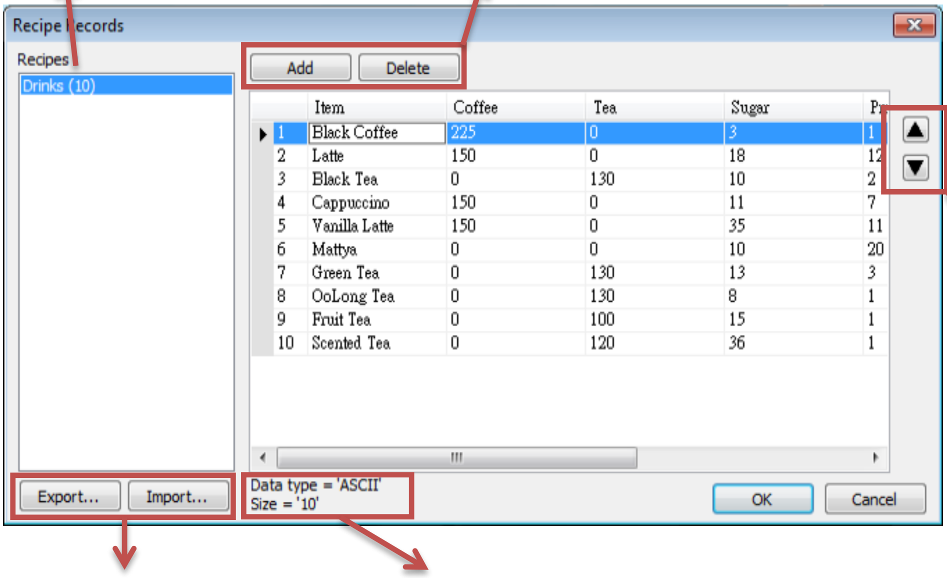
Item name	Data type	Size	Display wi...	Decimal Pt.	Alignm...
Item	ASCII	10	12	0	Align left
Coffee	16-bit U...	1	7	0	Align right
Tea	16-bit U...	1	4	0	Align right
Sugar	16-bit U...	1	6	0	Align right
Protein	16-bit U...	1	7	0	Align right
Calories	16-bit U...	1	8	0	Align right
Price	16-bit U...	1	5	0	Align right

Recipe Records

Open [Recipe Records] by clicking [Library] icon from the Main Menu. Please go to [System Parameter Settings] » [Recipes] tab to build recipe data before using this object.

[Recipe List] Shows the recipes created in [System Parameter Settings]. The number enclosed in parentheses is the total number of records of one recipe.

[Add] / [Delete]
Click [Add] to insert a new item and edit.
Click [Delete] to delete the selected item.



	Item	Coffee	Tea	Sugar	Pr
1	Black Coffee	225	0	3	1
2	Latte	150	0	18	12
3	Black Tea	0	130	10	2
4	Cappuccino	150	0	11	7
5	Vanilla Latte	150	0	35	11
6	Mattya	0	0	10	20
7	Green Tea	0	130	13	3
8	OoLong Tea	0	130	8	1
9	Fruit Tea	0	100	15	1
10	Scented Tea	0	120	36	1

Data type = 'ASCII'
Size = '10'

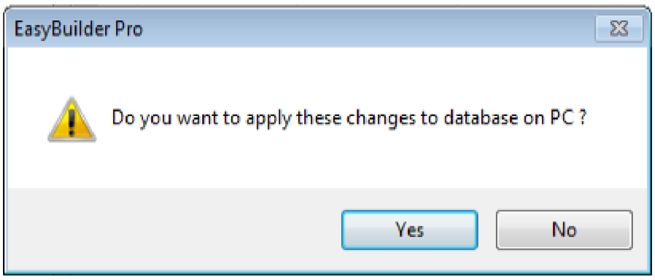
Click to move the selected record.

[Export] / [Import] Exports the current recipe records or imports new data to overwrite current data.

[Item Info]
Shows the data format of the selected item.

Example 1

1. Open [Recipe Records] dialog box to view all the recipe data built in [Recipes] tab.
2. Click [Add] to add a new record and edit the content.
3. At the bottom of the dialog box shows the information of the selected item.
4. Click the up and down buttons to change the order of records.
5. Click [OK], a dialog box shows to confirm if to apply the changes to database on PC. Clicking [Yes] will overwrite the old recipe data.



