

New Features for Enhanced Recipes

The new features for enhanced recipes provide increased convenience.

Y Importing/exporting enhanced recipes

In addition to element values, you can now import and export element labels and recipe labels. You can also import enhanced recipes to GP-Pro EX in CSV files.

Y Mapped address

You can now edit element values in the device/PLC.

Y Send options

When transferring an enhanced recipe to a device/PLC, you can write the enhanced recipe's label and ID to the defined address. This feature is useful for confirming the recipe that you transferred.

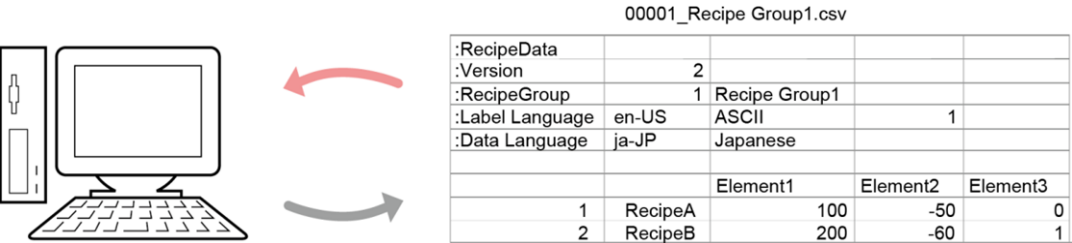
Y Enhanced recipe search

Use system variables to search enhanced recipes.

Settings Menu

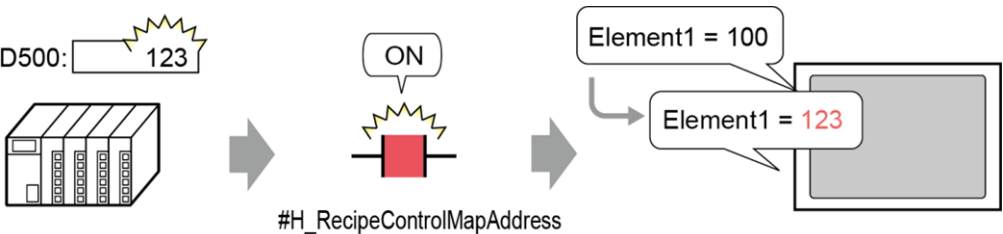
Editing enhanced recipe data with CSV files

Export the GP-Pro EX enhanced recipe settings to a CSV file, which you can edit in a spreadsheet application.



Editing Specific Element Values

Reflect changes in element values from mapped addresses.



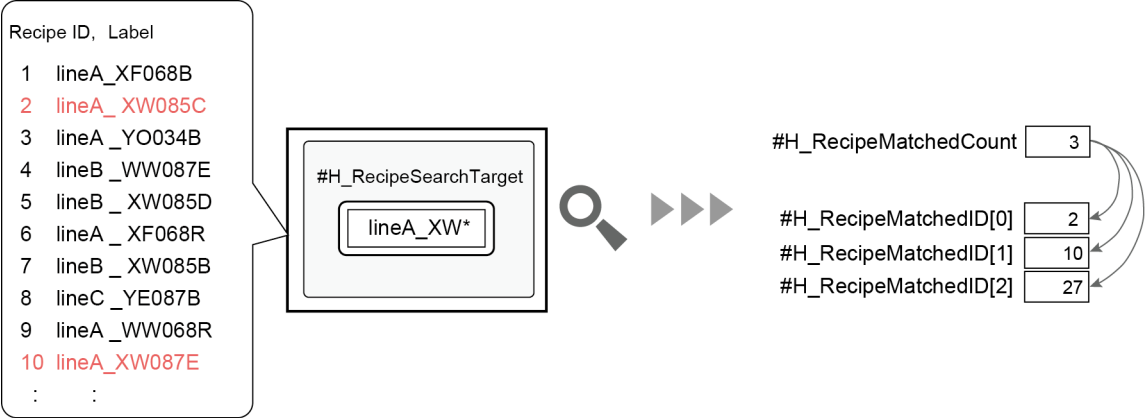
Confirming Which Recipe Data Is Transferred

While transferring an enhanced recipe to the connected device, write the recipe ID and recipe label of the transferred enhanced recipe to the specified address.



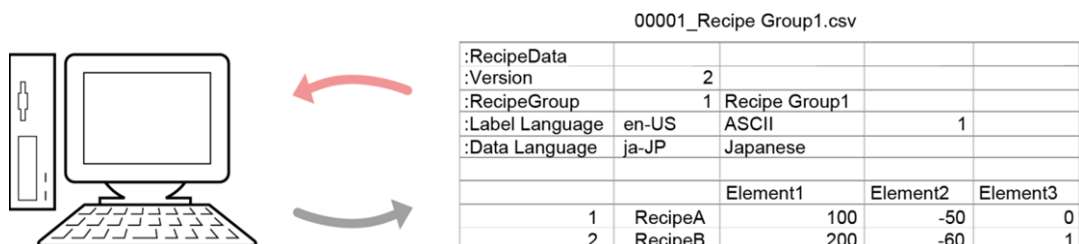
Searching for Enhanced Recipes Using System Variables

Using system variables, you can specify search conditions, run searches, and confirm search results.



Editing enhanced recipe data with CSV files

You can export your GP-Pro EX enhanced recipe settings to a CSV file, where you can edit enhanced recipe settings such as change element values and labels, and add and delete recipes.



Operating Procedure

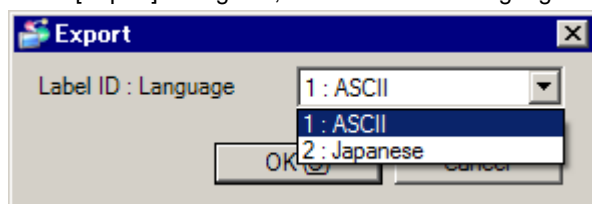
The following describes the procedure to export Recipe Group 1, edit the values, and add a recipe in the CSV file. The [Language] setting for Recipe Group 1 is "Japanese", and the [Select label language] settings are "ASCII" and "Japanese".

| | | | | | | |
|----------|---------|--|---------|---------------------|------------------------|------------------------------|
| Recipe | | | | Add | Remove | Send Options |
| RecipeID | Label 1 | | Label 2 | | Security Level | |
| 1 | RecipeA | | レシピA | | | |
| 2 | RecipeB | | レシピB | | | |
| | | | | | | |

| | | | | | | | |
|---------|----------|--|---------|--|---------|------------------------|-----------------------------|
| Element | | | | | | Remove | Show All Co |
| | Label 1 | | Label 2 | | RecipeA | RecipeB | |
| 1 | Element1 | | 要素 1 | | 100 | 200 | |
| 2 | Element2 | | 要素 2 | | -50 | -60 | |
| 3 | Element3 | | 要素 3 | | OFF | ON | |

Step 1: Exporting

- In the [Common Settings] menu, select [Enhanced Recipe Settings] - [Enhanced Recipe Group List], select the enhanced recipe group (Recipe Group1), and then click [Export].
- In the [Export] dialog box, select the label language for the CSV file.



