

CHAPTER 10

RECIPES AND RECIPES OBJECTS

| | |
|---|-----------|
| 10.1. Recipes | 1 |
| 10.2. Recipe Data Flow and Memory Allocation | 2 |
| 10.3. Working with Recipes | 3 |
| 10.4. Transferring Recipes | 4 |
| 10.4.1. Between PC and PM (HMI) | 4 |
| 10.4.2. Between PM (HMI) and USB Memory Stick/Micro SD Card..... | 4 |
| 10.4.3. Between battery backed memory and flash ROM..... | 4 |
| 10.4.4. Between PM (HMI) and PLC/Controller | 5 |
| 10.5. Setting up Recipe Blocks..... | 6 |
| 10.5.1. General Settings..... | 6 |
| 10.5.2. Data Item Settings..... | 9 |
| 10.6. Selecting a Recipe Using Recipe Selectors | 11 |
| 10.6.1. Basic Operations..... | 11 |
| 10.6.2. Operation Options..... | 12 |
| 10.6.3. Settings | 12 |
| 10.6.4. General Settings..... | 12 |
| 10.7. Displaying and Modifying Recipe Data Using Recipe Tables | 14 |
| 10.7.1. Basic Operations..... | 14 |
| 10.7.2. Operation Options..... | 15 |
| 10.7.3. Settings | 15 |
| 10.7.4. General Settings..... | 15 |
| 10.7.5. Data Item Settings..... | 17 |

In this chapter we will explain how recipes in the software can be set up, stored and transferred. We also describe how to configure the basic functions and recipe objects (recipe selector and recipe table) used for recipes.

10.1. Recipes

■ Recipe Block

A recipe block is a memory block stored recipe data as a two dimensional array in the panel. The memory size of each recipe block is the product of the size of a recipe by number of recipes in word. You can create up to 16 recipe blocks for your application. Each recipe block can contain at most 65535 recipes. Each recipe can have as many as 4096 words of data.

■ Recipe

A recipe is a group of data items. You can use a recipe number or a recipe name to index a recipe in the corresponding recipe block.

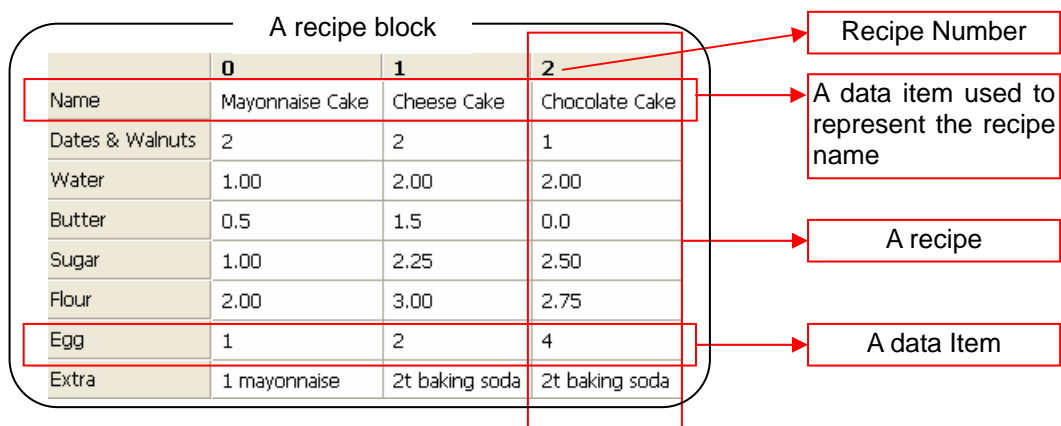
The recipe number is a sequence number between 0 and number of recipes. It is unique among all recipes of the recipe block. The current recipe number of the recipe block m is saved in the current recipe number register \$RNM (m: The recipe block ID).

The recipe name can be represented by a specified ASCII or Unicode String data item of the recipe. To specify an ASCII or Unicode String data item as the recipe name, you need to open the dialog box of a recipe block and set the data type of the related data item into ASCII String or Unicode String in the data item page. For details about recipe data item settings, please see [Section 10.5.2](#).

■ Recipe Data Item

A data item is a word or words of data used to represent an application related data or a machine setup parameter used in process and production control. You can specify the format such as name, data type, size, scaling and range check for each data item in recipe block dialog box.

The following is a sample of a recipe block with 3 recipes and each recipe has 8 data items.



■ Recipe Data

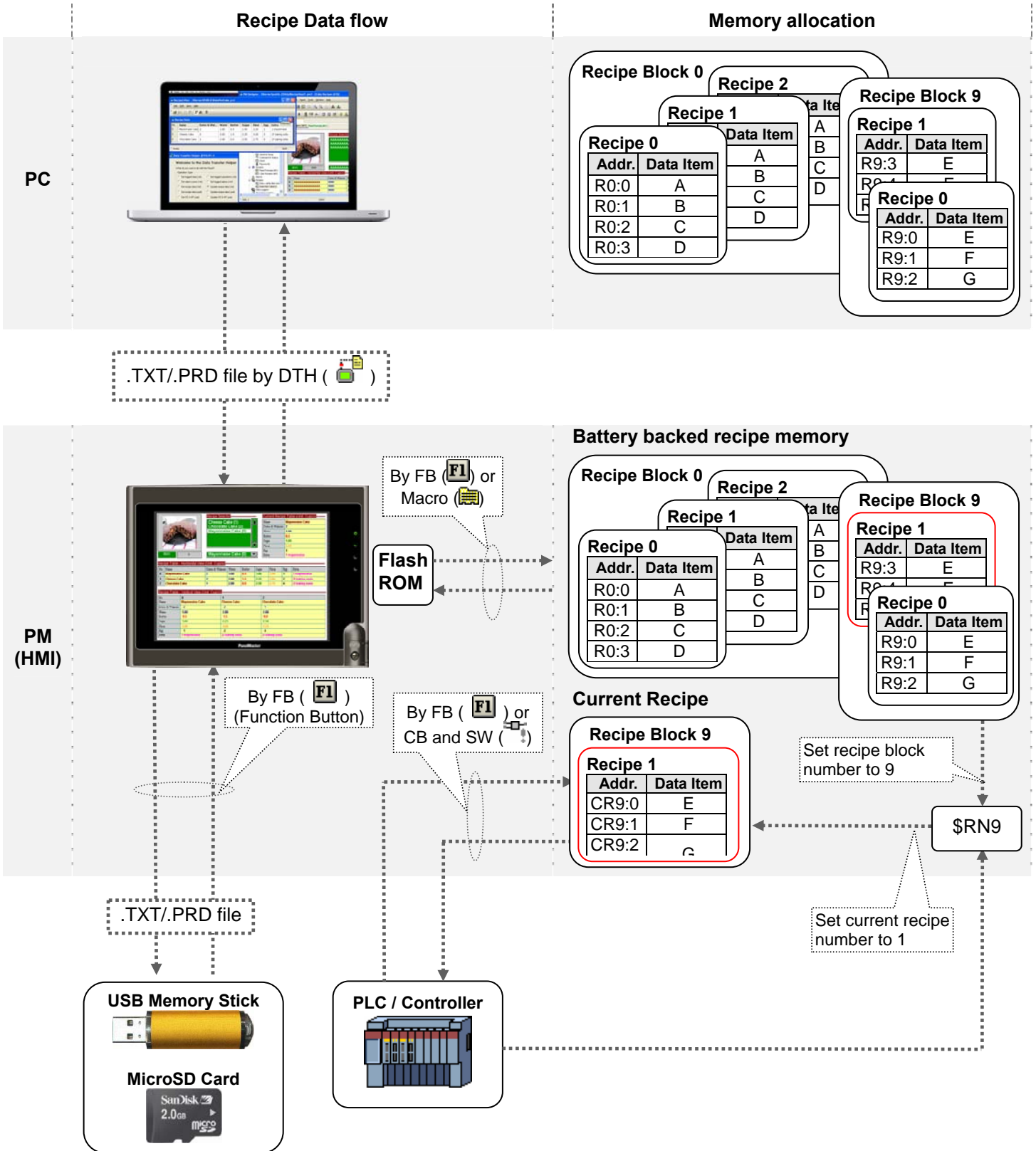
There are two types of recipe data: TXT Data and PRD Data. These recipes can be transferred between PC and HMI or between HMI and USB Memory Stick/Micro SD card directly.