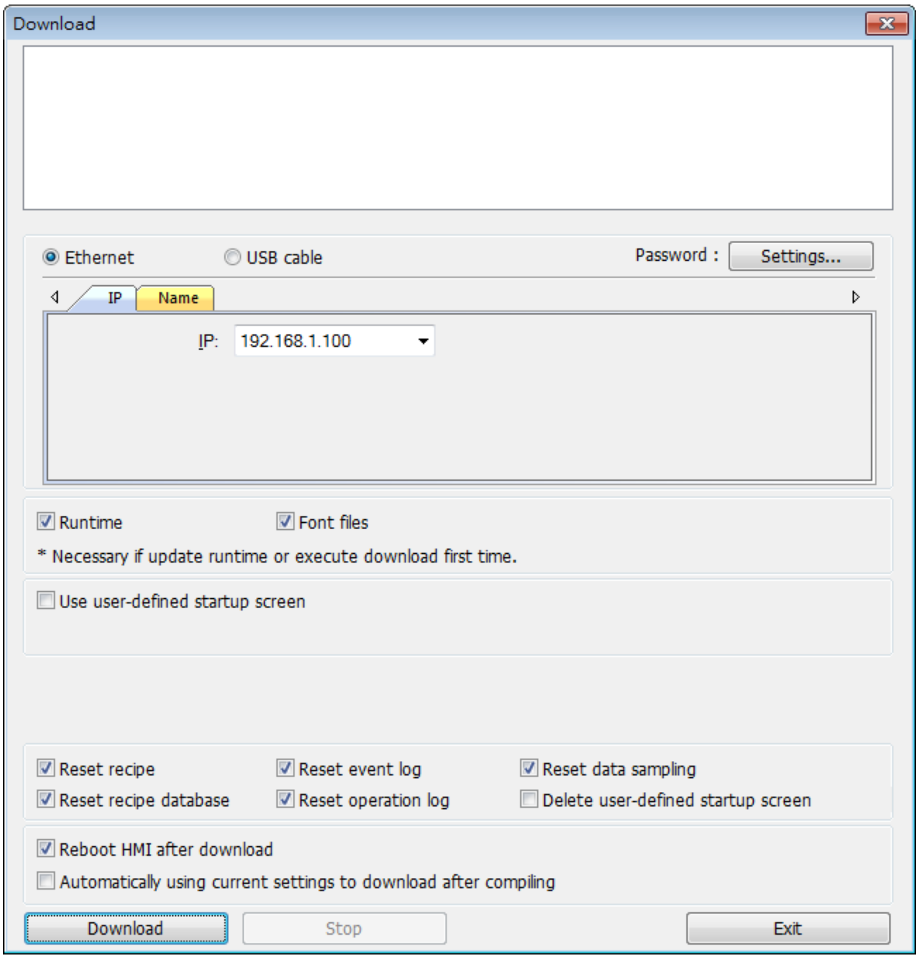
EasyBuilder Pro

Do you want to apply these changes to database on PC ?

6. Click [Export] to export recipe data in *.db file format.

Notes

- Each recipe can hold up to 10000 records.
- If click [Import], the current recipe records and also the recipes built in [System Parameter Settings] » [Recipes] tab will be overwritten by new data.
- The recipe records will be stored in *.exob file after compilation and will be downloaded to the HMI. These recipes are not allowed to be shared with other project files. If users need to modify the recipe content using Recipe Records and to download it to the HMI, make sure to select [Reset recipe database] check box. If not, the recipe database on HMI will not be updated.



Download

● Ethernet ○ USB cable Password : Settings...

IP: 192.168.1.100

☑ Runtime ☑ Font files

* Necessary if update runtime or execute download first time.

☐ Use user-defined startup screen

☑ Reset recipe ☑ Reset event log ☑ Reset data sampling

☑ Reset recipe database ☑ Reset operation log ☐ Delete user-defined startup screen

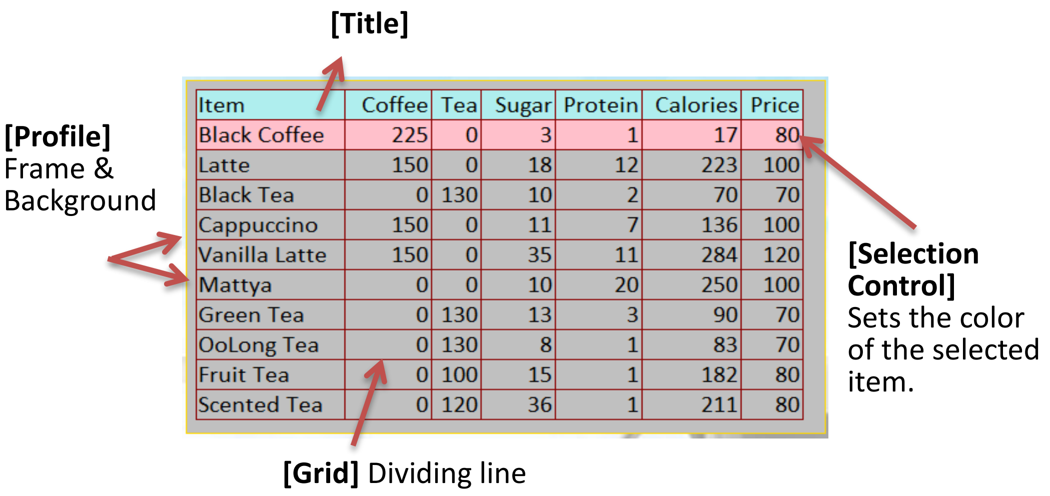
☑ Reboot HMI after download

☐ Automatically using current settings to download after compiling

Recipe View

Recipe View object can display a specific recipe data. Users can use this object to view all items and values of the recipe, and use certain registers to monitor or modify recipe records. Before using this object, please build recipe first using Recipe Records.

General



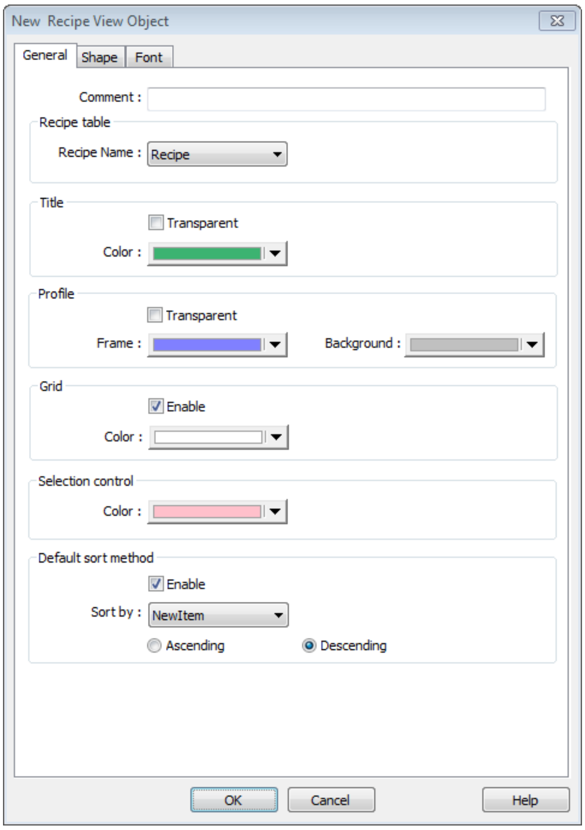
Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120
Mattya	0	0	10	20	250	100
Green Tea	0	130	13	3	90	70
OoLong Tea	0	130	8	1	83	70
Fruit Tea	0	100	15	1	182	80
Scented Tea	0	120	36	1	211	80

[Title]

[Profile]
Frame &
Background

[Selection Control]
Sets the color of the selected item.