MEMO

- When the enhanced recipe group has just one label, the [Export] dialog box for selecting the [Label ID: Language] does not display.
- The CSV file is named automatically based on the enhanced recipe ID and name.

Step 2: Editing the CSV file

- Using a spreadsheet application, open the CSV file exported in Step 1 (00001_Recipe Group1.csv). (Change Element 2 of Recipe A to -10. Change Element 1 of Recipe B to 300, and add Recipe C)

| | | | | |
|-----------------|-----------|----------|---------------|----------|
| :RecipeData | | | | |
| :Version | | 2 | | |
| :RecipeGroup | | 1 | Recipe Group1 | |
| :Label Language | en-US | ASCII | | 1 |
| :Data Language | ja-JP | Japanese | | |
| | | | | |
| | | Element1 | Element2 | Element3 |
| | 1 RecipeA | 100 | -10 | 0 |
| | 2 RecipeB | 300 | -60 | 1 |
| | 3 RecipeC | 500 | -70 | 1 |

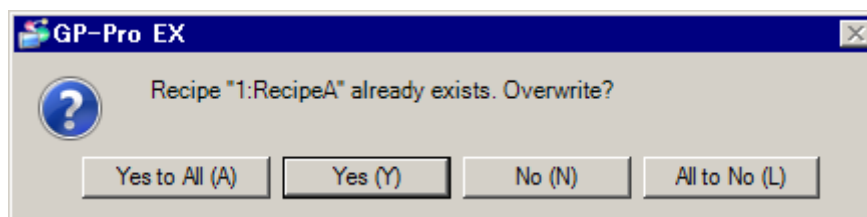
- Overwrite and save the CSV file.

NOTE

Y Save the CSV file in comma-delimited format.

Step 3: Importing

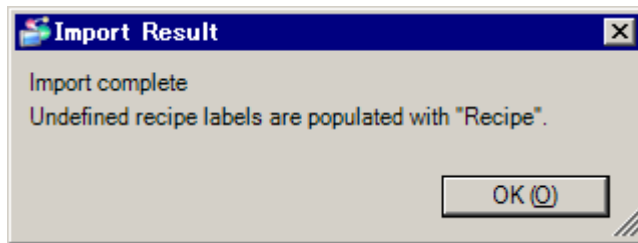
- In [Common Settings] - [Enhanced Recipe Settings], select the recipe group (Recipe Group1), and then click [Import].
- In the file selection dialog box, select the CSV file that you edited in Step 2 (00001_Recipe Group1.csv), and click [Open].
- On the confirm overwrite dialog box, select "Yes". Next, a dialog box corresponding to the recipe number is displayed. Select [Yes] or [Yes to All].



NOTE

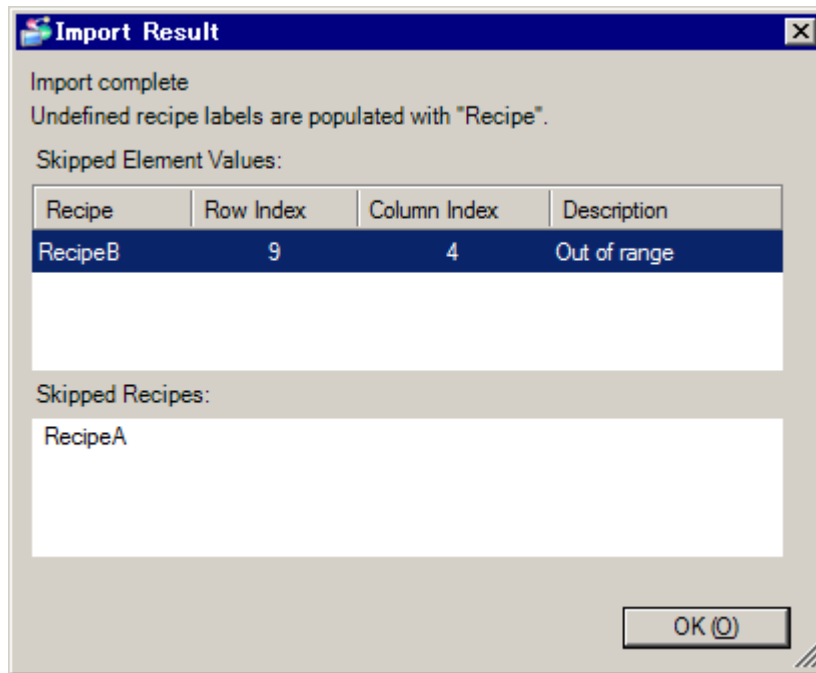
- The overwrite confirmation dialog box is displayed whenever the CSV file contains a recipe ID that exists in the recipe group.
- The overwrite confirmation dialog box is displayed for each pre-existing recipe ID. To overwrite the recipe and all subsequent recipes, click [Yes to All].

- When all recipes are imported, from the [Import Results] dialog box, click [OK].



NOTE

- Y The [Import Results] dialog box shows the items that were not imported (such as enhanced recipes that you did not overwrite, and element values out of range). It also shows details such as recipe labels and column numbers in the CSV file.



- When importing is complete, open the enhanced recipe group (Recipe Group 1), and configure any recipes you may have added in Step 2 above.