| Recipe Data | Description |
|-------------|--|
| TXT Data | <ul style="list-style-type: none"> Can be created and edited in Microsoft Excel or text editor software (e.g., Notepad) |
| PRD Data | <ul style="list-style-type: none"> Binary Data created in the software Can be edited in RecipeEditor |

10.2. Recipe Data Flow and Memory Allocation

Assume there are two recipe blocks (Recipe Block 0 and Recipe Block 9) in an application. Recipe Block 0 has 3 recipes and each recipe has 4 data items named A,B,C and D. Recipe Block 9 has 2 recipes and each recipe has 3 data items named E,F and G.

The following illustration gives an overview of recipe data flow and recipe memory allocation.



10.3. Working with Recipes

To work with recipes, you may need to do the followings:

■ **Create and set up a recipe block**

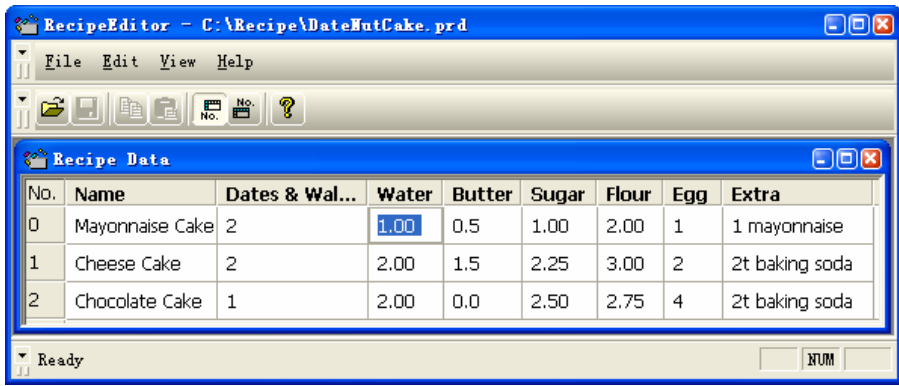
To create a recipe block, you may do one of the followings:

- 1) In the Project Manager tool window, right-click the Recipes node of the concerned panel application and select Add Recipe Block.
- 2) In the menu bar, click Panel to bring up the Panel sub-menu. Click Recipe Block in the Panel sub-menu to bring up the Recipe Block pop-up menu. Select Add in the pop-up menu.

For details about how to set up a recipe block, please see [Section 10.5](#).

■ **View and edit recipe data on PC**

On PC, you can use RecipeEditor to view and edit recipe data saved in *.prd file. The following is an example of RecipeEditor.



▶▶ To run the RecipeEditor, choose Start > Programs > "The software" > RecipeEditor.

▶▶ To edit recipe data directly in the cell, right click the cell and key in the value you want. Note that any value unmatched the predefined format will cause an error when using the recipe at the runtime.

■ **View and edit recipe data on PM (HMI)**

On PM (HMI), you can use a recipe table to view and edit recipe data. For details about how to create and set up a recipe table for an application, please see [Section 10.7](#).

■ **Select a recipe on PM (HMI)**

On PM (HMI), you can select a recipe by using recipe selector. For details about how to create and set up a recipe selector for an application, please see [Section 10.6](#).

■ **Transfer recipes**

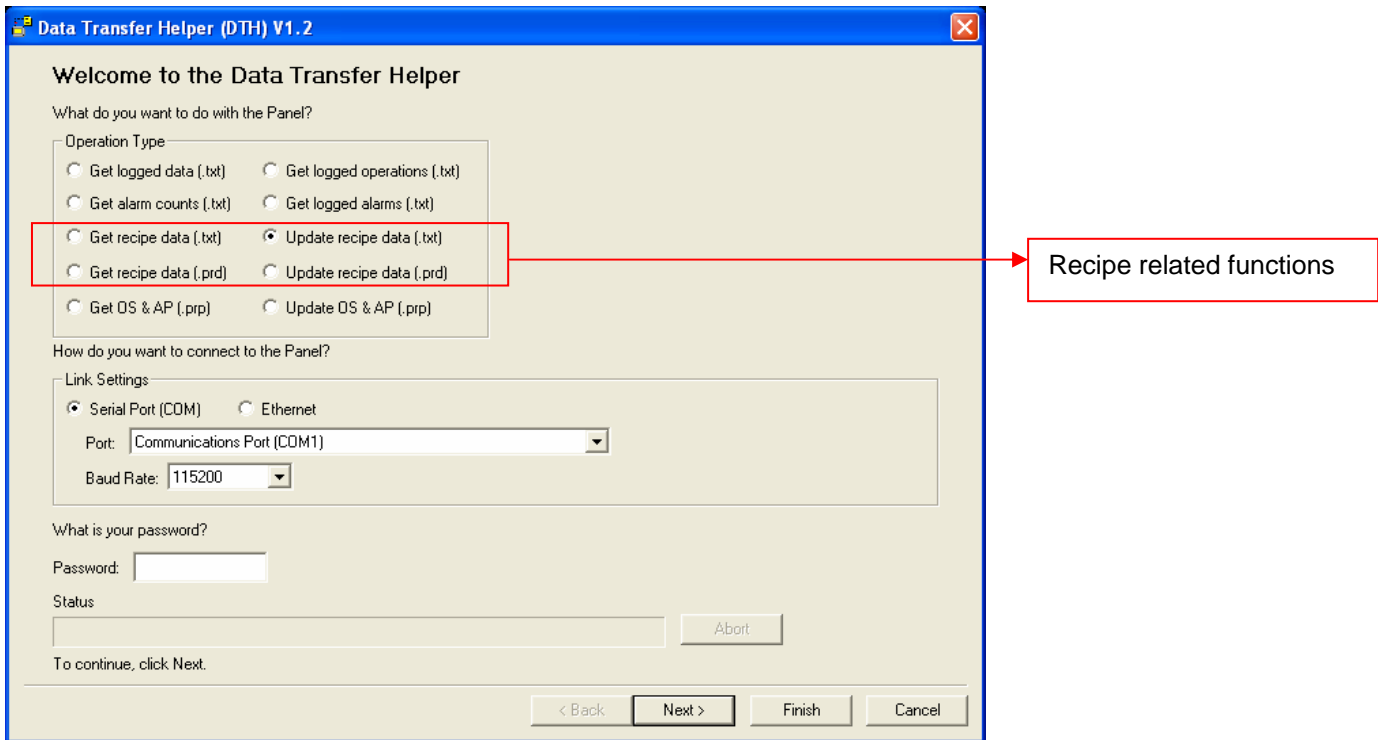
Described in [Section 10.4](#)


10.4. Transferring Recipes

10.4.1. Between PC and PM (HMI)

- By using Data Transfer Helper (DTH) ()

You can use DTH to download recipe data in *.prd/*.txt file from PC to HMI. By DTH, you can also get recipe data from HMI and save the data in *.prd/*.txt file on PC. The following is an example of Data Transfer Helper (DTH) used to get or update recipe data.