[Grid] Dividing line



New Recipe View Object

General | Shape | Font

Comment :

Recipe table

Recipe Name : Recipe

Title

□ Transparent

Color : 🟩

Profile

□ Transparent

Frame : 🟦 Background : ▨

Grid

☑ Enable

Color : ⬜

Selection control

Color : 🟥

Default sort method

☑ Enable

Sort by : NewItem

○ Ascending ● Descending

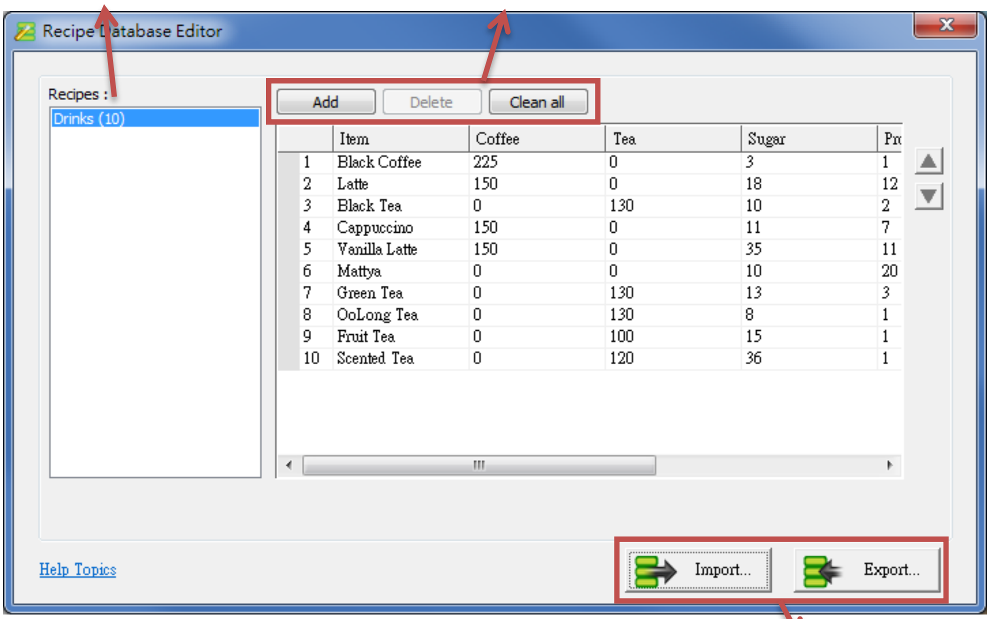
Setting	Description
Recipe table	**Recipe Name** Select a recipe name from the drop-down list.
Title	Each item has a title. The title is referring to the setting in [System Parameter Settings] » [Recipes] tab. **Transparent** If this check box is selected, the title row wouldn't show background color, and the color selection will not be available.
Profile	Sets the frame and background color of the object. **Transparent** If this check box is selected, the frame and background will be transparent, and the color selection will not be available.
Grid	The dividing line divides each data. **Transparent** If this check box is selected, the dividing line will be transparent, and the color selection will not be available.
Selection control	The displayed color when pointing to a specific row.
Default sort method	Sorts records in Recipe View object in Ascending or Descending order.

Recipe Database Editor

Use this tool to edit recipe data without opening EasyBuilder Pro, and then upload or download recipes by using Utility Manager.

[Recipe List]
Shows the recipes created in [System Parameter Settings]. The number enclosed in parentheses is the total number of records in one recipe.

[Add] / [Delete]
Click [Add] to insert a new item and edit. Click [Delete] to delete the selected item.



	Item	Coffee	Tea	Sugar	Pr
1	Black Coffee	225	0	3	1
2	Latte	150	0	18	12
3	Black Tea	0	130	10	2
4	Cappuccino	150	0	11	7
5	Vanilla Latte	150	0	35	11
6	Mattya	0	0	10	20
7	Green Tea	0	130	13	3
8	OoLong Tea	0	130	8	1
9	Fruit Tea	0	100	15	1
10	Scented Tea	0	120	36	1

[Import] / [Export]
Import *.db file for editing, and then export after editing.