Recipe

[Add](#)

[Remove](#)

[Send Options](#)

| RecipeID | Label 1 | Label 2 | Security Level | |
|----------|---------|---------|----------------|--|
| 1 | RecipeA | レシピA | 0 | |
| 2 | RecipeB | レシピB | 0 | |
| 3 | RecipeC | Recipe | 0 | |

Element

[Remove](#)

[Show All Columns >>](#)

| | Label 1 | Label 2 | RecipeA | RecipeB | RecipeC |
|---|----------|---------|---------|---------|---------|
| 1 | Element1 | 要素 1 | 100 | 300 | 500 |
| 2 | Element2 | 要素 2 | -10 | -60 | -70 |
| 3 | Element3 | 要素 3 | OFF | ON | ON |

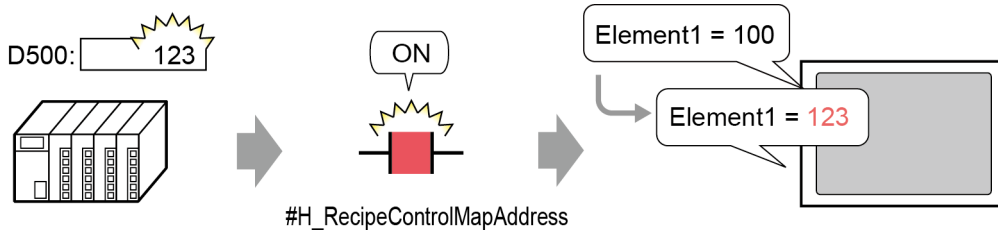
NOTE

- Y If you have two or more label types, "Recipe" is automatically assigned to undefined recipe labels.

Editing Specific Element Values (Mapped Address)

You can edit specific element values on the device/PLC.

Assign a mapped address to the recipe element that you want to edit, and then change the value of the mapped address to change the element value. To set the mapped address value as the recipe element value, you need to save it using the system variable #H_RecipeControlMapAddress.



Setup Procedure

Assign a mapped address to Element 1 of Recipe Group 1.

| | Data type | Address | # | Recipe A (ID: 1) | Recipe B (ID: 2) |
|-----------|------------|----------|---|---------------------|---------------------|
| Element 1 | 16 Bit Dec | D0000100 | 1 | 100 | 200 |
| Element 2 | 16 Bit Dec | D0000110 | 1 | -50 | -60 |
| Element 3 | Bit | D0000120 | 1 | OFF | ON |

1. Set the mapped address in the element.

In [Common Settings] - [Enhanced Recipe Settings], open the recipe group (Recipe Group 1), select the [Map] check box, and set the [Mapped Address] (D500).

| Recipe | | | | Add | Remove | Send Options | | | | | |
|-----------|----------|---------------------|----------------|---------------------|-------------------------------------|------------------------------|------------------------|--|-------------------------------------|----------------|---------|
| Recipe ID | Label 1 | Security Level | | | | | | | | | |
| 1 | RecipeA | 0 | | | | | | | | | |
| 2 | RecipeB | 0 | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| Element | | | | | | | Remove | Show Only Label and Value << | | | |
| | Label 1 | Data Type | Address | Units | Editable | Input cor | From | To | Map | Mapped Address | RecipeA |
| 1 | Element1 | 16 Bit Dec Unsigned | [PLC1]D0000100 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input checked="" type="checkbox"/> | [PLC1]D0000500 | 100 |
| 2 | Element2 | 16 Bit Dec Signed | [PLC1]D0000110 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input type="checkbox"/> | | -10 |
| 3 | Element3 | Bit | [PLC1]X00120 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input type="checkbox"/> | | OFF |

2. Set up a function to turn ON the system variable #H_RecipeControlMapAddress.
3. Set up a function to change the value of the mapped address (D500).
4. Set up additional enhanced recipe functions to perform the following:
 Select which enhanced recipe to work with (system variables #H_RecipeGroupID, #H_RecipeID).
 Confirm the processing state of the enhanced recipe (system variable #H_Processing).
 Show the result of changes to the mapped address (system variable #H_RecipeResultTransfer)

Operating Procedure

Change the mapped address value (D500), and reflect it in Element 1 of Recipe A.

1. Select the recipe to edit. (Set system variable #H_RecipeGroupID to "1" and #H_RecipeID to "1")

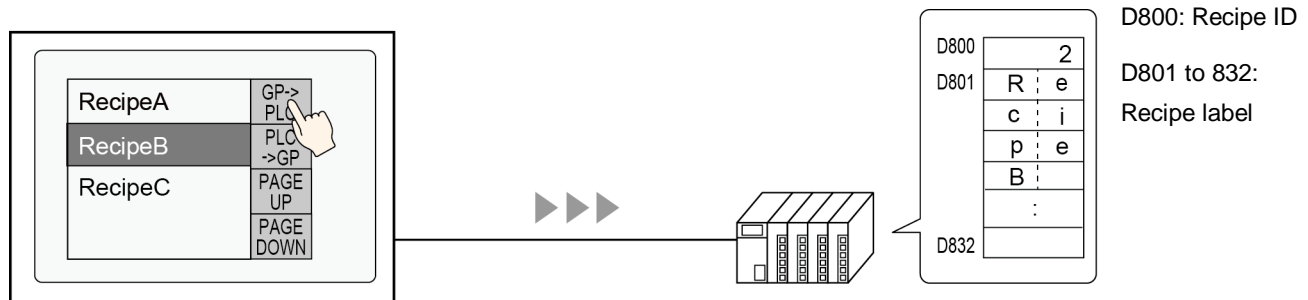
NOTE

Y The mapped address stores the element value of the recipe being edited.

2. Change the value of the mapped address (D500). (Change the value from 100 to 123.)
3. To save, turn ON the system variable #H_RecipeControlMapAddress. Confirm the save status with #H_Processing. If the save fails, confirm the cause in #H_RecipeResultTransfer.
Use the above steps to change Element 1 of Recipe A.

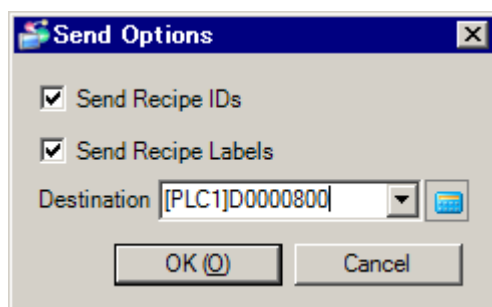
Confirming Which Recipe Data Is Transferred (Send Options)

When you transfer enhanced recipes to the device/PLC, you can write the recipe ID and recipe label of these enhanced recipes to a defined address.