To run the DTH, choose Start > Programs > “The software” >  Data Transfer Helper (DTH).

10.4.2. Between PM (HMI) and USB Memory Stick/Micro SD Card

- By using Function Button ()

You can use a function button to save recipe data of the specified recipe block in a .txt\prd file; load recipe data of the specified recipe block from a .txt\prd file. For details, please see [Section 5.4.1 Basic Operations](#) of function buttons

10.4.3. Between battery backed memory and flash ROM

- By using Macro Command ()

You can use RB2ROM to save the data of the specified recipe block to flash ROM, and use ROM2RB to load recipe data from flash ROM. For details, please see [Section 5.4.1 Basic Operations](#) of Macro command

- By using Function Button ()

You can use a function button to save recipe data to flash ROM, and load recipe data from flash ROM. For details, please see [Section 5.4.1 Basic Operations](#) of function buttons

10.4.4. Between PM (HMI) and PLC/Controller

- **By using Command Block and Status Words ()**

You can use command flag setting in Command Block to request the panel to set the current recipe number, read recipe from PLC or write recipe to PLC. You can also receive current recipe block ID or current recipe number by checking the related status word and specifying the word variable in Status Words. For details, please see [Section 3.5.1 Command Block and Status Words](#).

To change current recipe number, the PLC first sets the Parameter Two Register to the desired recipe block and the Parameter One Register to the desired recipe number, then turns on the Set Current Recipe Number (#2) command flag. Also \$RNM (Current Recipe Number Register, m: Recipe Block ID) of the panel can be changed by the PLC.

To update a recipe in the panel, the PLC first sets the Parameter Two Register to the desired recipe block and the Parameter One Register to the desired recipe number, then turns on the Read Recipe From PLC (#3) command flag. The panel reads data in Recipe Block to update the specified recipe in the panel.

To receive a recipe, the PLC first sets the Parameter Two Register to the desired recipe block and Parameter One Register to the desired recipe number, then turns on the Write Recipe To PLC (#4) command flag. The panel sends the specified recipe data to the Recipe Block in PLC.

Note: You do not need to specify the recipe block if the application has only one recipe block.

Note: To make the above operation work, the specified recipe block must exist, or the panel ignores the request. And the specified recipe number in the Parameter One Register must be between 0 and the maximum recipe number - 1. If the Parameter One Register is greater than or equal to the maximum recipe number, the panel ignores the request.

- **By using Function Button ()**

You can use a function button to write the current recipe to controller, or update current recipe by reading the recipe from controller. For details, please see [Section 5.4.1 Basic Operations](#) of function buttons

10.5. Setting up Recipe Blocks

You can set up a recipe block with the Recipe Block dialog box. There are two ways to open the dialog box of a recipe block:

- 1) In the Project Manager window, right-click the node of the desired Recipe Block and select Properties.
- 2) In the menu bar, click Panel to bring up the Panel sub-menu. Click Recipes in the Panel sub-menu to bring up the Recipe Block pop-up menu. Select Properties in the pop-up menu to bring up the recipe block list of the current panel application. Select the recipe block in the list.

The Recipe Block dialog box contains the following two pages:

- **General**

Described in [Section 10.5.1](#).

- **Data Item**

Described in [Section 10.5.2](#).

10.5.1. General Settings

Use the General page to define the general settings for a recipe block. The following is an example of the General page.

Recipe Block

General | Data Item

Name: ID:

Recipe Size: words Number Of Recipes:

Memory Required: words

Write Recipe To PLC
Write Address:

Notification Bit:

Read Recipe From PLC Read Address Identical To Write Address
Read Address:

Notification Bit:

Read/Write Size: words

Reverse the order of the high word and low word of 32-bit data

Recipe Memory

Bit Address Range:

Word Address Range:

Current Recipe









Bit Address Range:

Word Address Range:

Current Recipe Number Register:

OK Cancel Help

The table below describes each property in the General page.

