MITSUBISHI ELECTRIC

Changes for the Better

Numerical Control (CNC)

Macro Function Manual

NC Designer2

MELSOFT

IB-1501500(ENG)-G

Introduction

This instruction manual describes how to use the macro function of NC Designer2. Incorrect handling may lead to unforeseen accidents, so make sure to read this instruction manual thoroughly before operation to ensure correct usage.
NC Designer2 supports the following NC series. Some of the functions are not adapted to the NC Designer2, depending on its series.

Supported models	Abbreviations in this manual
M800VW (Windows-based display unit)	M800VW/M80VW (Windows-based display unit)
M80VW (Windows-based display unit)	
M800W (Windows-based display unit)	M800/M80 (Windows-based display unit)
M80W (Windows-based display unit)	
M800VW (Windows-less display unit)	M800VW/M80VW (Windows-less display unit)
M800VS	
M80VW (Windows-less display unit)	
M80V	
M800W (Windows-less display unit)	M800/M80 (Windows-less display unit)
M800S	
M80W (Windows-less display unit)	
M80	
E80	
M700VW	M700VW
M700VS	M700VS/M70V/E70
M70V	
E70	

Notes on Reading This Manual

(1) For the specifications of individual machine tools, refer to the manuals issued by the respective machine tool builders. The "restrictions" and "available functions" described by the machine tool builders have precedence over this manual.

(2) This manual describes as many special operations as possible, but it should be kept in mind that operations not mentioned in this manual cannot be performed.

In this manual, the following abbreviations might be used.
MTB: Machine tool builder

Precautions for Safety

Always read the specifications issued by the machine tool builder, this manual, related manuals and attached documents before installation, operation, programming, maintenance or inspection to ensure correct use.
Understand this numerical controller, safety items and cautions before using the unit.
This manual ranks the safety precautions into "DANGER", "WARNING" and "CAUTION".

⚠ DANGER	When the user may be subject to imminent fatalities or major injuries if handling is mistaken.
⚠ WARNING	When the user may be subject to fatalities or major injuries if handling is mistaken.
⚠ CAUTION	When the user may be subject to injuries or when property damage may occur if handling is mistaken.

The following sings indicate prohibition and compulsory.

	This sign indicates prohibited behavior (must not do). For example,  indicates "Keep fire away".
	This sign indicated a thing that is pompously (must do). For example,  indicates "it must be grounded".

The meaning of each pictorial sing is as follows.

 CAUTION	 CAUTION rotated object	 CAUTION HOT	 Danger Electric shock risk	 Danger explosive
 Prohibited	 Disassembly is prohibited	 KEEP FIRE AWAY	 General instruction	 Earth ground

Note that even items ranked as " ⚠ **CAUTION**", may lead to major results depending on the situation. In any case, important information that must always be observed is described.

⚠ DANGER

Not applicable in this manual.

⚠ WARNING

Not applicable in this manual.

⚠ CAUTION

1. Items related to product and manual

⚠ If the descriptions relating to the "restrictions" and "allowable conditions" conflict between this manual and the machine tool builder's instruction manual, the latter has priority over the former.

⚠ The operations to which no reference is made in this manual should be considered impossible.

⚠ This manual is compiled on the assumption that your machine is provided with all optional functions. Confirm the functions available for your machine before proceeding to operation by referring to the specification issued by the machine tool builder.

⚠ In some NC system versions, there may be cases that different pictures appear on the screen, the machine operates in a different way on some function is not activated.

⚠ To protect the availability, integrity and confidentiality of the NC system against cyber-attacks including unauthorized access, denial-of-service (Dos) (*1) attack, and computer virus from external sources via a network, take security measures such as firewall, VPN, and anti-virus software.

(*1) Denial-of-service (Dos) refers to a type of cyber-attack that disrupts services by overloading the system or by exploiting a vulnerability of the system.

⚠ Mitsubishi Electric assumes no responsibility for any problems caused to the NC system by any type of cyber-attacks including DoS attack, unauthorized access and computer virus.

Trademarks

MELDAS, MELSEC, EZSocket, EZMotion, iQ Platform, MELSOFT, GOT, CC-Link, CC-Link/LT and CC-Link IE are either trademarks or registered trademarks of Mitsubishi Electric Corporation in Japan and/or other countries.

Ethernet is a registered trademark of Xerox Corporation in the United States and/or other countries.
TRELLIX is a registered trademark or a trademark of Musarubra US LLC or its subsidiaries in the United States and other countries.
Microsoft and Windows are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.
CompactFlash and CF are either trademarks or registered trademarks of SanDisk Corporation in the United States and/or other countries.
SD Logos are trademarks or registered trademarks of SD-3C, LLC in the United States, other countries or both.
Wind River Systems, Inc., VxWorks and Tornado are either trademarks or registered trademarks of Wind River Systems, Inc. in the United States and/or other countries.

Other company and product names that appear in this manual are trademarks or registered trademarks of the respective companies.

CONTENTS

1. What Is the Macro Function? 1
2. Macro Execution Conditions 1
    2.1 Project Macro Execution Condition 1
    2.2 Screen Macro Execution Condition 2
    2.3 Macro Execution Timing 3
3. Macro Editing 4
    3.1 Macro Editing 4
    3.2 Macro Editing Dialog Box 5
    3.3 Event Creation Dialog Box 6
4. Macro Programming 7
    4.1 Macro Editing Area 7
    4.2 Program Describing Method 7
    4.3 Programming Language 10
5. Function Details 30
    5.1 Normal Command 30
    5.2 Control Common Command 41
    5.3 Button 60
    5.4 CheckBox 77
    5.5 Edit 89
    5.6 HtmlBrowser 113
    5.7 Label 120
    5.8 List 124
    5.9 Picture 143
    5.10 ProgressBar 150
    5.11 RadioButton 157
    5.12 ScrollBar 170
    5.13 TextBox 184
    5.14 NCPLCButton 209
    5.15 NCPLCTextbox 239
    5.16 NCDataTextBox 272
    5.17 Menu 304
    5.18 FileInOut 324
    5.19 AlarmMessage 332
    5.20 MonitorStatus 333
    5.21 Counter 334
    5.22 CycleTime 336
    5.23 LoadMeter 337
    5.24 SPCommand 338
    5.25 NCTable 339
    5.26 InputBox 357
    5.27 SoftKey 371
    5.28 Graph 388