Setup Procedure

1. In [Common Settings] - [Enhanced Recipe Settings], open the Recipe Group tab (Recipe Group 1), and then click [Send Options].
2. In the [Send Options] dialog box, set the destination address and the items you want to transfer, and then click "OK".



Send Options Data Structure

To write to the [Destination], use one word for the recipe ID and 32 words for the recipe label. If you select both the recipe ID and recipe label in the options above, the recipe ID is stored first, followed by recipe label.

The recipe label is stored in the [Destination] using the character code defined in the enhanced recipe group's [Language] setting. If there are two or more label languages, the system uses the language of the label displayed when the enhanced recipe is transferred as defined in [Select label language].

| | |
|-----|--------------|
| +0 | Recipe ID |
| +1 | Recipe Label |
| +32 | |

NOTE

Y The order that the recipe label text is stored follows the settings in [System Settings] - [Device/PLC] - [Text Data Mode].

Character codes for recipe labels

Recipe labels written to the [Destination] are stored using one of the following language codes, depending on the selected language.

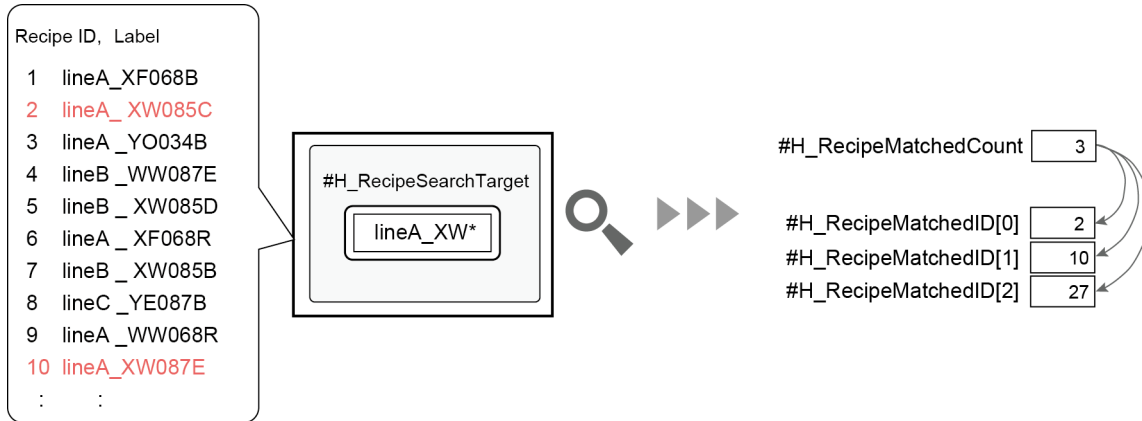
| Language | Compatible Character Codes |
|--------------------------|----------------------------|
| ASCII | ASCII |
| Japanese | Shift_JIS |
| Chinese (Traditional) | Big5 |
| Chinese (Simplified) | GB |
| Korean | KS C 5601 |
| Russian (Cyrillic) | Code page 866 |
| Thai | Code page 874 |

Searching for Enhanced Recipes Using System Variables

Using the system variables, you can specify search conditions, run searches, and confirm search results.

When you search for an enhanced recipe label, the number of recipes matching your search conditions and their IDs are stored in system variables.

This gives you more flexibility in creating screens, as you can search without using dedicated parts of enhanced recipes.



Setup Procedure

To search for enhanced recipes using system variables, use the following system variables.

| Variable name | Description | Type | Type |
|--|--|--|---------|
| #H_RecipeSearchTarget | Search text when searching enhanced recipe label | String | Control |
| #H_RecipeSearchOption | Conditions for enhanced recipe search | 16-bit integer variable | |
| #H_RecipeControlSearch | Search for enhanced recipes | Bit variable | |
| #H_RecipeResultSearch | Search status | 16-bit integer variable | Status |
| #H_RecipeMatchedCount | Number of search results | 16-bit integer variable | |
| #H_RecipeMatchedID[xx] | Matching enhanced recipe IDs found in search | 16-bit integer variable array xx: 0 to 63 | |

Operating Procedure

- Specify the search text in #H_RecipeSearchTarget.
Example: "lineA_XW**"

NOTE

Y When there are two or more label languages, the search looks targets the language set at the time of the search.
Define the search text string in that language's character code.

- Specify the search conditions in #H_RecipeSearchOption.
- Turn ON #H_RecipeControlSearch to start the enhanced recipe search. When the search process starts, bit 0 of #H_RecipeResultSearch turns ON.

4. When the search process is complete, bit 1 of #H_RecipeResultSearch turns ON. In #H_RecipeMatchedCount, you can identify the number of enhanced recipes matching the search condition. Starting at #H_RecipeMatchedID[0], see the IDs of the enhanced recipes that match the search conditions.
Example: When #H_RecipeMatchedCount is 10, the values of #H_RecipeMatchedID[0] to #H_RecipeMatchedID[9] are the IDs of the enhanced recipes that match the search conditions.
5. Confirm the search is complete in #H_RecipeResultSearch bit 1, and turn OFF #H_RecipeControlSearch.

NOTE

⚠ When you turn OFF #H_RecipeControlSearch, bit 1 of #H_RecipeResultSearch also turns OFF.

Common Settings

Enhanced Recipe Group Settings

| Group ID | Name | Language | Address | Comment |
|----------|---------------|----------|---------|---------|
| 1 | Recipe Group1 | ASCII | Random | |

Importing to

Import the recipe group settings created in CSV format.

When you select the recipe group to overwrite from the recipe group list, [Import] becomes selectable.

Recipe Group Settings

Send Options

When you transfer enhanced recipe data to the device/PLC, write the [Recipe ID] and [Recipe Label] to the [Destination].

Send Recipe IDs

Select whether or not to send a recipe ID to the address.

Send Recipe Labels

Select whether or not to send the recipe label to the address.

Destination

Set the destination address to store the recipe ID and recipe label.

You need 1 word for the recipe ID, and 32 words for the recipe label.

| | |
|-----|--------------|
| +0 | Recipe ID |
| +1 | Recipe Label |
| +32 | |

If you do not send the recipe ID, the recipe label is stored in 32 words starting from the start address.

Element List

Map

Link the element with the mapped address. When you link the element, the element value of the selected recipe is written to the mapped address. This write operation occurs whenever you select a recipe or change an element value. To reflect the mapped address value in the recipe element value, use the system variable #RecipeControlMapAddress.