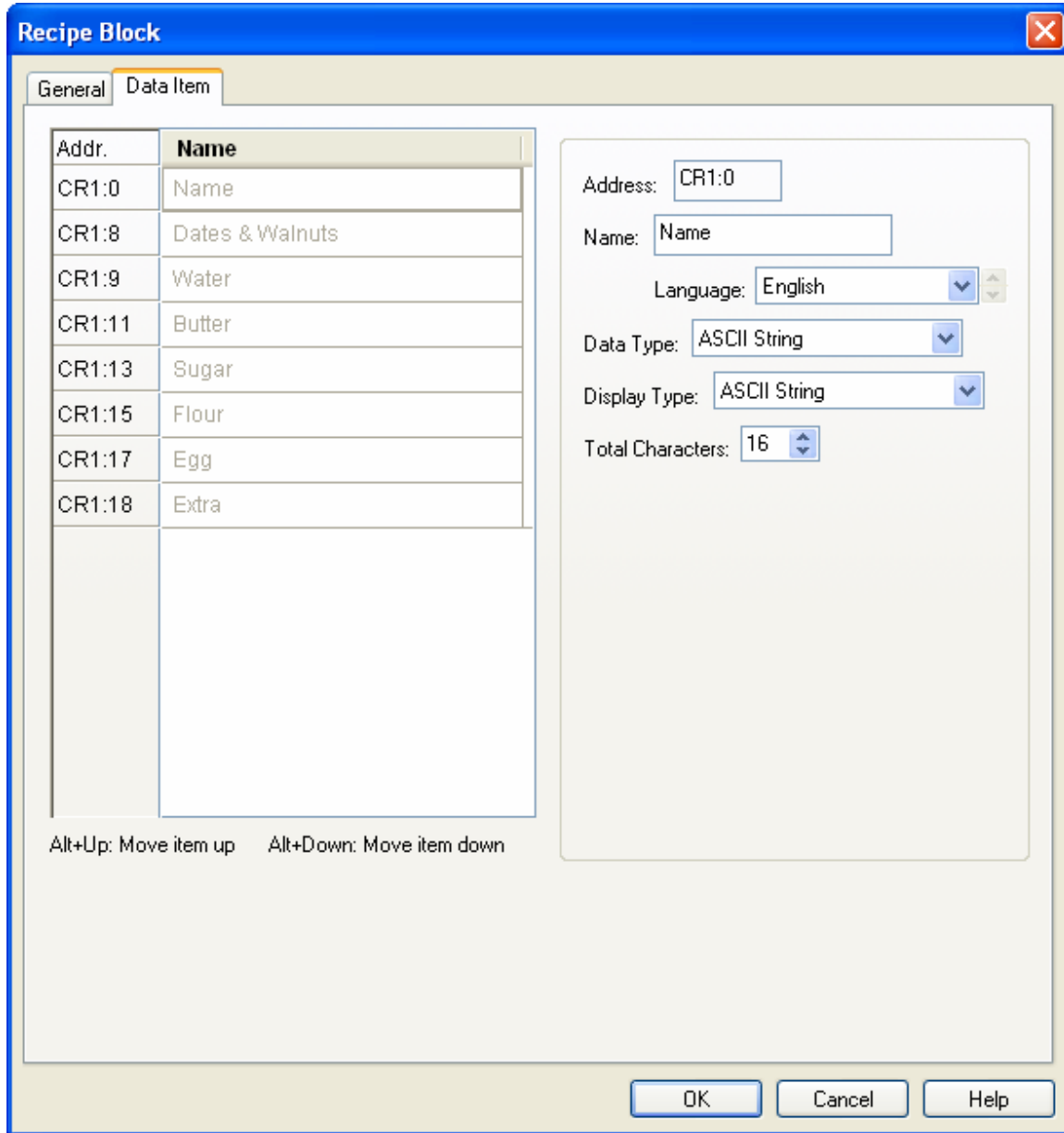
| Property | Description | | | | | | | | | |
|--|--|---|----------------|-------------|-------------------|--------------------|---|--------------------|--------|--|
| Name | The recipe block's name. The maximum length of the name is 48 characters. | | | | | | | | | |
| ID | The recipe block's ID number. Select a number between 0 and 15. The number is unique among all recipe blocks of the panel application. | | | | | | | | | |
| Recipe Size | Specifies the data size that each recipe contains. The unit is word | | | | | | | | | |
| Number of Recipes | Specifies the maximum number of recipes that the recipe memory can hold. | | | | | | | | | |
| Memory Required | The size of the recipe memory. The unit is word. The formula to calculate the size is the size is: $\text{Memory Required} = \text{Recipe Size} * \text{Number of Recipes}$ | | | | | | | | | |
| Write Recipe To PLC | Check the option if you want to write the recipe to PLC | | | | | | | | | |
| Write Address | Available when the Write Recipe To PLC is checked. Specifies the variable that is the starting address of the Recipe Block in your PLC. Click  to enter an address for this field. Click  to select a tag for this field. The size of the recipes to be written is specified in the Read/Write Size field. | | | | | | | | | |
| Notification | Check the option if you want the recipe to set the bit specified in the Bit Field to On when finishing writing recipe to PLC. | | | | | | | | | |
| Bit | Available when the Notification field is checked. Specifies the bit for the operation done notification. Click  to enter an address for this field. Click  to select a tag for this field. | | | | | | | | | |
| Read Recipe From PLC | Check this option if you want to read recipes from PLC. | | | | | | | | | |
| Read Address Identical To Write Address | Specifies that the Read Address is identical to the Write Address. With this item checked, you don't need to specify the Read Address again. This item is available when the option Read Recipe From PLC is checked. | | | | | | | | | |
| Read Address | Available when the Read Recipe From PLC is checked. Specifies the variable representing the starting address of the recipe block in your PLC. Click  to enter an address for this field. Click  to select a tag for this field. The size of the recipes to be read is specified in the Read/Write Size field. | | | | | | | | | |
| Notification | Check the option if you want the recipe to set the bit specified in the Bit Field to On when finishing reading recipe from PLC. | | | | | | | | | |
| Bit | Available when the Notification field is checked. Specifies the bit for the operation done notification. Click  to enter an address for this field. Click  to select a tag for this field. | | | | | | | | | |
| Read/Write Size | The size of recipe for reading and writing. | | | | | | | | | |
| Reverse the order of the high word and low word of 32-bit data | Check this option if the Write Address or the Read Address belongs to a controller that stores data in big-endian byte order and if there are 32-bit data items, such as 32-bit signed integers and 32-bit floating point numbers, defined in the recipe block. | | | | | | | | | |
| Recipe Memory | The address range of the internal memory in the panel that the overall recipe block locates. <table border="1" data-bbox="478 1675 1481 1890"> <thead> <tr> <th>Range Type</th> <th>Address Format</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Bit Address Range</td> <td>\$Rm:n.b b: 0~f</td> <td>Each bit address in the range refers to a bit of a recipe word in specified recipe block.</td> </tr> <tr> <td>Word Address Range</td> <td>\$Rm:n</td> <td>Each word address in the range refers to a recipe word</td> </tr> </tbody> </table> <p>Legend: m = Recipe Block ID; n = The Number of Recipe Word, b = Bit Number;</p> | Range Type | Address Format | Description | Bit Address Range | \$Rm:n.b b: 0~f | Each bit address in the range refers to a bit of a recipe word in specified recipe block. | Word Address Range | \$Rm:n | Each word address in the range refers to a recipe word |
| Range Type | Address Format | Description | | | | | | | | |
| Bit Address Range | \$Rm:n.b b: 0~f | Each bit address in the range refers to a bit of a recipe word in specified recipe block. | | | | | | | | |
| Word Address Range | \$Rm:n | Each word address in the range refers to a recipe word | | | | | | | | |

Continued

| Property | Description | | |
|--|---|---|---|
| Current Recipe | The address range of the internal memory in the panel that the current recipe locates. | | |
| | Range Type | Address Format | Description |
| | Bit Address Range | \$CRm:n.b b: 0~f | Each bit address in the range refers to a bit of a recipe word in the current recipe of the specified recipe block. |
| | Word Address Range | \$CRm:n | Each word address in the range refers to a recipe word in the current recipe of the specified recipe block. |
| Current Recipe Number Register | \$RNm | An internal register of the panel that specifies the current recipe number of the specified recipe block. | |
| Legend: m = Recipe Block ID; n = The Number of Recipe Word, b = Bit Number; | | | |
| Need space in flash ROM to save backup | Check this option if you need the space in flash ROM to save the backup recipes. | | |
| Do not use battery backed RAM | Check this option so the memory of the recipes will be located in ordinary RAM and the recipe memory will be cleared whenever the target panel is powered up. If this option is not selected, the memory of the recipes will be located in the battery backed RAM. The recipe data will not be lost after power down if the battery backed RAM is used for the recipes. | | |

10.5.2. Data Item Settings

Use the Data Item page to define the data items of the data for a recipe. The following is an example of the Data Item page.



The Data Item page contains two parts. The left part is the data item list that shows the address and name of each data item in a row. The right part shows the properties of the selected data item. To select a data item, click the row of that data item in the data item list. The following table describes each property of the data item.

| Property | Description |
|-----------|--|
| Address | You can use the address shown here to refer to the latest value of the data item. |
| Name | Specifies the name of the data item for the language specified in the Language field. |
| Language | Select a language so you can view and edit the name of the data item for that language. |
| Data Type | The data type of the data item. The supported data types include: 16-Bit Unsigned Integer, 32-Bit Unsigned Integer, 16-Bit Signed Integer, 32-Bit Signed Integer, 16-Bit BCD, 32-Bit BCD, 32-Bit Floating Point, ASCII String, and Unicode String. Note that Unicode String is supported for PanelExpress only. |