5.29 AlarmList 411
5.30 Extension Menu 416
6. NC Data Access Function 468
7. String Operation Functions 479
8. Other Functions 487
9. Error Message List 488
Appendix 490
Appendix 1. Data Type Definitions 490
Appendix 2. Macro Event Arguments 498
Appendix 2.1 OnKeyPress/OnKeyRelease Arguments 499
Appendix 2.2 OnPress/OnRelease Arguments 500
Appendix 2.3 OnScroll Arguments 501

1. What Is the Macro Function?

The macro function allows you to execute various processes during operation of a control according to processes described in the macro program. Because control processes are described on NC Designer2 with the macro language, those who are unfamiliar with C++ language programming can add control processes easily. The macro includes two variations: "project macro" specified for the project and "screen macro" specified for controls on the screen.

2. Macro Execution Conditions

The macro includes two types: "project macro" specified for the project and "screen macro" specified for screens. In this section, the macro execution conditions and timing are described.

2.1 Project Macro Execution Condition

The project macro is executed repetitively while the application window is open. The execution conditions of the project macro are specified below.

Macro execution condition	Description
Launch of application window	The macro starts when the application window is launched, and it is executed repetitively until the application window is closed.

2.2 Screen Macro Execution Condition

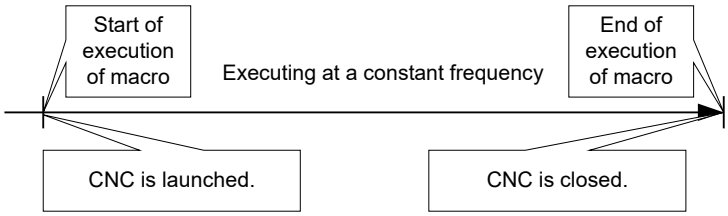
The screen macro is specified for controls existing on the screen. Execution of a screen macro corresponds to the callback function of the control. That is, the event held by each control is the execution condition of the macro.

Macro execution condition	Description
OnKeyPress	Executed after the key is pressed. Each control parts can acquire the key event only when focus is active.
OnKeyRelease	Executed after the key is released. Each control parts can acquire the key event only when focus is active.
OnPress	Executed after the mouse or another pointing device is pressed.
OnRelease	Executed after the mouse or another pointing device is released.
OnClick	Executed after the mouse or another pointing device is clicked. If the pointing device is released on the same control, execution follows OnRelease.
OnDraw	Executed after the image is drawn.
OnTimer	Executed after the timer event process is called.
OnSetFocus	Executed for the focus hits the control.
OnKillFocus	Executed after the focus moves apart from the control.
OnCreate	Executed after the page/control is generated.
OnDelete	Executed before the page/control is deleted.
OnUser	Executed after an original event of the user.
OnScroll	Executed after the scroll bar is clicked with the mouse or another pointing device.
OnScrollFinish	Executed after the caption character string has finished scrolling.
OnSelectChange	Executed when the selected line is changed in the list.

2.3 Macro Execution Timing

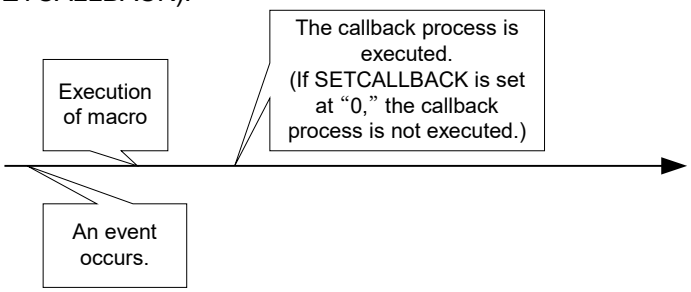
Project Macro

The starting and finishing timings of the project macro are shown below.



Screen Macro

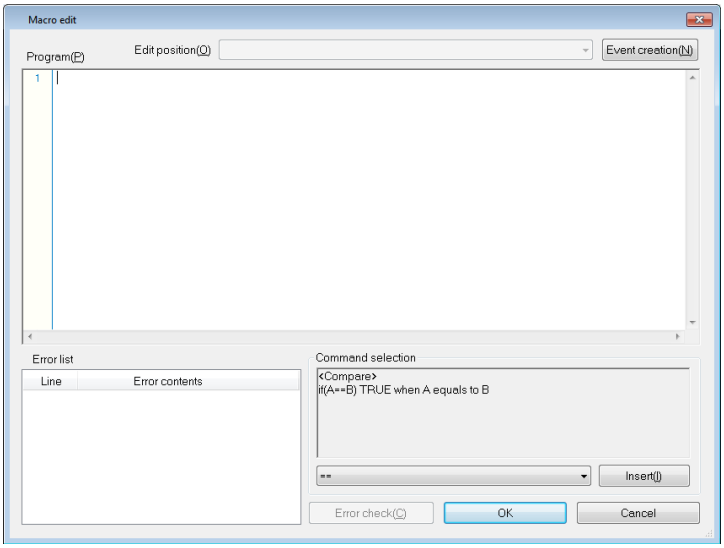
The screen macro execution timing is the same as the execution timing of the callback function of each control. The callback function is executed after the macro is executed. Whether the callback process is executed or not after execution of the macro can be controlled with the macro command (SETCALLBACK).