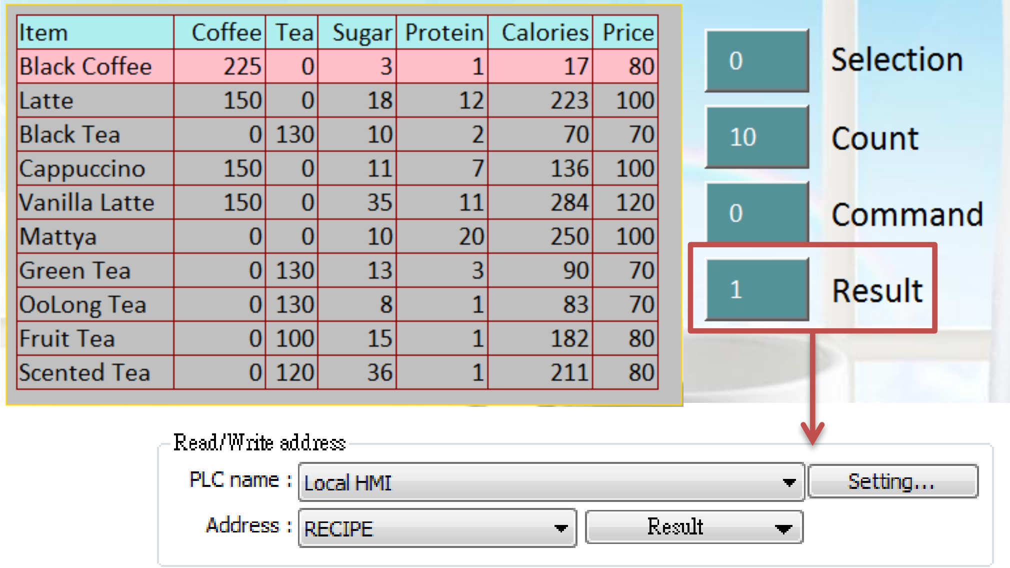
Example 2

1. Click [Recipe Database Editor] application to open the editing dialog box.
2. Click [Import] to import *.db files and edit recipe contents.
3. After editing click [Export] to save the file to *.db format.
4. Download Recipe DataBase by using Utility Manager. When downloading, if select [Reset recipe database], the Recipe Database in HMI will be overwritten with the new settings.

3. Monitoring and Modifying Recipe Records

Monitoring Recipe Data

To watch / add / delete the displayed records, certain registers can be used. Create 4 Numeric Input objects, set addresses respectively to: RECIPE-Selection, RECIPE-Count, RECIPE-Command, and RECIPE-Result.



Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120
Mattya	0	0	10	20	250	100
Green Tea	0	130	13	3	90	70
OoLong Tea	0	130	8	1	83	70
Fruit Tea	0	100	15	1	182	80
Scented Tea	0	120	36	1	211	80

0	Selection
10	Count
0	Command
1	Result

Read/Write address

PLC name : Local HMI

Address : RECIPE Result

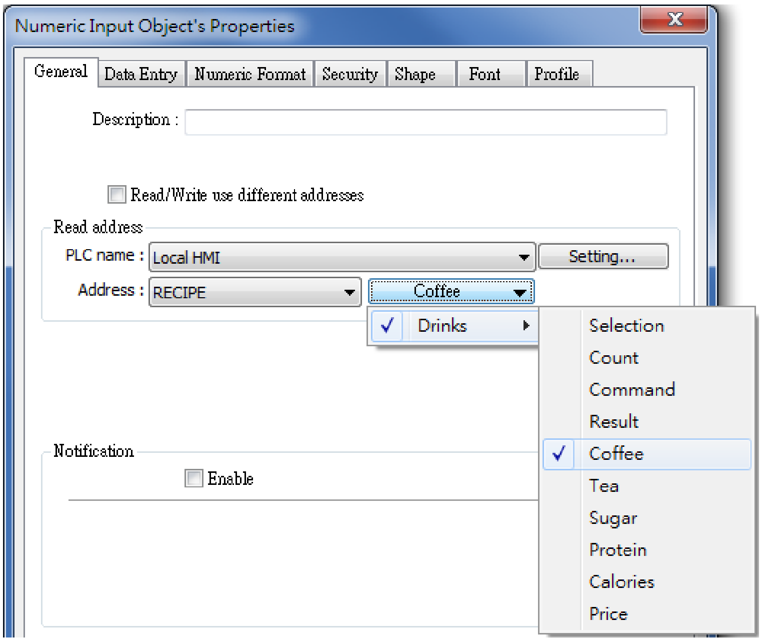
Setting	Description
Selection	The currently selected record. The records are numbered from zero. If choose the first record, the value of Selection will show "0", and so on.
Count	The number of records in the recipe.
Command	Entering certain values will send certain commands to the selected record. Enter "1", adds a new recipe record to the last row. Enter "2", updates the selected recipe record. Enter "3", deletes the selected recipe record. Enter "4", deletes all recipe data.
Result	View the result of executing commands. Displays "1", command successfully executed. Displays "2", the record does not exist. Displays "4", unknown command. Displays "8", records reach limit (10000 records), no new records can be added.

Recipe Name

The recipe data can be displayed if the item name in the recipe is selected in [Address].

[Numeric Input / Display]: Adjust the digits after the decimal point correctly and set the upper and lower limit.

[ASCII Input / Display]: Set the number of words.



Numeric Input Object's Properties

Description :

☐ Read/Write use different addresses

Read address

PLC name : Local HMI

Address : RECIPE Coffee

Notification

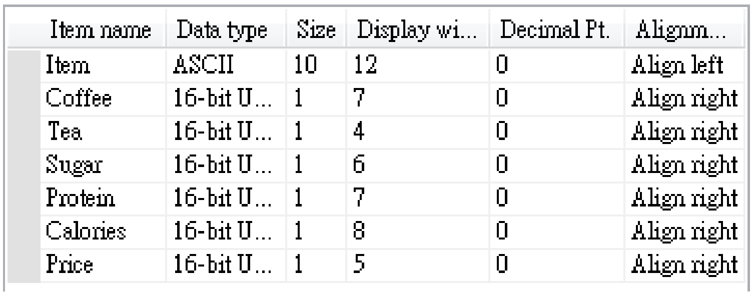
☐ Enable

Modifying Recipe Data on HMI

To modify recipe data, please create Numeric Input or ASCII Input objects first. Select the recipe item for address. After modifying, enter “2” in Command register to update.