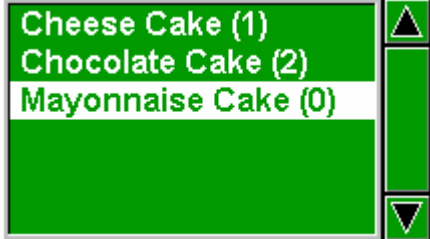

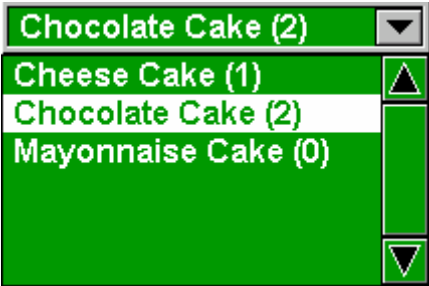
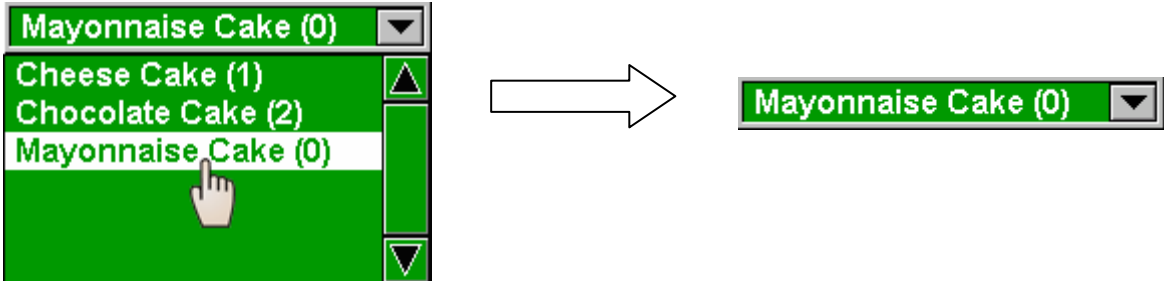
Continued

| Property | Description | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|--|-------------------|-------------------------|-------------------------|---|-------------------------|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------------------|------------|-------------------------|-----------------------|-----------------------|--------------|--------------|----------------|----------------|-----------------------|---|---|----|-------|
| Display Type | <p>The display type for the value of the data item. The following table shows the available display types for each data type.</p> <table border="1"> <thead> <tr> <th>Data Type</th> <th>Available Display Types</th> </tr> </thead> <tbody> <tr> <td>16-Bit Unsigned Integer</td> <td>16-Bit Unsigned Decimal, 16-Bit Hexadecimal, 16-Bit Octal</td> </tr> <tr> <td>32-Bit Unsigned Integer</td> <td>32-Bit Unsigned Decimal, 32-Bit Hexadecimal, 32-Bit Octal</td> </tr> <tr> <td>16-Bit Signed Integer</td> <td>16-Bit Signed Decimal</td> </tr> <tr> <td>32-Bit Signed Integer</td> <td>32-Bit Signed Decimal</td> </tr> <tr> <td>16-Bit BCD</td> <td>16-Bit Unsigned Decimal</td> </tr> <tr> <td>32-Bit BCD</td> <td>32-Bit Unsigned Decimal</td> </tr> <tr> <td>32-Bit Floating Point</td> <td>32-Bit Floating Point</td> </tr> <tr> <td>ASCII String</td> <td>ASCII String</td> </tr> <tr> <td>Unicode String</td> <td>Unicode String</td> </tr> </tbody> </table> | Data Type | Available Display Types | 16-Bit Unsigned Integer | 16-Bit Unsigned Decimal, 16-Bit Hexadecimal, 16-Bit Octal | 32-Bit Unsigned Integer | 32-Bit Unsigned Decimal, 32-Bit Hexadecimal, 32-Bit Octal | 16-Bit Signed Integer | 16-Bit Signed Decimal | 32-Bit Signed Integer | 32-Bit Signed Decimal | 16-Bit BCD | 16-Bit Unsigned Decimal | 32-Bit BCD | 32-Bit Unsigned Decimal | 32-Bit Floating Point | 32-Bit Floating Point | ASCII String | ASCII String | Unicode String | Unicode String | | | | | |
| Data Type | Available Display Types | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16-Bit Unsigned Integer | 16-Bit Unsigned Decimal, 16-Bit Hexadecimal, 16-Bit Octal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32-Bit Unsigned Integer | 32-Bit Unsigned Decimal, 32-Bit Hexadecimal, 32-Bit Octal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16-Bit Signed Integer | 16-Bit Signed Decimal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32-Bit Signed Integer | 32-Bit Signed Decimal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16-Bit BCD | 16-Bit Unsigned Decimal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32-Bit BCD | 32-Bit Unsigned Decimal | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32-Bit Floating Point | 32-Bit Floating Point | | | | | | | | | | | | | | | | | | | | | | | | | |
| ASCII String | ASCII String | | | | | | | | | | | | | | | | | | | | | | | | | |
| Unicode String | Unicode String | | | | | | | | | | | | | | | | | | | | | | | | | |
| Total Digits | Specifies the number of digits to be displayed for the value of the data item. | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fractional Digits | <p>Specifies how to display the fractional part for the value of the data item. When the Display Type is 32-bit Floating Point, this property specifies the number of fractional digits to be displayed. When the Display Type is not 32-bit Floating Point, this property specifies not only the number of fractional digits to be displayed but also the number of least significant digits to be displayed as the fractional part. With this feature, an integer can be shown as a fixed point number.</p> <p>Example:</p> <table border="1"> <thead> <tr> <th>Display Type</th> <th>Total Digits</th> <th>Fractional Digits</th> <th>Sampled Value</th> <th>Displayed Value</th> </tr> </thead> <tbody> <tr> <td>32-bit Floating Point</td> <td>4</td> <td>2</td> <td>12.34</td> <td>12.34</td> </tr> <tr> <td>32-bit Floating Point</td> <td>4</td> <td>2</td> <td>123.4</td> <td>23.40</td> </tr> <tr> <td>16-bit Signed Decimal</td> <td>5</td> <td>2</td> <td>12345</td> <td>123.45</td> </tr> <tr> <td>16-bit Signed Decimal</td> <td>5</td> <td>2</td> <td>-5</td> <td>-0.05</td> </tr> </tbody> </table> | Display Type | Total Digits | Fractional Digits | Sampled Value | Displayed Value | 32-bit Floating Point | 4 | 2 | 12.34 | 12.34 | 32-bit Floating Point | 4 | 2 | 123.4 | 23.40 | 16-bit Signed Decimal | 5 | 2 | 12345 | 123.45 | 16-bit Signed Decimal | 5 | 2 | -5 | -0.05 |
| Display Type | Total Digits | Fractional Digits | Sampled Value | Displayed Value | | | | | | | | | | | | | | | | | | | | | | |
| 32-bit Floating Point | 4 | 2 | 12.34 | 12.34 | | | | | | | | | | | | | | | | | | | | | | |
| 32-bit Floating Point | 4 | 2 | 123.4 | 23.40 | | | | | | | | | | | | | | | | | | | | | | |
| 16-bit Signed Decimal | 5 | 2 | 12345 | 123.45 | | | | | | | | | | | | | | | | | | | | | | |
| 16-bit Signed Decimal | 5 | 2 | -5 | -0.05 | | | | | | | | | | | | | | | | | | | | | | |
| Scaling | <p>Check this option if you want the value of the data item to be displayed in a scaled manner. The following is the scaling formula:</p> $\text{DisplayedValue} = \text{SampledValue} * \text{Gain} + \text{Offset}$ <p>Note: The <i>Gain</i> and <i>Offset</i> are 32-bit floating point numbers. They have at most 6 significant digits. The rounding and truncation errors may happen.</p> | | | | | | | | | | | | | | | | | | | | | | | | | |
| Gain | Available when the Scaling option is checked. Specifies the <i>Gain</i> used in the scaling formula. | | | | | | | | | | | | | | | | | | | | | | | | | |
| Offset | Available when the Scaling option is checked. Specifies the <i>Offset</i> used in the scaling formula. | | | | | | | | | | | | | | | | | | | | | | | | | |
| Range Check | Check this option if you want the data item to verify the entered value according to the specified minimum and maximum. If the entered value is not within the allowable range, the entered value will not be output. | | | | | | | | | | | | | | | | | | | | | | | | | |
| Min | Specifies the minimum value. | | | | | | | | | | | | | | | | | | | | | | | | | |
| Max | Specifies the maximum value. | | | | | | | | | | | | | | | | | | | | | | | | | |