3. Macro Editing

3.1 Macro Editing

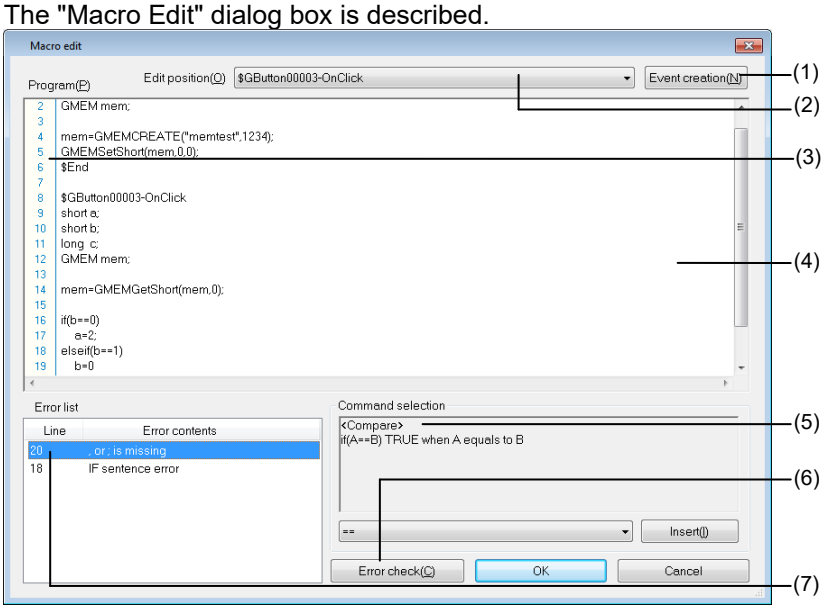
The procedure for registering or editing the project macro or screen macro is described here.
1. From the [Settings] menu, select [Project macro edit]/[Panel macro edit].
2. The "Macro edit" dialog box is displayed.



3. Write the program and click on the "OK" button to register the macro.

3.2 Macro Editing Dialog Box

The "Macro Edit" dialog box is described.



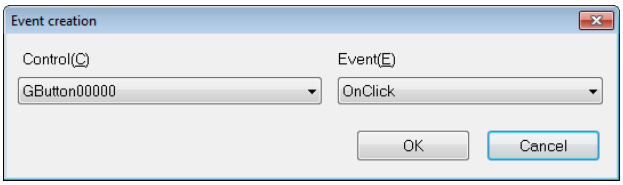
No.	Item	Description
(1)	Event creation	The "Event Creation" dialog box for adding an event in the program editing area is displayed. This part may not be operated when the project macro is edited.
(2)	Edit position	A list of macro events is displayed. Select a desired event from the combo box to move the cursor to the editing position of the selected event. The events are displayed in the alphabetical order in the combo box.
(3)	Row number	Indicates a row number.
(4)	Program	Area for describing the macro program
(5)	Command selection	When a command is selected from the combo box, description about the selected command is displayed in the above field. Click on the [Insert] button to insert the selected command at the cursor in the program editing area.
(6)	Error check	Syntax errors in the program description are checked for.
(7)	Error list	A list of detected errors is displayed. Double click on an error to move the cursor to the top of the corresponding line in the editing area.

NOTE

◆ To add events, use [Event Creation] dialog which is displayed by pressing [Event Creation] button.

3.3 Event Creation Dialog Box

Create the control event for which the macro program is described.



Item	Description
Control	A list of all controls existing in the corresponding screen is displayed. The control names are displayed in the alphabetical order in the combo box.
Event	A list of events that can be used for the selected control is displayed. The event names are displayed in the alphabetical order in the combo box.
OK	Insert the selected control name and event name to the cursor position in the program editing area of the macro editing dialog box and close the "Event creation" dialog box.

4. Macro Programming

The macro program describing method and programming language are described here.

4.1 Macro Editing Area

The following shows the maximum number of lines per screen that can be edited in the macro editing area, and the number of characters that can be displayed in one row.

Item	Description
Max. number of lines per screen	10000
Number of characters that can be displayed in one row	100 one-byte characters (Note)

(Note) If the number of characters exceeds the window width, the horizontal scroll bar appears.
When the macro editing dialog box is resized, the number of characters displayed in one row is also changed.

NOTE

- In the macro editing dialog box, copy, paste, cut, and undo (once only) are available with [Ctrl]+[C], [Ctrl]+[V], [Ctrl]+[X], and [Ctrl]+[Z] respectively.
  The same operations are available from the pop-up menu displayed by right-clicking in the macro editing area.

4.2 Program Describing Method

Screen Macro

The configuration of the macro program for each event is specified below.

Item	Description
Header	Specify the control name and event name in the following format. $control name - event name Example: $GButton00000-OnClick
Main body of program	Describe the main body of the macro program.
Footer	$End

Example:

```
$GButton00000-OnCreate                          ...Header
    GMEM mem;
    mem=GMEMCREATE("memetest",1234);            ...Main body of program
    GMEMsetshort(mem,0,0);
$End                                            ...Footer
```

NOTE

- To set the header and footer, use "Event Creation" dialog which is displayed by pressing [Event Creation] button in "Macro Edit" dialog.
- If it is set by the methods other than "Event Creation" dialog, the error check function for grammar may not operate properly.

Project Macro

The configuration of the macro program of the project macro is specified below.

Item	Description
Header	$Project-OnCycle
Main body of program	Describe the main body of the macro program.
Footer	$End

Example:

```
$Project-OnCycle                    ...Header
                                    ...Main body of program
$End                                ...Footer
```

Program Separation

Specify a semicolon (;) at the end of each command as a delimiter of the program. However, do not put a semicolon for IF(), ELSE, ELSEIF() and ENDIF. For FOR() and NEXT, the semicolon may not be put as far as no other command is specified in the same line.