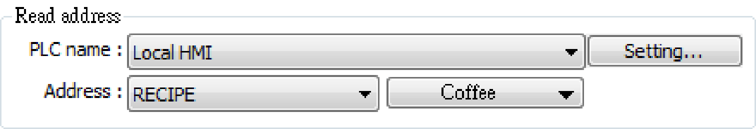
Example 3

To use this demo project, please add a recipe in [System Parameter Settings] » [Recipes] tab first, and then edit the content in [Recipe Records].



Item name	Data type	Size	Display wi...	Decimal Pt.	Alignm...
Item	ASCII	10	12	0	Align left
Coffee	16-bit U...	1	7	0	Align right
Tea	16-bit U...	1	4	0	Align right
Sugar	16-bit U...	1	6	0	Align right
Protein	16-bit U...	1	7	0	Align right
Calories	16-bit U...	1	8	0	Align right
Price	16-bit U...	1	5	0	Align right

1. Create a [Numeric Input] object; select the item to modify in address field. Adjust the digits after the decimal point and set the upper and lower limit.

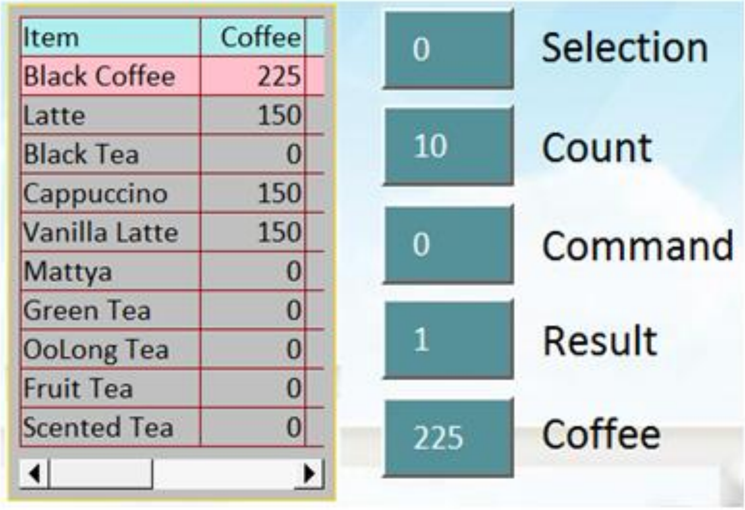


Read address

PLC name : Local HMI ▾ Setting...

Address : RECIPE ▾ Coffee ▾

2. In Recipe View object, select the record to modify or enter the number of it in Selection field and then enter the new value to the corresponding register.



Item	Coffee
Black Coffee	225
Latte	150
Black Tea	0
Cappuccino	150
Vanilla Latte	150
Mattya	0
Green Tea	0
OoLong Tea	0
Fruit Tea	0
Scented Tea	0

0	Selection
10	Count
0	Command
1	Result
225	Coffee

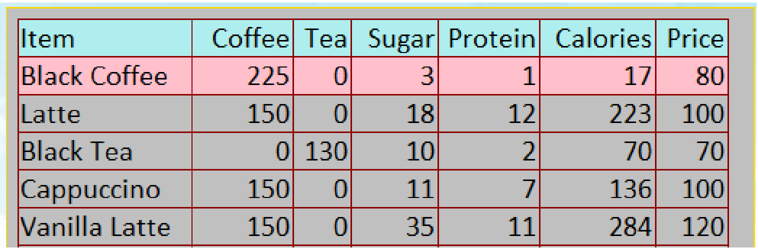
3. Enter “2” in Command register to update. Please note that entering “2” in Command will complete updating Recipe DataBase, and the setting in LB-9029 is irrelevant.

Transferring Recipe Data

The edited recipe can be operated or adjusted using [Transfer (Trigger-Based) object or the designated register.

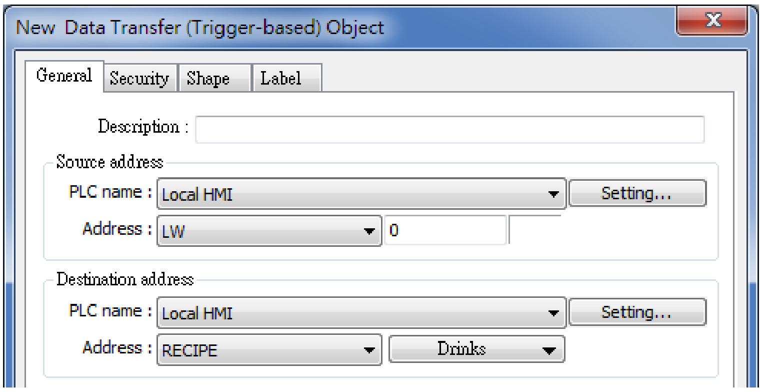
Example 4

This example explains how to transfer a complete recipe data. As shown in the following figure, the data to be transferred contains the following information: Item Coffee, Tea, Sugar, Protein, Calories.



Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120

1. Create a Data Transfer (Trigger-based) object; designate the destination address to a specific recipe.



Description :

Source address

PLC name : Local HMI

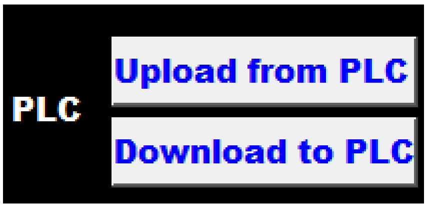
Address : LW 0

Destination address

PLC name : Local HMI

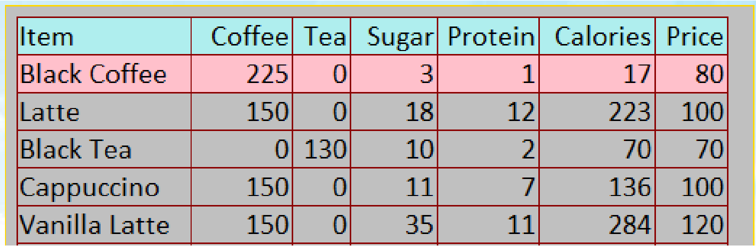
Address : RECIPE Drinks

2. Create a local address object; the data format must be set identically to the recipe. For example, if a recipe includes two data types: 16-BCD and 32-BCD, the local address must set the same: LW-0 -> 16-BCD, LW-1->32-BCD.
3. In Recipe View object select the record to be transferred, or enter the number of the record in Selection.
4. Click Data Transfer (Trigger-based) object to transfer data. If transfer PLC data to Recipe register, enter “2” in Command register to finish updating.