10.6. Selecting a Recipe Using Recipe Selectors

10.6.1. Basic Operations

A recipe selector can be configured to perform as one of the following types of controls:

| Type | Description |
|----------------|--|
| List | <p>The recipe selector is a list box. It displays a list of index strings of the recipes in the specified recipe block. The index string's format is recipe name(recipe number). One recipe index string per line.</p> <p>The index string of the current recipe is highlighted. If the desired recipe is not in the view, you can scroll the list by the scroll bar attached to the right side of the list box. When you select a desired recipe by touching its index string, the recipe selector writes the recipe number of the selected recipe to current recipe number register.</p>  |
| Drop-down List | <p>The recipe selector is a drop-down list. It displays the index string of the current recipe and a button with the down arrow symbol as shown in the following example.</p>  <p>When the button is touched, the recipe selector displays a list box beneath itself as shown in the following example.</p>  <p>The list box lists the index strings of all recipes of the recipe block. One index string per line. The index string of the current recipe is highlighted. If the desired recipe is not in the view, you can use the scroll bar attached to the right side of the list to scroll the index string. When you select a desired recipe by touching its index string, the recipe selector writes the recipe number of the selected recipe to current recipe number register and closes the list box.</p>  <p>If you want to cancel the operation when the list box is showing, touch anywhere other than the index string in the list box.</p> |

10.6.2. Operation Options

The following operation option can be added to a recipe selector. Select and set up the option in the recipe selector property sheet.

| Options | Description |
|--------------------|--|
| Visibility Control | You can show or hide a recipe selector by a specified bit or the current user level. Select and set up this option in the Visibility page. |

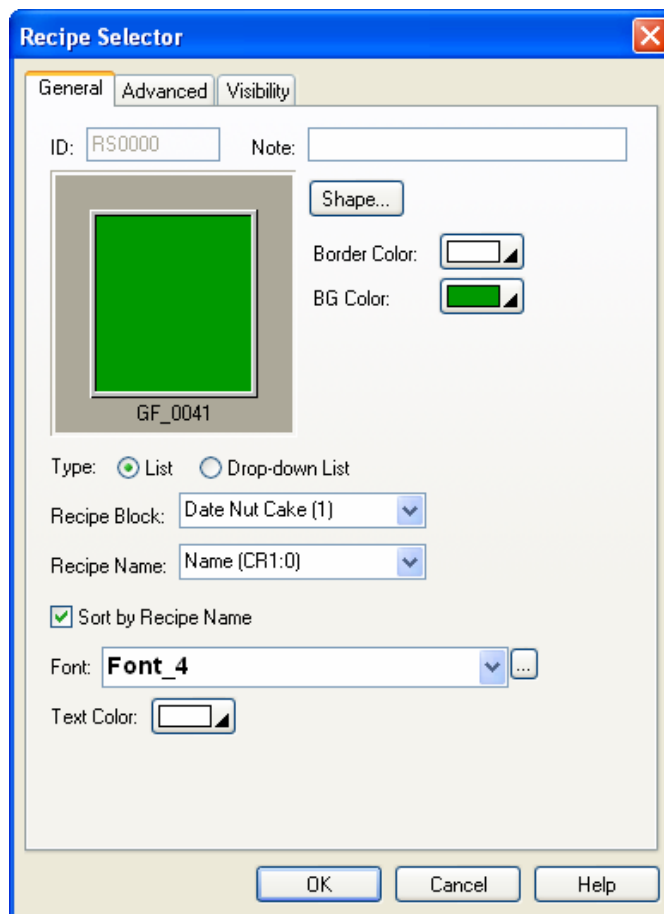
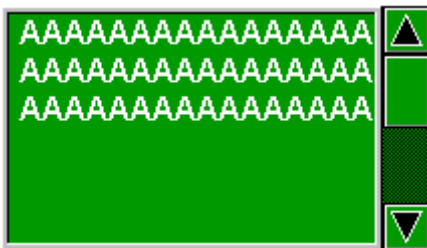
10.6.3. Settings

You can complete all the settings of a recipe selector in the Recipe Selector property sheet. This sheet contains the following three pages.

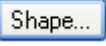
- **General**
Described in [Section 10.6.4.](#)
- **Advanced**
Described in [Section 4.4.5.](#)
- **Visibility**
Described in [Section 4.4.6.](#)

10.6.4. General Settings

This section describes how to define the general settings for a recipe selector. The following is an example of the General page.



The following table describes each property in the General page.

| Property | Description | | | | | | |
|---------------------|---|------|-------------|------|---|----------------|---|
| ID | The object's identifier. It is generated when the object is created and is unchangeable. The identifier is unique within the screen where the object is on. The format of the ID's for the recipe selectors is RSnnnn. | | | | | | |
| Note | You can type a note for the object. | | | | | | |
| Shape settings | For details about the following properties, Section 4.3.4 Setting up the Shape of an Object.  , Border Color, BG Color | | | | | | |
| Type | Select one of the following types for the recipe selector: <table border="1" data-bbox="534 562 1492 719"> <thead> <tr> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>List</td> <td>The list box is displayed at all times.</td> </tr> <tr> <td>Drop-down List</td> <td>The list box is not displayed unless the user clicks the down arrow icon next to the static-text control.</td> </tr> </tbody> </table> | Type | Description | List | The list box is displayed at all times. | Drop-down List | The list box is not displayed unless the user clicks the down arrow icon next to the static-text control. |
| Type | Description | | | | | | |
| List | The list box is displayed at all times. | | | | | | |
| Drop-down List | The list box is not displayed unless the user clicks the down arrow icon next to the static-text control. | | | | | | |
| Recipe Block | Select the recipe block whose recipe is to be selected by the Recipe Selector object. | | | | | | |
| Recipe Name | Select a data item from the list as the recipe name. You can select any data items with ASCII String data type as the name of the recipe from the drop down list. | | | | | | |
| Sort by Recipe Name | Check this option to automatically sort all recipe names added to the list box. | | | | | | |
| Font | The font of the displayed string. | | | | | | |
| Text Color | The color of the displayed string. | | | | | | |

10.7. Displaying and Modifying Recipe Data Using Recipe Tables

10.7.1. Basic Operations

There are three types of recipe tables.