Example:

```
IF(a==0)
   GCSTextboxSetString(-1,"GTextBox00001","000");
ELSEIF (a==1)
   GCSTextboxSetString(-1,"GTextBox00001","001");
ENDIF
```

Comment

To add a comment in the program, put a single quotation mark (') at the top. The part between the single quotation mark and the end of the line is considered to be a comment.
Example:

```
$GButton00000-OnCreate
GMEM mem;                                ' Comment
mem=GMEMCREATE("memetest",1234);
' Comment
GMEMsetshort(mem,0,0);
$End
```

Programming Language Description

Because the upper case and lower case characters are not separated for the programming language used in the macro program, you can specify the program either with the upper or lower case characters.
Mixture of upper and lower case characters is processed correctly.
Example:

- "GMEMCREATE" and "GMemCreate" are handled as the same function.
- "SHORT VAR_A" and "short var_a" are handled as the same variable.

Function Argument

Specify the constant or the variable as the argument.
Specify the variable which stores the return value to use the functions as the argument.

Example:

```
LONG   lAddress;
LONG   lStatus;
LONG   lData;

lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
```

○ lStatus = melSetData(lAddress, 127, 4, H1, lData) //Only the constant or the variable is specified.

× lStatus = melSetData(ADR_MACHINE(1) | ADR_SYSTEM(1), 127, 4, H1, lData); // The function is specified to the argument.

4.3 Programming Language

Variable

The variable that can be used in the macro program are specified below.

Item	Description
Macro local variable	Variable used in only one project macro or screen macro
Argument variable	Variable of a fixed name holding the argument data of the callback function
Macro reserved word variable	Variable used without declaring the type of variable
NC system variable	Variable that can read and write system variable for NC (#+variable No.)

Macro Local Variable

The macro local variable is a variable that can used in only one project macro or screen macro. Shown below is a list of allowable macro local variables.

Variable	Description
CHAR {local variable name};	Signed 8-bit integer type
SHORT {local variable name};	Signed 16-bit integer type
LONG {local variable name};	Signed 32-bit integer type
DOUBLE {local variable name};	Singed 64-bit floating point type
STRING {local variable name};	Character string type (256 + 1 bytes are occupied)
GMEM {local variable name};	Global memory area type

The describing method of the macro local variable is described below.

Item	Specification
Variable name	Describe with one-byte letters and numbers starting at a letter ('A' to 'Z' or 'a' to 'z') or underscore ('_') (max. 31 characters). The same variable name as the variable type (reserved word) may not be used. Example: SHORT SHORT; ... Do not use.
Describing method	Describe in the following format: "(variable type) (variable name)". Example: SHORT VAR_A; Continuous definition with comma delimiter such as that used in the C++ language is not supported. Example: SHORT VAR_A, VAR_B; ... Do not use.
Allowable size	The local variables may use up to 32767 bytes in each event.

Macro Local Variable (specifying the array)

In M800V/M80V/M800/M80/E80 Series programs, macro local variable can be used to specify the array. The following is the local variables list that can be specified using the array.

Variables	Description
CHAR {local variable name} (no. of elements);	Signed 8bit integer type array
SHORT {local variable name} (no. of elements);	Signed 16bit integer type array
LONG {local variable name} (no. of elements);	Signed 32bit integer type array
DOUBLE {local variable name} (no. of elements);	Signed 64bit floating point type array
STRING {local variable name} (no. of elements);	Character string type array (256 + 1 bytes are occupied)

* Each element consumes the above written size out of the usable size (up to 32,767 bytes).

The following explains the description method of macro local variable (specifying the array).

Item	Specifications
Declaration	To designate an array, describe “variable type variable name (no. of elements)” Ex.: LONG VAR_A(10); Variable name is limited in the same way as the normal macro local variable. Designate the number of elements with the value at the declaration. The names same as other macro local variable names cannot be declared for array variable. Up to 100 elements can be specified as the array variable.

How to use	When the example above is declared, ten elements can be obtained.
	To access each element, designate among “variable name (0) to variable name (9)”.
	Number of elements can be designated with local variable.
	Ex.: To use fifth element of array.
	VAR_B = VAR_A(4);
	Ex.: To designate array elements with local variable.
	VAR_B = VAR_A(VAR_C);
	Ex.: To designate the local variable with a designated element as an argument of function. (When acquiring single data)
	lStat = melGetData(lAddr, lSection, lSubSection, 1, VAR_A(4));
	To designate the local variable for the argument of the function that acquires several data at a time, describe only the variable name.
	Ex.: To designate the array variable as an argument of the function. (When acquiring several data)
	lStat = melGetData(lAddr, lSection, lSubSection, 3, VAR_A);

Available size	Up to 32767 bytes are available in an event.

(Note) This can be used in only melGetData, melSetData, and melGetLumpData.

Global Memory Area

The macro local variable functions handling the global memory area are listed below.

Item	Specification
Memory area creation	GMEM variable = GMEMCREATE ({global area name}, {memory size});
Selection of previously created memory area	GMEM variable = GMEMSELECT ({global area name});
Acquisition of value from memory area CHAR	CHAR variable = GMEMGETCHAR({global area variable name}, {byte offset}); (The offset starts at "0".)
Acquisition of value from memory area SHORT	SHORT variable = GMEMGETSHORT({global area variable name}, {byte offset});
Acquisition of value from memory area LONG	LONG variable = GMEMGETLONG({global area variable name}, {byte offset});
Acquisition of value from memory area DOUBLE	DOUBLE variable = GMEMGETDOUBLE({global area variable name}, {byte offset});
Acquisition of value from memory area STRING	STRING variable = GMEMGETSTRING({global area variable name}, {byte offset});
Entry of a value in memory area CHAR	GMEMSETCHAR ({global area variable name}, {byte offset}, {value});
Entry of a value in memory area SHORT	GMEMSETSHORT({global area variable name}, {byte offset}, {value});
Entry of a value in memory area LONG	GMEMSETLONG ({global area variable name}, {byte offset}, {value});
Entry of a value in memory area DOUBLE	GMEMSETDOUBLE ({global area variable name}, {byte offset}, {value});
Entry of a value in memory area STRING	GMEMSETSTRING({global area variable name}, {byte offset}, {value});
Memory area release	GMEMDELETE ({global area variable name});

Example:
```
GMEM M_ABC;
SHORT VAR_A;
M_ABC = GMEMCREATE("ABC", 1024);
(Hereinafter M_ABC is used as a global area type.)
' Write two bytes 123, starting at the fourth byte of ABC.
GMEMSETSHORT(M_ABC, 4, 123);
' Acquire two bytes from the fourth byte of ABC to VAR_A.
VAR_A = GMEMGETSHORT(M_ABC, 4);
GMEMDelete(M_ABC);
(Releasing global area M_ABC)
```

NOTE

- The global memory area can be created or used in the program after source code generation.