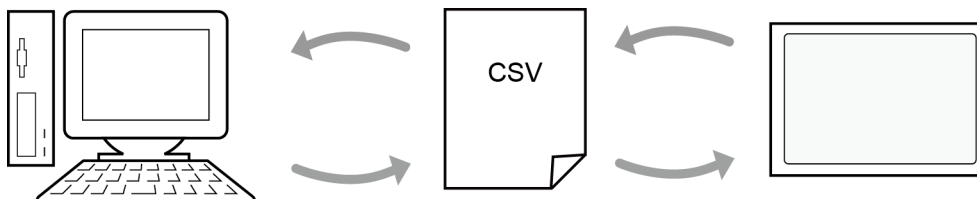
[Edit] check box must be selected to work with the element list.

Mapped Address

Set the element and associated address.

Importing/Exporting

You can now import CSV files into the enhanced recipe settings in GP-Pro EX. You can easily reflect recipe settings changed on the display unit or edited in CSV files in the GP-Pro EX project data.



Export Function

This function outputs the settings for enhanced recipe groups in CSV (comma-separated values) files. You can export settings for enhanced recipe groups from both GP-Pro EX and the display unit, and edit the recipe data in spreadsheet software.

NOTE

Y External storage is required when importing and exporting enhanced recipe data on the display unit.

Import Function

This function imports a CSV file exported from GP-Pro EX or the display unit, and updates the recipe group settings. Import enhanced recipe settings to update existing recipe settings. You cannot import new recipe groups to the project data.

The following conditions will interrupt importing.

| | Importing to GP-Pro EX | Importing to Display |
|--|------------------------|----------------------|
| Label Language is different (No labels match both the label No. and language code.) | Import interrupted | Import interrupted |
| Data Language is different | Import interrupted | Import interrupted |
| Number of elements is different | Import interrupted | Import interrupted |
| Recipe label in the CSV file is blank | Import interrupted | Import interrupted |
| Data format of element value is different | *1 | Import interrupted |
| Recipe label in the CSV file is blank | *2 | Import interrupted |
| Recipe ID in CSV file is blank / duplicated / out of range | *3 | Import interrupted |

*1. A confirmation dialog box is displayed to continue import processing. If you select [No], importing is interrupted. If you select [Yes] or [Yes to All], importing continues without updating the corresponding element values.

*2. "Recipe" is set as the recipe label and import continues. A confirmation message will not appear.

*3. If you set a blank or out-of-range ID, the recipe ID is automatically set and import continues. If the ID is a duplicate, a dialog box will appear, where you will choose whether to overwrite it. However, if you select "Yes to All", the ID will be overwritten with the recipe on the last row.

Editing the CSV File

After you export the enhanced recipe data, you can open and edit it in a spreadsheet. However, do not perform any action not listed below. An error will occur and you will not be able to import.

| Action | Details |
|-----------------------|--|
| Change element value | Change the set values for each recipe. Set text data with the same language code used for Data Language. |
| Add recipe | Add a recipe by adding one recipe data row in the CSV file. Be careful not to duplicate an existing recipe ID. |
| Delete recipe | Delete a recipe by deleting one recipe data row in the CSV file. |
| Change element labels | When you change an element label, use the same language code used for the Label Language. |
| Change recipe label | When you change a recipe label, use the same language code used for the Label Language. |

NOTE

- When there are two or more label languages, unless the CSV file defines the label, it is automatically populated with default text. When importing into GP-Pro EX, this text is "Recipe". When importing into the display, the label is "NewRecipe#". (# indicates a recipe ID)
- When adding a recipe, the security level of the added recipe is set to 0.

File Name

Export enhanced recipe data using the following file name structure.

Enhanced recipe group ID (5 digit number)_enhanced recipe group name.csv

Example: If the group ID is "1" and the enhanced recipe group name is "Recipe Group1", the file name is

"00001_Recipe Group1.csv"

NOTE

- Do not change the file name. If you import duplicate IDs into the display unit, an error will occur and the import will fail.

CSV File Format

The following illustrates the format of the CSV file when exporting an enhanced recipe group.

00001_Recipe Group1.csv (Group ID = "1", Name = "Recipe Group1", Language = "Japanese")

| | | | | |
|-----------------|-------|---------------|----------|----------|
| :RecipeData | | | | |
| :Version | 2 | | | |
| :RecipeGroup | 1 | Recipe Group1 | | |
| :Label Language | en-US | ASCII | 1 | |
| :Data Language | ja-JP | Japanese | | |
| | | | | |
| | | Element1 | Element2 | Element3 |
| | 1 | RecipeA | 100 | -50 |
| | 2 | RecipeB | 200 | -60 |

Y **:RecipeData**

A header that identifies the file as containing enhanced recipe data. Do not edit.

Y **:Version, <file version>**

This is the file version information. Do not edit.

Y **:RecipeGroup, <recipe group ID>, <recipe group name>**

This is the recipe group ID and recipe group name.

Y **:Label Language, <language code>,<language name>,<label No.>**

This is the language of the recipe label and element labels with the corresponding language code and label number. Do not edit.

If there are two or more label languages, when exporting from GP-Pro EX you can define the label language. When you export from the display unit, the current language is output. When you import, the label must use the same language code and label number as the recipe group you are importing to.

Y **:Data Language, <Language Code>,<Language Name>**

This is the language of the element value text and the corresponding language code. Do not edit.

This is the language information used for text data in the enhanced recipe group.

Y **<Blank line>**

This is a blank line. Do not edit this.

1 blank line is required. If the blank line is missing or if there are 2 or more blank lines, import will fail.

Y **, ,<Element 1 Label>,<Element 2 Label>,...,<Element n Label>**

The element label of the Label No. selected in Label Language is output. When you edit the label, use the language specified for the Label Language. You cannot add or delete elements. You cannot import when the number of elements in the recipe group in the import destination does not match the number of elements in the CSV file.

Y **<Recipe ID>, <Recipe label>, <Value of Element 1>, <Value of Element 2>, ..., <Value of Element n>**

This is recipe data. For each recipe, one row is output. The recipe label is output in the language set in the Label Language, and element values (Element 2), which are strings, are output in the language set in Data Language. To edit recipe labels and element values, edit each in the same language as the corresponding language code.