| Type | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|--|--------------------|------------------------|-----------------|-------|--------|------------------------|--------------------|-----------------------|-----------------|------|------------------------|------|-------|------|-------|--------------|--------|--------------|-----|--------------------|-------|------|------|------|-------|------|----------------|------|-----------------------|---|------|-----|-------|--------------|----------------|----------------|
| Horizontal View | <p>Displays the recipes row by row and recipe data items column by column.</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Name</th> <th>Dates & Walnuts</th> <th>Water</th> <th>Butter</th> <th>Sugar</th> <th>Flour</th> <th>Egg</th> <th>Extra</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Mayonnaise Cake</td> <td>2</td> <td>1.00</td> <td>0.5</td> <td>1.00</td> <td>2.00</td> <td>1</td> <td>1 mayonnaise</td> </tr> <tr> <td>1</td> <td>Cheese Cake</td> <td>2</td> <td>2.00</td> <td>1.5</td> <td>2.25</td> <td>3.00</td> <td>2</td> <td>2t baking soda</td> </tr> <tr> <td>2</td> <td>Chocolate Cake</td> <td>1</td> <td>2.00</td> <td>0.0</td> <td>2.50</td> <td>2.75</td> <td>4</td> <td>2t baking soda</td> </tr> </tbody> </table> <p>The above is an example of the recipe table with horizontal view. The first row displays the data item name of each column. The other rows display one recipe per row. The first column displays recipe number. You can create scroll button groups or scroll bars to scroll the contents.</p> | No. | Name | Dates & Walnuts | Water | Butter | Sugar | Flour | Egg | Extra | 0 | Mayonnaise Cake | 2 | 1.00 | 0.5 | 1.00 | 2.00 | 1 | 1 mayonnaise | 1 | Cheese Cake | 2 | 2.00 | 1.5 | 2.25 | 3.00 | 2 | 2t baking soda | 2 | Chocolate Cake | 1 | 2.00 | 0.0 | 2.50 | 2.75 | 4 | 2t baking soda |
| No. | Name | Dates & Walnuts | Water | Butter | Sugar | Flour | Egg | Extra | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | Mayonnaise Cake | 2 | 1.00 | 0.5 | 1.00 | 2.00 | 1 | 1 mayonnaise | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Cheese Cake | 2 | 2.00 | 1.5 | 2.25 | 3.00 | 2 | 2t baking soda | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Chocolate Cake | 1 | 2.00 | 0.0 | 2.50 | 2.75 | 4 | 2t baking soda | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vertical View | <p>Displays the recipes column by column and recipe data items row by row.</p> <table border="1"> <thead> <tr> <th>No.</th> <th>0</th> <th>1</th> <th>2</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Mayonnaise Cake</td> <td>Cheese Cake</td> <td>Chocolate Cake</td> </tr> <tr> <td>Dates & Walnuts</td> <td>2</td> <td>2</td> <td>1</td> </tr> <tr> <td>Water</td> <td>1.00</td> <td>2.00</td> <td>2.00</td> </tr> <tr> <td>Butter</td> <td>0.5</td> <td>1.5</td> <td>0.0</td> </tr> <tr> <td>Sugar</td> <td>1.00</td> <td>2.25</td> <td>2.50</td> </tr> <tr> <td>Flour</td> <td>2.00</td> <td>3.00</td> <td>2.75</td> </tr> <tr> <td>Egg</td> <td>1</td> <td>2</td> <td>4</td> </tr> <tr> <td>Extra</td> <td>1 mayonnaise</td> <td>2t baking soda</td> <td>2t baking soda</td> </tr> </tbody> </table> <p>The above is an example of the recipe table with vertical view. The first column displays the data item name of each row. The other columns display one recipe per column. The first row displays the recipe number. You can create scroll button groups or scroll bars to scroll the contents.</p> | No. | 0 | 1 | 2 | Name | Mayonnaise Cake | Cheese Cake | Chocolate Cake | Dates & Walnuts | 2 | 2 | 1 | Water | 1.00 | 2.00 | 2.00 | Butter | 0.5 | 1.5 | 0.0 | Sugar | 1.00 | 2.25 | 2.50 | Flour | 2.00 | 3.00 | 2.75 | Egg | 1 | 2 | 4 | Extra | 1 mayonnaise | 2t baking soda | 2t baking soda |
| No. | 0 | 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name | Mayonnaise Cake | Cheese Cake | Chocolate Cake | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dates & Walnuts | 2 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Water | 1.00 | 2.00 | 2.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Butter | 0.5 | 1.5 | 0.0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sugar | 1.00 | 2.25 | 2.50 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flour | 2.00 | 3.00 | 2.75 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Egg | 1 | 2 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Extra | 1 mayonnaise | 2t baking soda | 2t baking soda | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Current Recipe | <p>Displays the recipe data items of the current recipe row by row.</p> <table border="1"> <tbody> <tr> <td>Name</td> <td>Mayonnaise Cake</td> </tr> <tr> <td>Dates & Walnuts</td> <td>2</td> </tr> <tr> <td>Water</td> <td>1.00</td> </tr> <tr> <td>Butter</td> <td>0.5</td> </tr> <tr> <td>Sugar</td> <td>1.00</td> </tr> <tr> <td>Flour</td> <td>2.00</td> </tr> <tr> <td>Egg</td> <td>1</td> </tr> <tr> <td>Extra</td> <td>1 mayonnaise</td> </tr> </tbody> </table> <p>The above is an example of the current recipe. The first column displays the data item name of each row. The other column displays the current recipe. You can create scroll button groups or scroll bars to scroll the contents.</p> | Name | Mayonnaise Cake | Dates & Walnuts | 2 | Water | 1.00 | Butter | 0.5 | Sugar | 1.00 | Flour | 2.00 | Egg | 1 | Extra | 1 mayonnaise | | | | | | | | | | | | | | | | | | | | |
| Name | Mayonnaise Cake | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Dates & Walnuts | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Water | 1.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Butter | 0.5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sugar | 1.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Flour | 2.00 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Egg | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Extra | 1 mayonnaise | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

10.7.2. Operation Options

The following operation option can be added to a recipe table. Select and set up the option in the recipe table property sheet.

| Options | Description |
|--------------------|---|
| Visibility Control | You can show or hide a recipe table by a specified bit or the current user level. Select and set up this option in the Visibility page. |

10.7.3. Settings

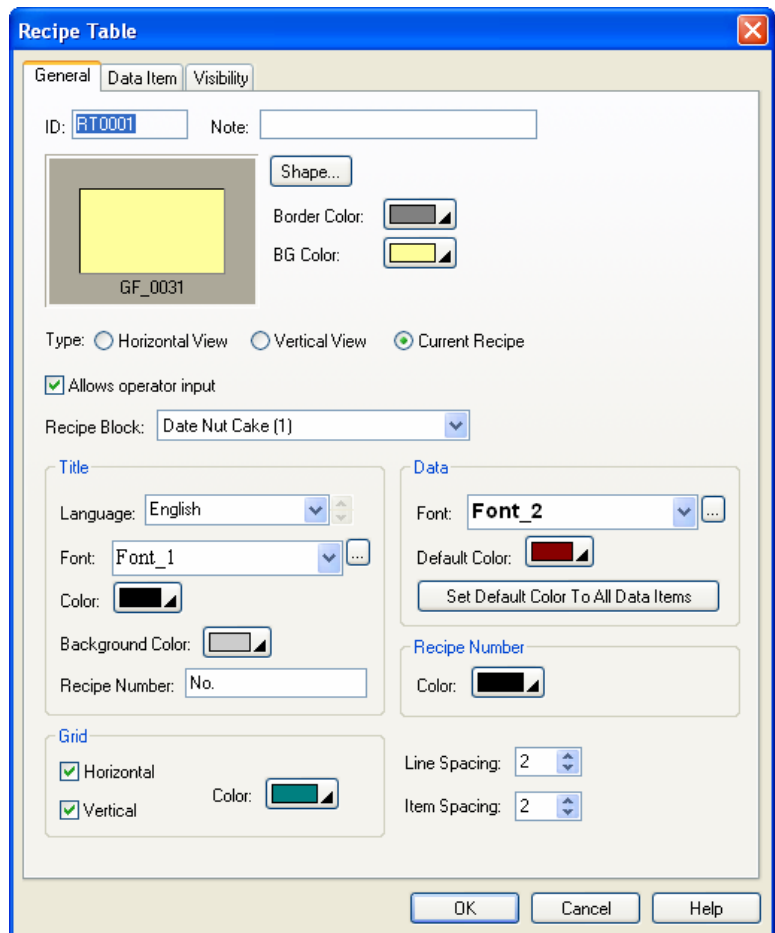
You can complete all the settings of a recipe table in the Recipe Table property sheet. This sheet contains the following three pages.