Argument Variable

The argument variable is a fixed-name variable that holds the argument data of the callback function. A list of argument variables and correspondence with the callback function argument are shown below.

Argument of callback function	Macro argument variable
unsigned short usMessage	USMESSAGE
long lLParam	LLPARAM
long lUParam	LUPARAM

Macro reserved word variable

The macro reserved word variable can be used across all projects without declaring the type of variable. There is LONG type and DOUBLE type for the macro reserved word variable. As the macro reserved word variable is common across all projects, always initialize before use.
The below explains the macro reserved variable.

Macro reserved word variable	Explanation
@1 to @99	32bit integer type with a sign
#1 to #99	64bit floating decimal point type

The below explains how to define the macro reserved variable.

Item	Specification
Describing method	Describe the macro reserved word variable as a variable name in the form of "@"＋Number or "#"+Number. Example:@1 = 10; #10 = 100.5;

NC system variable

NC system variable is a variable that can read/write to the system variable for NC.

(1) Undefined variable (#0)

Undefined variable is a variable that is always <empty>. No declaration of the variable type is required. It is used when inputting <empty> into the common variable of NC system variable and inputting "0" into the macro local variable.

NC system variable	Description
#0	Only reading with 64bit floating decimal point type is enabled. Substituting a value to "#0" will cause a command error at error checking. When substituting "#0" into a common variable of NC system variable, it becomes <empty> value. When substituting "#0" into a macro local variable and macro reserved word variable, it becomes "0" if a variable is a integer type, and it become "0.0" if it is a floating decimal type.

The below explains how to describe an undefined variable.

Item	Specification
Description method	Define "#0" as a variable name. Example: #1 = #0;

(Note 1) An <empty> will be handled as same as "0" in an arithmetic expression. Therefore, it will not be used as denominator for division. If it is used as denominator, a running macro will be interrupted because the expression will be treated as a division by 0.

Example:

#100 = #0 ----------#100 = <empty>

#1 = #0; ----------#1 = 0.0

@1 = #0; ----------@1 = 0

#2 = #0 + 1; ---------#2 = 1

@3 = 1 + #0; --------@3 = 1

#4 = #0 * 10; --------#4 = 0

#105 = #0 + #0; ----#105 = 0

#106 = 10 / #0; -----10 / 0 The macro will be interrupted at this line.

(Note 2) An <empty> will be handled as same as "0.0" in a conditional expression. However, when a common variable is compared, it will be different depending on the contents of common variable.

Example:

IF[#1 EQ #0] ----------IF[#1 EQ 0.0]

IF[@1 EQ #0] ---------IF[@1 EQ 0.0]

IF[#0 EQ 0.0] ---------IF[0.0 EQ 0.0]

IF[#100 EQ #0] -------depending on the contents of "#100"

Refer to "Note 3" in "Conditional expression".

(Note 3) If an <empty> is designated to an argument of a function, it will be handled as same as "0.0". However, it will be handled as same as "0" if the type of argument is other than "double".

(2) Common variable

Common variable is a 64bit floating decimal type variable that can read/write to the common variable for NC. It can be used across all projects without declaring the type of variable. The number of variable sets that can be used will differ depending on NC specification (such as parameter setting). If unusable common variable is used, a running macro will be interrupted at the line where it is used.

NC system variable	Description
#100 to #199	Used as 1st part system common variables #100 to #199. These variables cannot handle the common variable #100 to #199 for the 2nd part system and the following. (Note1)
#100100 to #100199 #200100 to #200199 #300100 to #300199 #400100 to #400199 #500100 to #500199 #600100 to #600199 #700100 to #700199 #800100 to #800199	Used as variables for common variables #100 to #199. These variables can handle all of the common variables for all part systems. Common variables #100 to #199 by part system can be handled with "# + part system number * 100000 + variable No". The variables which exceed the valid part systems might become common variable shared among the part systems depending on the setting of "#1316 CrossCom". (Note 1)
#400 to #999	Used as variables for common variables "#400 to #999" shared among all part systems. Some or all of the variables from #500 to #900 might become common variables for each part system depending on the setting of "#1052 MemVAl" and "#1304 V0comN ". If "#1052 MemVal" is set to "1", these become variables to handle common variables #500 to #900 for 1st part system.
#100500 to #100999 #200500 to #200999 #300500 to #300999 #400500 to #400999 #500500 to #500999 #600500 to #600999 #700500 to #700999 #800500 to #800999	When NC parameter "#1052 MemVal" is set to "1", these are variables handling the common variable #500 to #999 for each part system set in "#1304 V0comV". The common variable #500 to #999 for each part system can be handled with "# + part system number * 100000 + variable No". These variables might become common variables shared among all part systems depending on the setting of "#1052 MemVal" and "#1304 V0comN ". If a part system number exceeding the number of valid part systems is designated, a running macro will be interrupted at the line where it is designated.
#900000 to #907399	Used as variables for common variable #900000 to #907399 shared among all part systems.

(Note 1) These variables might become common variable shared among the part systems depending on the setting of "#1052 MemVal" and "#1303 V1comN".

The below shows the NC common variables which read and write by NC parameters when a macro is executed.