Do not duplicate recipe IDs. If you import a duplicate ID into GP-Pro EX, the recipe is overwritten with the recipe in the final row. You cannot import duplicate IDs into the display unit. An error will occur.

Language Code

Depending on the selected language, the following language code and language name are output when exporting.

| Language Code | Language |
|---------------|-----------------------|
| ja-JP | Japanese |
| en-US | ASCII |
| zh-TW | Chinese (Traditional) |
| zh-CN | Chinese (Simplified) |
| ko-KR | Korean |
| ru-ru | Russian (Cyrillic) |
| th-TH | Thai |

Mapped Address

The mapped address is an address linked to a defined element. Each element of an enhanced recipe group can be linked to a mapped address. When you select a recipe or edit an element value, the value of the element is written to the mapped address. By assigning a mapped address, you can edit and display element values without using enhanced recipe parts.

After editing an element value using a mapped address, use a system variable to save the element value. However, editing the mapped address value alone will not change the enhanced recipe in the display unit. If you transfer an enhanced recipe without saving the element value, you will transfer the value before the edit.

NOTE

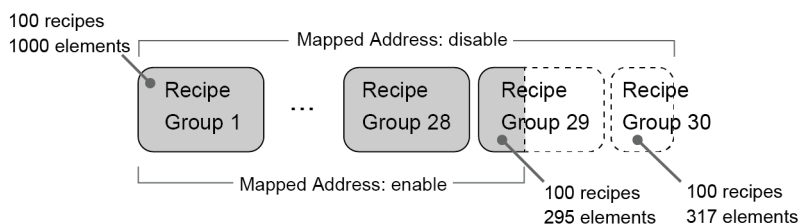
Y Log in at or above the security level set for the recipe, before performing these operations. If you are logged in at a security level less than the recipe security level, you cannot save mapped address values as element values even if you use system variables.

Y Using [Mapped Address] reduces the maximum number of enhanced recipes.

Example: Recipe group with two label languages, 16-bit data format for all elements, and no input limitations

If you don't use mapped addresses: You can create 29 recipe groups with 100 recipes and 1000 elements each, and one more recipe group with 100 recipes and 317 elements.

If you use mapped addresses (for all elements): You can create 28 recipe groups with 100 recipes and 1000 elements each, and one more recipe group with 100 recipes and 295 elements.



Updating mapped addresses and element values

| Recipe | | | | | | | | | | | | | |
|--|----------|---------------------|----------------|-------|-------------------------------------|--------------------------|------|----|-------------------------------------|----------------|---------|---------|---------|
| <div>Add Remove Send Options</div> | | | | | | | | | | | | | |
| RecipeID | Label 1 | Security Level | | | | | | | | | | | |
| 1 | RecipeA | 0 | | | | | | | | | | | |
| 2 | RecipeB | 0 | | | | | | | | | | | |
| 3 | RecipeC | 0 | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| Element | | | | | | | | | | | | | |
| <div>Remove Show Only Label and Value <<</div> | | | | | | | | | | | | | |
| | Label 1 | Data Type | Address | Units | Editable | Input c | From | To | Map | Mapped Address | RecipeA | RecipeB | RecipeC |
| 1 | Element1 | 16 Bit Dec Unsigned | [PLC1]D0000100 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input checked="" type="checkbox"/> | [PLC1]D0000500 | 100 | 300 | 500 |
| 2 | Element2 | 16 Bit Dec Signed | [PLC1]D0000110 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input type="checkbox"/> | | -10 | -60 | -70 |
| 3 | Element3 | Bit | [PLC1]X00120 | 1 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | | <input type="checkbox"/> | | OFF | ON | ON |

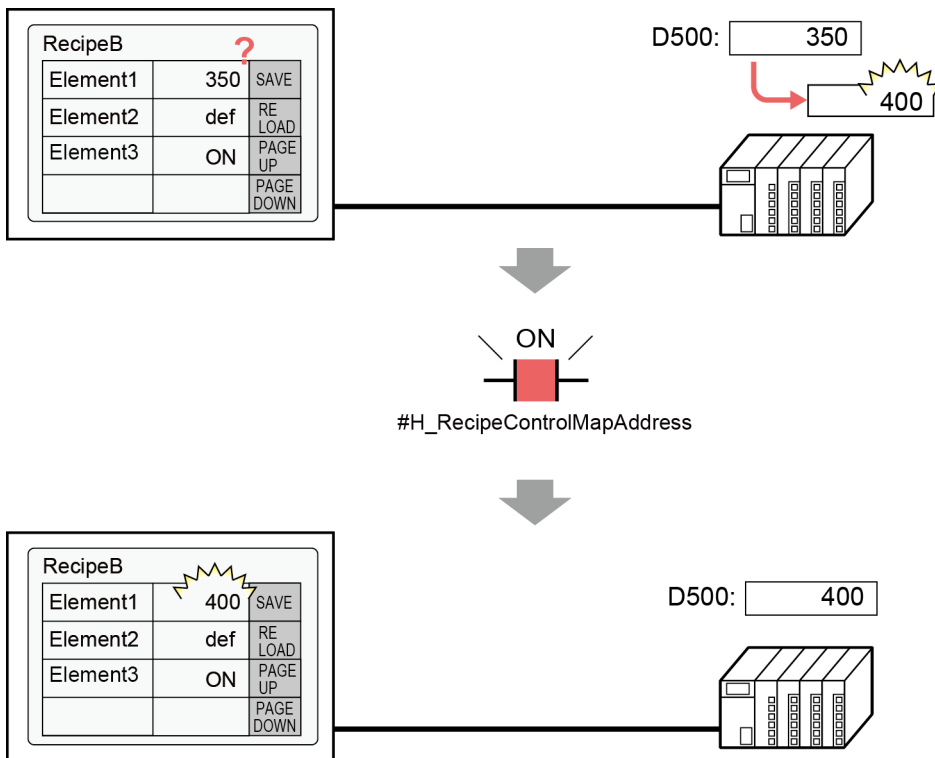
If you select Recipe A, the value of Element 1 (100) is written to the mapped address (D500). If you select Recipe B, the value of Element 1 (300) is written to the mapped address.



If you change the value of an element value for Recipe B from the display, and save this change, the value of the mapped address (D500) will also change.