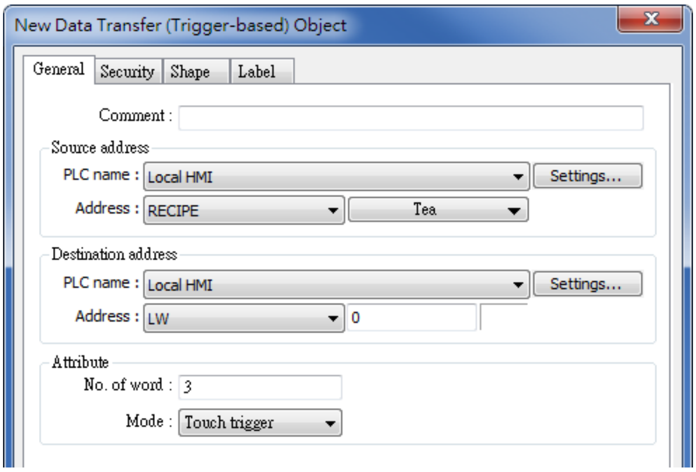
Example 5

This example explains how to transfer recipe data of specified column. As shown in the following figure, when attempting to transfer data in Tea, Sugar, Protein columns, please follow the steps described.



Item	Coffee	Tea	Sugar	Protein	Calories	Price
Black Coffee	225	0	3	1	17	80
Latte	150	0	18	12	223	100
Black Tea	0	130	10	2	70	70
Cappuccino	150	0	11	7	136	100
Vanilla Latte	150	0	35	11	284	120

1. Create a Data Transfer (Trigger-based) object, set the source address to “Tea”, the destination address to LW-0, and number of words to 3.



New Data Transfer (Trigger-based) Object

General | Security | Shape | Label

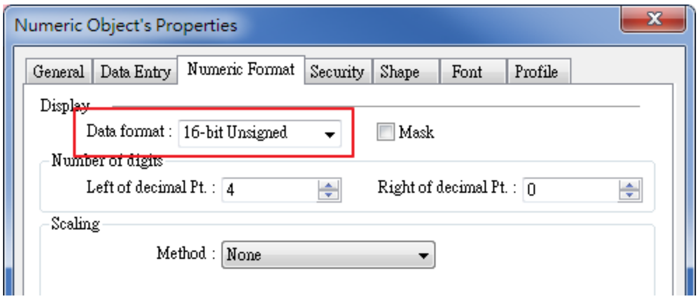
Comment :

Source address
PLC name : Local HMI
Address : RECIPE | Tea

Destination address
PLC name : Local HMI
Address : LW 0

Attribute
No. of word : 3
Mode : Touch trigger

2. Create three Numeric Input objects, set addresses respectively to LW-0, LW-1, LW-2. The value format must be identical to the settings in Recipe. As shown in the demo project, the format of Tea, Sugar, Protein items is 16-bit Unsigned, please set the same format in addresses LW-0, LW-1, LW-2.



Numeric Object's Properties

General | Data Entry | Numeric Format | Security | Shape | Font | Profile

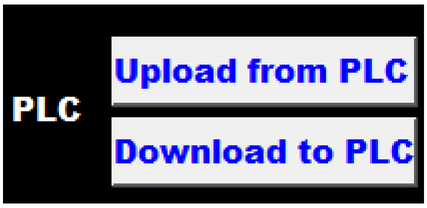
Display
Data format : 16-bit Unsigned
☐ Mask

Number of digits
Left of decimal Pt. : 4
Right of decimal Pt. : 0

Scaling
Method : None

3. In Recipe View object select the item to be transferred, or enter the number of the item in Selection.

4. Click Data Transfer (Trigger-based) object to transfer data. If transfer PLC data to Recipe register, enter “2” in Command register to finish updating.

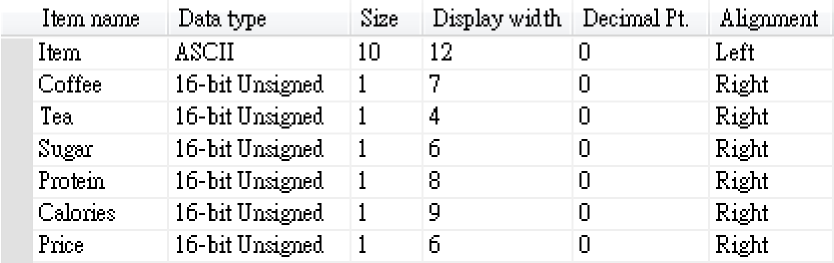


Reading and Writing Bits in Recipe DataBase

The bit address of recipe item can be read / written. This feature only supports Unsigned format.