NC parameter setting		Macro program to run		
		#100 = 10;	**#500 = 10;**	**#100999 = 10;**
#1316 CrossCom	0	Write "10" to "#100" for 1st part system	Write "10" to "#500" that is common for part systems	Write "10" to "#999" that is common for part systems
#1052 MemVal	0			
#1316 CrossCom	0	Write "10" to "#100" that is common for part systems	Write "10" to "#500" that is common for part systems	Write "10" to "#999" for 1st part system
#1052 MemVal	1 (#1303 V1comN = 10 #1304 V0comN = 10)			
#1316 CrossCom	1	Write "10" to "#100" for 1st part system	Write "10" to "#500" that is common for part systems	Write "10" to "#100999" for 1st part system
#1052 MemVal	Setting invalid			

Parameter setting		Macro program to run		
		#200110 = 10;	**#300520 = 10;**	**#500150 = 10;**
#1316 CrossCom	0	Write "10" to "#110" for 2nd part system	Write "10" to "#520"that is common for part systems	The running macro is interrupted at this line.
#1052 MemVal	0			
#1316 CrossCom	0	Write "10" to "#110" for 2nd part system	Write "10" to "#520" for 3rd part system	The running macro is interrupted at this line.
#1052 MemVal	1 (#1303 V1comN = 10 #1304 V0comN = 10)			
#1316 CrossCom	1	Write "10" to "#200110" for 1st part system	Write "10" to "#520" that is common for part systems	Write "10" to "#500150" that is common for part systems
#1052 MemVal	Setting invalid			

(Note 1) When the parameter "#1316 CrossCom" is set to "0", NC will handle #100100 to #100110 as system variables for reading PLC data, but NC Designer2 will handle them as common variables for 1st part system.

(Note 2) Even if the data protection key 2 is valid, an error will not occur and a value can be written into NC common variable.

Constant

The constant that can be used in the macro program and its description method are specified below.

Item	Description
Decimal constant	-2147483648 to 2147483647
Hexadecimal constant	H0 to HFFFFFFFF
Character string	Enclose the character string with double quotation marks " ". Example: "ABCDE"

(Note) The maximum number of characters for a character string type variable is set to 256, therefore, if more than 256 characters are described in the macro program, the characters in excess are truncated when the macro program is executed.

Operator

The operators that can be used in the macro program are shown below.

Function	Operator	Description example	Function
Substitution	=	B = A	Substitute B with A.
Addition	+	C = A + B	Substitute C with A + B.
Subtraction	-	C = A - B	Substitute C with A - B.
Multiplication	*	C = A * B	Substitute C with A × B.
Division	/	C = A / B	Substitute C with A / B.
Remainder	%	C = A % B	Substitute C with the remainder of A / B.
	MOD	C = A MOD B	
Logical sum (OR)	|	C = A | B	Substitute C with the logical sum of A and B. Possible to substitute every 32 bit.
	OR	C = A OR B	
Logical product (AND)	&	C = A & B	Substitute C with the logical product of A and B. Possible to substitute every 32 bit.
	AND	C = A AND B	
Negation (NOT)	!	B = !A	Substitute B with negation of A.
Exclusive OR (XOR)	^	C = A ^ B	Substitute C with exclusive OR of A and B. Possible to substitute every 32 bit.
	XOR	C = A XOR B	
1's complement	~	B = ~A	Substitute B with A's complement.
Arithmetic bit shift (left)	<<	C = A << B	Substitute C with A shifted to the left by B bits.
Arithmetic bit shift (right)	>>	C = A >> B	Substitute C with A shifted to the right by B bits.

(Note) MOD, OR, AND, and XOR are reserved words. Put a space both before and after these words when using.

Multiple operators can be combined.
Example:
A=(B+C)*(D+E/2);
A=[B OR C]+(D MOD E);

Priority among operators is shown below.

Priority	Operator
Higher	(,) , [,]
↑	~ , !
	* , / , % , MOD
	+ , -
	<< , >>
	& , AND
	^ , XOR
↓	| , OR
Lower	=

Operation function

The operators that can be used in the macro program are shown below.

Function	Operation command	Description example	Function
Sine(Unit: degree)	SIN	B = SIN[A]	Substitute the result of sine A into B
Cosine(Unit: degree)	COS	B = COS[A]	Substitute the result of cosine A into B
Tangent(Unit: degree)	TAN	B = TAN[A]	Substitute the result of tangent A into B (Note 2)
Arcsine	ASIN	B = ASIN[A]	Substitute the result of arcsine A into B (Note 2)
Arccosine	ACOS	B = ACOS[A]	Substitute the result of arccosine A into B (Note 2)
Arctangent	ATAN	B = ATAN[A]	Substitute the result of arctangent A into B
	ATN	B = ATN[A]	
Square root	SQRT	B = SQRT[A]	Substitute the result of square root A into B
	SQR	B = SQR[A]	
Absolute value	ABS	B = ABS[A]	Substitute the result of absolute value A into B
Conversion from BCD to BIN	BIN	B = BIN[A]	Substitute the result of BIN for BCD A into B (Note 1)
Conversion from BIN to BCD	BCD	B = BCD[A]	Substitute the result of BCD for BINARY A into B (Note 1)
Rounding off	ROUND	B = ROUND[A]	Substitute the rounded value of A into B
Discard fractions less than 1	FIX	B = FIX[A]	Substitute the result of A rounded down to an integer into B
Add for fractions less than 1	FUP	B = FUP[A]	Substitute the result of A rounded up to an integer into B
Natural logarithm	LN	B = LN[A]	Substitute the result of natural logarithm A into B (Note 2)
Exponent with e (=2.718 .....) as bottom	EXP	B = EXP[A]	Substitute the result of EXP A into B

- The decimal point accuracy is guaranteed for up to 8 places of decimals of 16 digits decimal.

(Note 1) The followings are the precautions when using BIN and BCD.

- If the decimal point is used, they are processed as an integer.
- If a negative number is used, it is processed as an integer with BCD but a running macro will be interrupted with BIN.
- If an <empty> is used, a running macro will be interrupted with both BIN and BCD.