- **General**
Described in [Section 10.7.4.](#)
- **Data Item**
Described in [Section 10.7.5.](#)
- **Visibility**
Described in [Section 4.4.6.](#)

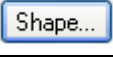
10.7.4. General Settings

This section describes how to define the general settings for a recipe table. The following is an example of the General page.

| | |
|-----------------|------------------|
| Name | AAAAAAAAAAAAAAAA |
| Dates & Walnuts | 9999 |
| Water | -99.99 |
| Butter | -999.9 |
| Sugar | -99.99 |
| Flour | -99.99 |
| Egg | 9999 |
| Extra | AAAAAAAAAAAAAAAA |

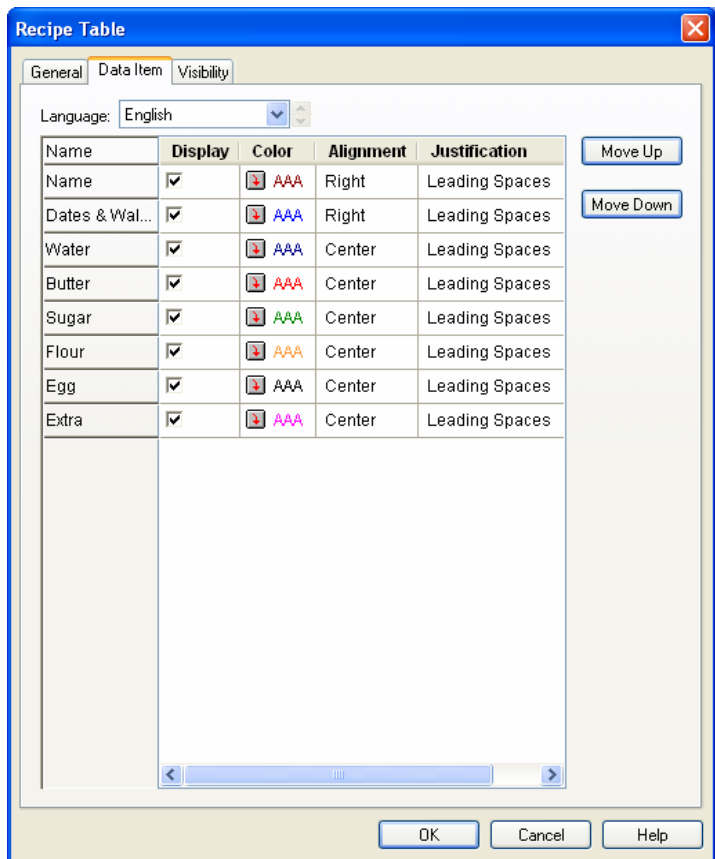


The following table describes each property in the General page.

| Property | | Description | | | | | | | | |
|-----------------------|---|---|------|-------------|-----------------|---|---------------|---|----------------|--|
| ID | | The object's identifier. It is generated when the object is created and is unchangeable. The identifier is unique within the screen where the object is on. The format of the ID's for the recipe tables is RTnnnn. | | | | | | | | |
| Note | | You can type a note for the object. | | | | | | | | |
| Shape settings | | For details about the following properties, Section 4.3.4 Setting up the Shape of an Object.  , Border Color, BG Color | | | | | | | | |
| Type | | Select one of the following types for the recipe table: <table border="1" data-bbox="534 548 1460 806"> <thead> <tr> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Horizontal View</td> <td>Displays the recipes row by row and recipe data items column by column.</td> </tr> <tr> <td>Vertical View</td> <td>Displays the recipes column by column and recipe data items row by row.</td> </tr> <tr> <td>Current Recipe</td> <td>Displays the recipe data items of the current recipe row by row.</td> </tr> </tbody> </table> | Type | Description | Horizontal View | Displays the recipes row by row and recipe data items column by column. | Vertical View | Displays the recipes column by column and recipe data items row by row. | Current Recipe | Displays the recipe data items of the current recipe row by row. |
| Type | Description | | | | | | | | | |
| Horizontal View | Displays the recipes row by row and recipe data items column by column. | | | | | | | | | |
| Vertical View | Displays the recipes column by column and recipe data items row by row. | | | | | | | | | |
| Current Recipe | Displays the recipe data items of the current recipe row by row. | | | | | | | | | |
| Allows operator input | | Check this option if you allow the operator to update the value of the recipe data item. | | | | | | | | |
| Recipe Block | | Select the recipe whose collected data is to be displayed by the object. | | | | | | | | |
| Title | Language | Select a language so you can view and edit the settings of the title row for that language. | | | | | | | | |
| | Font | Select a font for the title text. | | | | | | | | |
| | Color | Select a color for the title text. | | | | | | | | |
| | Background Color | Select a color for the background of the title row. | | | | | | | | |
| | Recipe Number | Specifies the title for the recipe number column. | | | | | | | | |
| Grid | Vertical | Check this option if you want the object to have vertical grids. | | | | | | | | |
| | Horizontal | Check this option if you want the object to have horizontal grids. | | | | | | | | |
| | Color | Select a color for the grids. | | | | | | | | |
| Data | Font | Select a font for displaying data. | | | | | | | | |
| | Default Color | Select a color as the default color for displaying data. | | | | | | | | |
| | Set Default Color to All Data Items | Click this button to set the colors of all the data items to the Default Color. | | | | | | | | |
| Recipe Number | Color | Select a color for the recipe number. | | | | | | | | |
| Line Spacing | | Specifies the extra space in pixels for two adjacent rows in the table. | | | | | | | | |
| Item Spacing | | Specifies the extra space for every column in the table. | | | | | | | | |

10.7.5. Data Item Settings

This section describes how to define the display format for the values of each data item. The following is an example of the Data Item page.



The following table describes each property in the Data Item page.

| Property | | Description | | | | | | | |
|------------------------------|---|--|--------|-------------|---------------|--|---------------|--------------------------|----------------|
| Language | | Select a language so you can view and edit the settings for that language. | | | | | | | |
| Row #n of the property table | Name | The name of data item #n. The data item names are defined in the Data Item page of the Data Logger dialog box. | | | | | | | |
| | Display | Check this option if you want the object to display data item #n. | | | | | | | |
| | Color | Select a color for displaying data item #n. | | | | | | | |
| | Alignment | The alignment for displaying data item #n. There are three types of alignment: Left, Center, and Right. | | | | | | | |
| | Justification | The justification for displaying data item #n. There are three types of justification: <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Zero Suppress</td> <td>The leading digits will not display when they are 0.</td> </tr> <tr> <td>Leading Zeros</td> <td>All digits will display.</td> </tr> <tr> <td>Leading Spaces</td> <td>The leading digits will display as blank character when they are 0.</td> </tr> </tbody> </table> | Option | Description | Zero Suppress | The leading digits will not display when they are 0. | Leading Zeros | All digits will display. | Leading Spaces |
| Option | Description | | | | | | | | |
| Zero Suppress | The leading digits will not display when they are 0. | | | | | | | | |
| Leading Zeros | All digits will display. | | | | | | | | |
| Leading Spaces | The leading digits will display as blank character when they are 0. | | | | | | | | |
| Move Up | | Click the button to move the selected data item before the previous data item. The Move Up button will help you to reorder the display sequence of the data items It will not be available when multiple rows are selected or no row is selected. | | | | | | | |
| Move Down | | Click the button to move the selected data item after the next data item. The Move Down button will help you to reorder the display sequence of the data items It will not be available when multiple rows are selected or no row is selected. | | | | | | | |