Even if you directly edit the value in the mapped address (D500), the element values of the enhanced recipe will not change. You can use the system variable #H_RecipeControlMapAddress to save the element value in the current enhanced recipe. Always save the element value using system variables before transferring enhanced recipes.



System Variables

The following are new and updated system variables.

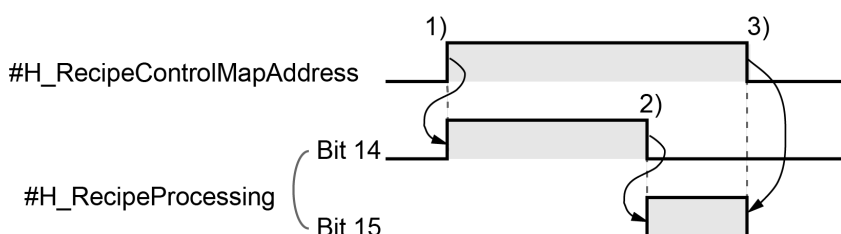
| Variable name | Description | Type | Read | Write |
|--|--|--------------------|------|-------|
| #H_RecipeControlMapAddress | Save the mapped address value in the recipe data | Bit | ✓ | ✓ |
| #H_RecipeProcessing | Processing status such as transferring or editing enhanced recipe data | Integer | ✓ | - |
| #H_RecipeResultTransfer | Transfer result of enhanced recipe data | Integer | ✓ | - |
| #H_RecipeSearchTarget | Search text when searching enhanced recipe labels | String | ✓ | ✓ |
| #H_RecipeSearchOption | Conditions for searching enhanced recipes | Integer | ✓ | ✓ |
| #H_RecipeControlSearch | Search control for enhanced recipes | Bit | ✓ | ✓ |
| #H_RecipeResultSearch | Search status | Integer | ✓ | - |
| #H_RecipeMatchedCount | Number of hits | Integer | ✓ | - |
| #H_RecipeMatchedID[xx] | Recipe IDs that match the search conditions | Integer array [64] | ✓ | - |
| #H_RecipeResultCSV | Results of importing/exporting enhanced recipe data | Integer | ✓ | - |

#H_RecipeControlMapAddress

This bit variable saves the mapped address value as an enhanced recipe element value.

When you turn ON this variable, you overwrite the element value of the enhanced recipe data with the value of the mapped address. The recipe selected on the display is edited. You cannot cancel a save in progress.

This system variable is not turned OFF automatically. Confirm with #H_RecipeProcessing that the update is complete, and then turn OFF the variable.



1) When you turn ON #H_RecipeControlMapAddress, this starts overwriting the element value with the mapped address, and bit 14 of #H_RecipeProcessing turns ON.

2) When the overwrite operation is complete, bit 14 of #H_RecipeProcessing turns OFF and bit 15 turns ON.

3) Confirm that bit 15 of #H_RecipeProcessing is ON, and then turn OFF #H_RecipeControlMapAddress. When you turn OFF #H_RecipeControlMapAddress, bit 15 of #H_RecipeProcessing also turns OFF.

#H_RecipeProcessing

This 16-bit integer variable stores the processing status of transferring and editing enhanced recipe data. Each bit indicates if processing is in progress or complete.

| Bit | Description | Details |
|-----|--------------------------------|---|
| 0 | Transfer in progress | Stores the status of transfer from the display unit to the device/PLC |
| 1 | Transfer complete | |
| 2 | Receive in progress | Stores the status of transfer from the device/PLC to the display unit |
| 3 | Receive complete | |
| 4 | Export in progress | Stores the status of CSV output to external storage |
| 5 | Export complete | |
| 6 | Import in progress | Stores the import status from external storage |
| 7 | Import complete | |
| 8 | Updating enhanced recipe list | Stores the update status of the enhanced recipe list |
| 9 | Copying enhanced recipe list | Stores the processing status for editing, copying, and deleting the recipe label |
| 10 | Deleting enhanced recipe list | |
| 11 | Editing enhanced recipe list | |
| 12 | Reloading enhanced recipe list | Stores the update status of the enhanced recipe list |
| 13 | Saving enhanced recipe list | Stores the save status of the edited element values |
| 14 | Saving mapped address | When the mapped address value starts saving over the enhanced recipe's element value, this bit turns ON. When the save is complete, this bit turns OFF. |
| 15 | Saving mapped address complete | Turns ON when the save operation for bit 14 (saving mapped address) is complete. |

#H_RecipeResultTransfer

This 16-bit integer variable stores the results of enhanced recipe data transfer/receive processes and mapped address save processes.

| Error code Dec | Description | Details |
|-------------------|----------------------------------|--|
| 0 | Completed successfully | |
| 1 | Incorrect #H_RecipeGroupID value | The recipe group ID value is incorrect. The specified recipe group does not exist or is out of range. |
| 2 | Incorrect #H_RecipeID value | The recipe ID value is incorrect. The specified recipe ID does not exist or is out of range. |
| 3 | Access failed | During transfer/receive, access to the device failed. |
| 4 | Incorrect element value | Receive process is canceled because values outside the setting range are included in the element values. |

| | | |
|--------|--------------------------------|--|
| 5 | Failed to update element value | Receive process was canceled because the element values are being edited. You can use #H_RecipeStatusEdit. |
| 6 | Failed to update element value | Receive process is canceled because invalid characters are included in the received element values. |
| 7 | Failed to update element value | Could not save the mapped address value in the element because the recipe was being saved or received. |
| 8 | Incorrect security level | Access is not allowed because the recipe's security level is higher than the current login security level. Please login with a security level at or above the recipe security level. |
| 9 ~ 12 | Reserved | - |
| 13 | Write error | Write unsuccessful. Transfer again. |
| 14 | Write error | |
| 15 | Insufficient capacity | Cannot update due to insufficient space in the screen area |

#H_RecipeSearchTarget

Specify the enhanced recipe label to target in your search. Using this system variable you can specify up to 64 bytes of continuous data, up to 64 single-byte or 32 double-byte characters.

NOTE

- Y When there are two or more label languages, the search looks targets the language set at the time of the search. When specifying your search text, use character codes that match language of the recipe label.
- For the various character codes supported by each language, refer to "Character codes for recipe labels" in the [Confirming Which Recipe Data Is Transferred](#) section.

For searching the recipe label, you can specify the wildcards "*" and "?".