(Note 2) A running macro will be interrupted because the operation is impossible when:

- The value over 1 or less than -1 is designated with ASIN and ACOS.
- 90.270 etc. is designated with TAN.
- 0, a negative number or <empty> is designated with LN.

(Note 3) The degree of ASIN varies according to the parameter value "#1273/bit0 Switch ASIN calculation results range".

#1273/Bit0 0 : -90° to 90°
1: 270° to 90°

Example)

Sine SIN	#501 = SIN[60]; #502 = SIN[60.]; #503 = 1000*SIN[60]; #504 = 1000*SIN[60.]; #505 = 1000.*SIN[60]; #506 = 1000.*SIN[60.];	#501 0.866 #502 0.866 #503 866.025 #504 866.025 #505 866.025 #506 866.025
Cosine COS	#511 = COS[45]; #512 = COS[45.]; #513 = 1000*COS[45]; #514 = 1000*COS[45.]; #515 = 1000.*COS[45]; #516 = 1000.*COS[45.];	#511 0.707 #512 0.707 #513 707.107 #514 707.107 #515 707.107 #516 707.107
Tangent TAN	#521 = TAN[60]; #522 = TAN[60.]; #523 = 1000*TAN[60]; #524 = 1000*TAN[60.]; #525 = 1000.*TAN[60]; #526 = 1000.*TAN[60.];	#521 1.732 #522 1.732 #523 1732.051 #524 1732.051 #525 1732.051 #526 1732.051
Arcsine ASIN	#531 = ASIN[100.500/201]; #532 = ASIN[100.500/201.]; #533 = ASIN[0.500]; #534 = ASIN[-0.500];	#531 30.000 #532 30.000 #533 30.000 #534 -30.000 (Note) When "1" is set to "#1273/bit0", the result is "#534 = 330°".
Arccosine ACOS	#541 = ACOS[100/141.4213]; #542 = ACOS[100./141.4213];	#541 45.000 #542 45.000
Arctangent ATAN ATN	#551 = ATAN[173205/100000]; #552 = ATAN[173205/100000.]; #553 = ATAN[173.205/100]; #554 = ATAN[173.205/100.]; #555 = ATN[1.73205];	#551 60.000 #552 60.000 #553 60.000 #554 60.000 #555 60.000
Square root SQRT SQR	#561 = SQRT[1000] #562 = SQRT[1000.]; #563 = SQR[10.*10.+20.+20.];	#561 31.623 #562 31.623 #563 22.360
Absolute value ABS	#571 = ABS[-1000]; #572 = ABS[50-100];	#571 1000.000 #572 50.000
BIN BCD	#581 = BIN[100]; #582 = BCD[100];	#581 64 #582 256
Rounding off ROUND	#591 = ROUND[14/3]; #592 = ROUND[14./3]; #593 = ROUND[14/3.]; #594 = ROUND[14./3.]; #595 = ROUND[-14/3]; #596 = ROUND[-14./3]; #597 = ROUND[-14/3.]; #598 = ROUND[-14./3];	#591 5 #592 5 #593 5 #594 5 #595 -5 #596 -5 #597 -5 #598 -5
Discard fractions less than 1 FIX	#601 = FIX[14/3]; #602 = FIX[14./3]; #603 = FIX[-14/3.] #604 = FIX[-14./3.];	#601 4.000 #602 4.000 #603 -4.000 #604 -4.000
Add for fractions less than 1 FUP	#611 = FUP[14./3.]; #612 = FUP[14/3.]; #613 = FUP[-14./3]; #614 = FUP[-14/3];	#611 5.000 #612 5.000 #613 -5.000 #614 -5.000
Natural logarithm LN	#621 = LN[5]; #622 = LN[0.5]; #623 = LN[-5.0]	#621 1.609 #622 -0.693 A running macro will be interrupted.
Exponent	#631 = EXP[2]; #632 = EXP[1]; #633 = EXP[-2];	#631 7.389 #632 2.718 #633 0.135

Custom API Function

(1) Setting address

For custom API function, designate NCs, part systems, or axes to operate to the address of argument.
Use the following macros for the address.

Name of function/constants	Description of function
ADR_MACHINE(n)	Designates NC (n: 1 and other numbers.)
ADR_SYSTEM(n)	Designates part systems (n: 1 and other numbers.)
ADR_GROUND(0)/ADR_GROUND(1)	Designates foreground/background.
ADR_BASE_SYSTEM	Designates basic part system.
ADR_CROSS_CURRENT	Designates current part system in cross.
ADR_FORCE_WRITE	Forced writing mode.

- Combine function and constants with the theory sum (|) and set.
- The designation for necessary address differs depending on the function used. Refer to "6. NC Data Access Function".
- Specify 1 for designating NC. ADR_MACHINE(1) can be omitted.
- Designate part systems according to specification of NC being used.

Examples:

Ex. 1) Specify NC No.1 and background

```
ADR_MACHINE(1) | ADR_GROUND(1)
```

Ex. 2) Specify NC No. 1 and basic second part system

```
ADR_MACHINE(1) | ADR_BASE_SYSTEM | ADR_SYSTEM(2)
```

Ex. 3) Specify NC No. 1 and forced writing mode

```
ADR_MACHINE(1) | ADR_FORCE_WRITE
```

(2) Error judgment

This function judges the execution results (normal/abnormal) of custom API functions.

Name of function	Description of function
RetvIsError(lStatus)	Judges the execution results of custom API functions.

- Designate the execution results of custom API functions to lStatus.
- When judging normal, returns “0”.
- When judging abnormal, returns “1”.

Ex.: Execute the function "melGetData()" and judge the result.

```
LONG lStatus;
LONG lData;
LONG lAddress;

'Acquire data
lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
lStatus = melGetData(lAddress, 127, 2002, H1, lData);
IF(RetvIsError(lStatus))
   'Error when executing
ENDIF
```

String Operation Functions

The following string operation functions are available in the macro program.