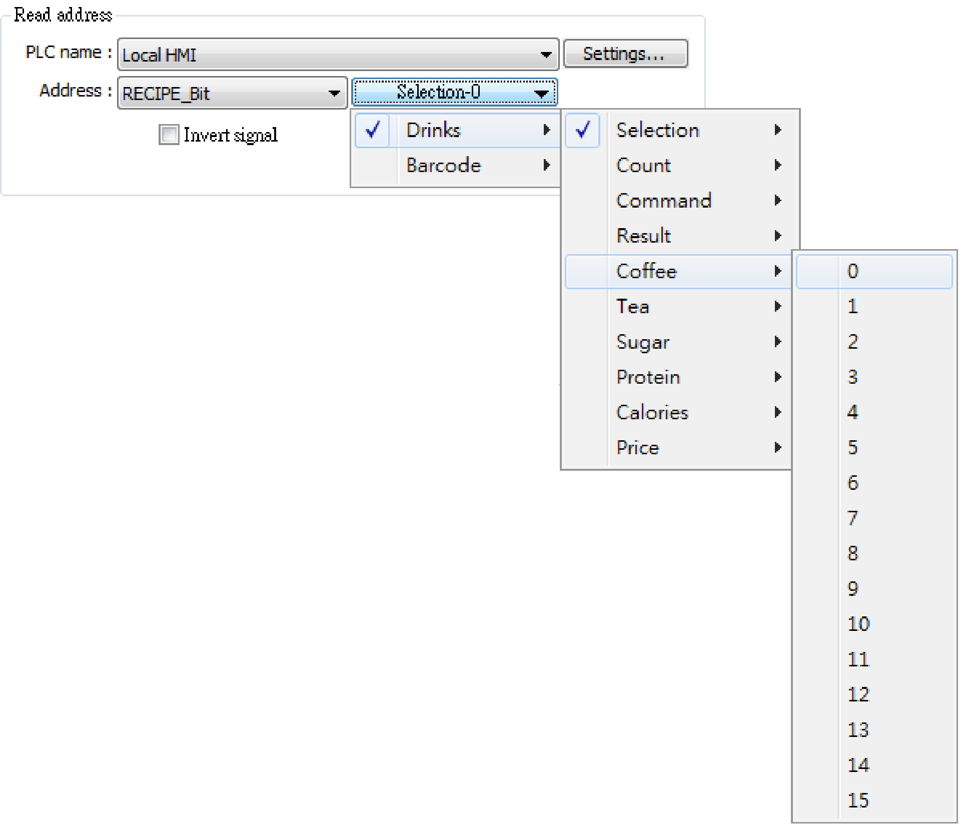
Example 6

1. As shown in the following figure, except for "Item", the data type of the rest items is 16-bit Unsigned.



Item name	Data type	Size	Display width	Decimal Pt.	Alignment
Item	ASCII	10	12	0	Left
Coffee	16-bit Unsigned	1	7	0	Right
Tea	16-bit Unsigned	1	4	0	Right
Sugar	16-bit Unsigned	1	6	0	Right
Protein	16-bit Unsigned	1	8	0	Right
Calories	16-bit Unsigned	1	9	0	Right
Price	16-bit Unsigned	1	6	0	Right

2. Create a Bit object, set the address to Recipe_Bit. When pointing to an item, its available number of bits will be displayed automatically. As shown in the following figure, the item "Coffee" can have 16 bits.



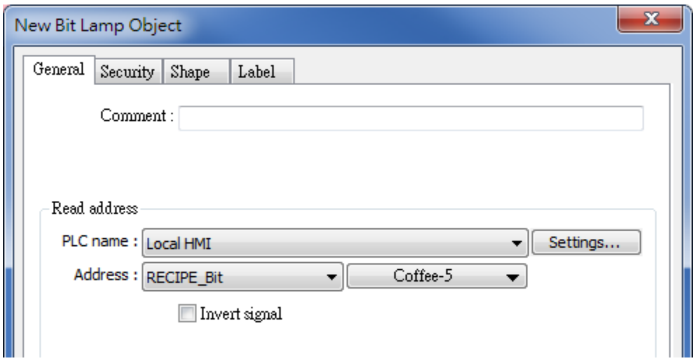
Read address

PLC name : Local HMI

Address : RECIPE_Bit Selection-0

☐ Invert signal

3. Select the read/write address. The address will be "Recipe_Bit\item name\*bit number". *As shown in the following figure, the 6th bit of Coffee is displayed in the address field.


New Bit Lamp Object

Comment :