Y *: Any text string

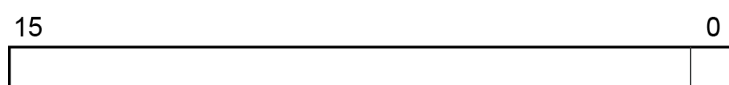
Example: When you specify "Recipe*" in #H_RecipeSearchTarget, the search results include Recipe A, Recipe AB, Recipe ABC, and so on.

Y ?: Any one character

Example: When you specify "Recipe??" in #H_RecipeSearchTarget, the search results include Recipe AB, Recipe XX, and so on. In this case, Recipe A has fewer letters than the search conditions, and Recipe ABC has more, so these are not in the search results.

#H_RecipeSearchOption

This 16-bit integer variable specifies the enhanced recipe search conditions.



Y Bit 0: Distinguishes between upper and lower case letters.

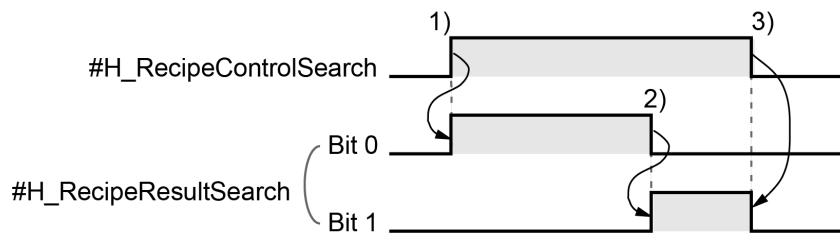
When OFF, the search is case-sensitive. When ON, the search is not case-sensitive.

#H_RecipeControlSearch

This bit-type system variable performs the enhanced recipe search.

Turn ON to start the recipe search within the enhanced recipe group specified in #H_RecipeGroupID. You cannot cancel a search in progress. Before you start a search, specify the target enhanced recipe label in #H_RecipeSearchTarget, and the search options in #H_RecipeSearchOption.

#H_RecipeControlSearch does not automatically turn OFF. Confirm with #H_RecipeResultSearch that the search is complete, and then turn OFF the variable.



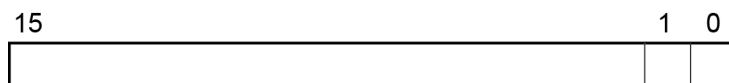
1) When the search starts, #H_RecipeResultSearch bit 0 turns ON.

2) When the search ends, the #H_RecipeResultSearch bit 1 changes from OFF to ON.

3) Confirm that #H_RecipeResultSearch bit 1 is ON, and then turn OFF #H_RecipeControlSearch. When you turn OFF #H_RecipeControlSearch, #H_RecipeResultSearch bit 1 turns OFF.

#H_RecipeResultSearch

This 16-bit integer variable stores the processing status of the enhanced recipe search.



Y Bit 0: Search in progress

When you start the enhanced recipe search, this variable turns ON. When the search is complete, it automatically turns OFF.

Y Bit 1: Search complete

When the enhanced recipe search is complete, this variable turns ON. When the search execution system variable #H_RecipeControlSearch turns OFF, this variable automatically turns OFF.

#H_RecipeMatchedCount

This 16-bit integer variable stores the number of enhanced recipes in the search results after the enhanced recipe search is complete. When no matching enhanced recipes are found, "0" is stored.

#H_RecipeMatchedID [0] to [63]

This system variable stores the recipe IDs of the enhanced recipes found in the enhanced recipe search. It is an integer-type array variable.

Up to 64 recipe IDs are stored in ascending order starting from #H_RecipeMatchedID[0]. When the search results contain 65 or more recipes, the 65th and subsequent results are discarded.

NOTE

- When no recipe labels match the enhanced recipe search conditions, the value of #H_RecipeMatchedID[n] is not updated. The value from the last search remains.
- You cannot use #H_RecipeMatchedID[n] in logic programs.

#H_RecipeResultCSV

This 16-bit integer variable stores the processing results for importing/exporting enhanced recipe data as a CSV file.

| Error code Dec | Description | Details |
|-------------------|---|--|
| 0 | Completed successfully | |
| 1 | Incorrect #H_RecipeGroupID value | |
| 2 | Incorrect #H_RecipeID value | Incorrect recipe ID value in CSV file. The recipe ID is duplicated, out of range, or not set. |
| 3 | Import unsuccessful | Wrong CSV file format |
| 4 | Import unsuccessful | There are no CSV files to be replaced |
| 5 | Import unsuccessful | Not enough columns in CSV file format |
| 6 | Import unsuccessful | Too many columns in CSV file format |
| 7 | Import unsuccessful | Not enough rows in CSV file format |
| 8 | Import unsuccessful | Too many element value characters (text string is too long), or invalid characters included |
| 9 | Import unsuccessful | Invalid values outside the element values (out of range) |
| 10 | Import stopped | Importing interrupted because enhanced recipe element values are being edited. |
| 11 | There are no files to be imported | Files for importing cannot be found |
| 12 | Generation of export files unsuccessful | Files for exporting could not be generated. The probable causes are as follows: <ul style="list-style-type: none"> • Insufficient external storage capacity at the export destination. • File is set to read-only. |
| 13 | Write error | Write unsuccessful. Try again to import or export. |
| 14 | Write error | |
| 15 | Insufficient capacity | Cannot update due to insufficient space in the screen area |
| 16 | Reserved | - |
| 17 | Import unsuccessful | The label language code is invalid. The language code listed as the Label Language in the CSV file is incorrect. Alternatively, the Label |

| | | |
|----|---------------------|--|
| | | Language in the CSV file and the language code of the current recipe label are different. |
| 18 | Import unsuccessful | The language code of the text data is invalid. The language code listed as the Data Language in the CSV file is incorrect. Alternatively, the Data Language in the CSV file and the language code of the current recipe label are different. |
| 19 | Import unsuccessful | There is an item in the CSV file not already set up in the recipe label. |
| 20 | Import unsuccessful | There is an element label item in the CSV file not already set up in the element label. |

Display-related Errors

| Error code | Error message | Cause | Solution |
|------------|--|--|---|
| RAAA162 | Enhanced recipe data (Index File) is invalid. Unable to use. | An abnormality has occurred in the enhanced recipe data. Alternatively, the display unit system may be old. | Transfer the enhanced recipe data. Alternatively, use the Transfer Tool to execute a forced transfer. |