Read address

PLC name : Local HMI

Address : RECIPE_Bit   Coffee-5

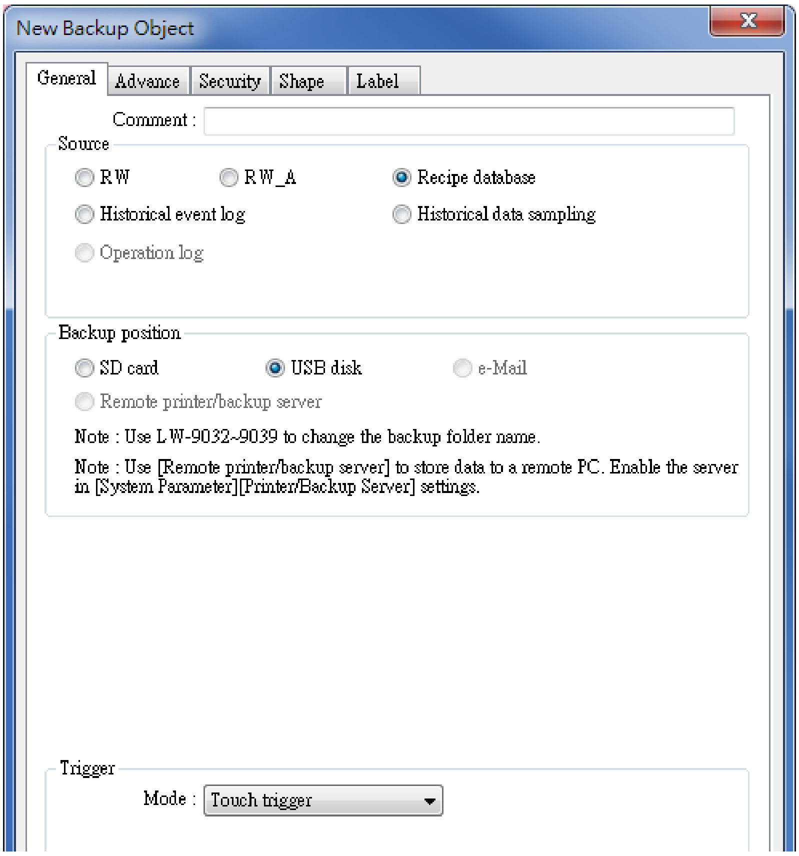
□ Invert signal

Backup Recipe DataBase

Backup object can be used to backup Recipe DataBase into USB drive / SD card, or send the data to the designated email box. The format of the backup data is .db.

Example 7

Create a Backup object, select Recipe DataBase for source, and select the position to save the data.


New Backup Object

Comment :

Source

○ RW ○ RW_A ● Recipe database

○ Historical event log ○ Historical data sampling

○ Operation log

Backup position

○ SD card ● USB disk ○ e-Mail

○ Remote printer/backup server

Note : Use LW-9032~9039 to change the backup folder name.

Note : Use [Remote printer/backup server] to store data to a remote PC. Enable the server in [System Parameter][Printer/Backup Server] settings.

Trigger

Mode : Touch trigger

Searching Recipe Data by Macros

Recipe Query Functions enable searching a specific ID or data in a recipe.

Four Functions are used to query recipe data:

1. RecipeGet Data: Get recipe data.
2. RecipeQuery: Query recipe data to obtain the number of records that meet the specified condition.
3. RecipeQueryGetData: From the result gained by RecipeQuery, get the data of the specific item.
4. RecipeQueryGetRecordID: From the result gained by RecipeQuery, get the specific record ID.

Name	RecipeGetData
Syntax	RecipeGetData (destination, recipe_address, record_ID)
Description	Get Recipe Data. The gained data will be stored in destination, and must be a variable. “recipe_address” consists of type name and item name: “recipetype_name.item_name”. “record_ID” specifies the ID number of the record in recipe being gained.

Example	macro_command main()
	int data=0
	char str[20]
	int recordID
	bool result
	
	recordID = 0
	result = RecipeGetData(data, "TypeA.item_weight", recordID)
	// From recipe "TypeA" get the data of the item “item_weight” in record 0.
	
	recordID = 1
	result = RecipeGetData(str[0], "TypeB.item_name", recordID)
	// From recipe "TypeB" get the data of the item “item_name” in record 1.
	
	end macro_command

Name	RecipeQuery
Syntax	RecipeQuery (SQL command, destination)
Description	Use SQL statement to query recipe data. The number of records of query result will be stored in the destination. This must be a variable. SQL command can be static string or char array. Example: RecipeQuery(“SELECT * FROM TypeA”, destination) or RecipeQuery(sql[0], destination) SQL statement must start with ”SELECT * FROM” followed by type name and query condition.

Example	macro_command main()
	
	int total_row=0
	char sql[100]="SELECT * FROM TypeB"
	bool result
	
	result = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	
	result = RecipeQuery(sql[0], total_row)
	// Query Recipe "TypeB". Store the number of records of query result in total_row.
	
	end macro_command

Name	RecipeQueryGetData
Syntax	RecipeQueryGetData (destination, recipe_address, result_row_no)
Description	Get the data in the query result obtained by RecipeQuery. This function must be called after calling RecipeQuery, and specify the same type name in recipe_address as RecipeQuery. result_row_no specifies the sequence row number in query result.
Example	macro_command main() int data=0 int total_row=0 int row_number=0

	bool result_query
	bool result_data
	
	result_query = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	
	if (result_query) then
	  for row_number=0 to total_row-1
	    result_data = RecipeQueryGetData(data, "TypeA.item_weight", row_number)
	  next row_number
	end if
	
	end macro_command

Name	RecipeQueryGetRecordID
Syntax	RecipeQueryGetRecordID (destination, result_row_no)
Description	Get the record ID numbers of those records gained by RecipeQuery. This function must be called after calling RecipeQuery. result_row_no specifies the sequence row number in query result, and write the obtained record ID to destination.
Example	macro_command main() int recordID=0 int total_row=0 int row_number=0 bool result_query bool result_id result_query = RecipeQuery("SELECT * FROM TypeA", total_row) // Query Recipe "TypeA". Store the number of records of query result in total_row. if (result_query) then for row_number=0 to total_row-1 result_id = RecipeQueryGetRecordID(recordID, row_number)

```
  next row_number
end if

end macro_command
```

4. References

- For details of [Recipes] setting tab please refer to EasyBuidler Pro User Manual Chapter 5.10 or click here: Chapter_05_System_Parameter_Settings
- For details of [Recipe Records] please refer to EasyBuidler Pro User Manual Chapter 24.3 or click here: Chapter_24_Recipe_Editor
- For details of [Recipe View] please refer to EasyBuidler Pro User Manual Chapter 13.33 or click here: Chapter_13_Objects
- For details of macro, please refer to EasyBuidler Pro User Manual Chapter 18.6.7 or click here: Chapter_18_Macro_Reference
- Demo Project of Transferring Recipe Data: Recipe Transferring
- Demo Project of Searching Recipe data by Macros: Macro Recipe
- Demo Project of Backing up Recipe data into USB drive: Backup_Recipe_Database_to_USB_Demo
- Example of how to export Database files to CSV format: FAQ_50_Export_Recipe_DataBase_to_CSV_file