String operation function	Command	Description example	Function
String concatenation	strcat	C = strcat(A,B)	Concatenates B to A.
String comparison	strcmp	n = strcmp(A,B)	Compares A with B.
String copy	strcpy	C = strcpy(A,B)	Copies B to A.
Get string length	strlen	n = strlen(A)	Gets the number of characters in A.
String concatenation (Number of characters specified)	strncat	C = strncat(A,B, n)	Concatenates B to A by n characters.
String comparison (Number of characters specified)	strncmp	n = strncmp(A,B, n)	Compares A with B by n characters.
String copy (Number of characters specified)	strncpy	C = strncpy(A,B, n)	Copies B to A by n characters.
Character search	strchr	n2 = strchr(A,n1)	Searches for the character specified in n1 from the beginning of A, and assigns the detection position to n2 (for n1, specify an ASCII code).
Character search	strrchr	n2 = strrchr(A,n1)	Searches for the character specified in n1 from the end of A, and assigns the detection position to n2 (for n1, specify an ASCII code).
Character group search	strpbrk	n = strpbrk(A,B)	Searches for any character included in the character group of B from the beginning of A, and assigns the detection position to n.
String search	strstr	n = strstr(A,B)	Searches for a character string of B from the beginning of A, and assigns the detection position to n.
Character group search	strspn	n = strspn(A,B)	Searches for any character not included in the character group of B from the beginning of A, and assigns the detection position to n.
Character group search	strcspn	n = strcspn(A,B)	Searches for any character included in the character group of B from the beginning of A, and assigns the detection position to n.

(Note1) The cautions on using string operation functions are as follows.

- The available characters are single-byte alphanumeric characters, including symbols.
- A string type variable, char type array variable, or string constant can be specified for a character string type argument.
  A string constant cannot be specified for the first argument of strcat, strcpy, strncat, or strncpy.
- Each string operation function handles up to the element count (up to 100 characters) in the char type array variable declaration as a character string.
  An element with value = 0 is recognized as the character string terminator.
  Example: When storing character string "ABC" in the char type array variable with element count 100:

```
CHAR chData(100);
  chData(0) = H41;   ' "A"
  chData(1) = H42;   ' "B"
  chData(2) = H43;   ' "C"
  chData(3) = H0;    ' Character string terminator
```

- The string comparison functions (strcmp and strncmp) judge the ASCII code sequence from the beginning of a character string.
  Example: When string A: "AA" and string B: "AB": String A is judged to be less than string B, and "< 0" is returned.
  Example: When character string A: "AB" and character string B: "AAA": Character string A is judged to be greater than character string B, and "> 0" is returned.
  Example: When character string A: "ABC" and character string B: "ABC": Character string A is judged to be equal to character string B, and "0" is returned.
- The detection position of each character string search function (strchr, strrchr, strpbrk, strstr, strspn, and strcspn) is returned with the value that is returned with 0 or greater value.

Type conversion

The following type conversion commands are available in the macro program.
A number read as a character string by type conversion can be handled as a numeric value of the integer type or floating point type.
Also, a numeric value of the integer type or floating point type can be converted into the character string type.

Type conversion command	Example	Description
ATOL	B = ATOL(A)	Converts a character string type variable A to the signed 32-bit integer type and substitute it into B.
ATOF	B = ATOF(A)	Convert a character string type variable A to the signed 64-bit floating point type and substitute it into B.
LTOA	A = LTOA(B,n)	Converts variable B of the signed 32-bit integer type to the character string type, and assigns it to variable A. Specify the conversion method based on the radix (2 to 16) of n.
FTOA	A = FTOA(B,n)	Converts variable B of the signed 64-bit floating point type to the character string type, and assigns it to variable A. You can specify the number of decimal places (0 to 10) for n to convert the data type.

(Note 1) When variable A includes characters other than a numeric value, decimal point, or positive and negative symbols, the numeric value before those characters is valid.

```
Example: A=”1a2”;     B=ATOL(A);   The value of B is "1".
         A=”a12”;     B=ATOL(A);   The value of B is "0".
         A=”-10”;     B=ATOL(A);   The value of B is "-10".
         A=”--10”;    B=ATOL(A);   The value of B is "0".
         A=”10”;      B=ATOL(A);   The value of B is "10".
         A=”12 3”;    B=ATOL(A);   The value of B is "12".
         A=”12.3”;    B=ATOF(A);   The value of B is "12.3".
         A=”1.2E1”;   B=ATOF(A);   The value of B is "12".
```

(Note 2) ATOL cannot convert a variable including the decimal point or index. Such a variable is handled in the same way as for other characters.

(Note 3) In LTOA, operations are performed as shown below.

- If the specified radix is outside the allowable range, the value is handled as radix 10.
- If alphabetical characters are included in the converted result, they are converted in uppercase.
- The converted character string is zero-suppressed.

```
Example: B=1000;  A=LTOA(B, 10);  The value of A is "1000".                          Converted in decimal notation.
         B=1000;  A=LTOA(B, 16);  The value of A is "3E8".                           Converted in hexadecimal notation.
         B=1000;  A=LTOA(B, 2);   The value of A is "1111101000".                    Converted in binary notation.
         B=1000;  A=LTOA(B, 10);  The value of A is "-1000".                         Converted in decimal notation.
         B=1000;  A=LTOA(B, 16);  The value of A is "FFFFFC18".                      Converted in hexadecimal notation.
         B=1000;  A=LTOA(B, 2);   The value of A is "11111111111111111111110000011000".
                                  Converted in binary notation.
```

(Note 4) In FTOA, the value is converted up to the specified number of decimal places, and the subsequent places are rounded off.
It is not converted into an exponential notation. If the number of significant digits is less than the number of decimal places, 0s are embedded.
When the number of decimal places exceeds 10, the value is handled as a 10-digit value. However, a negative value causes a conversion failure.

Example: B=1.23456; A=FTOA(B, 3); The value of A is "1.235". Indicated up to three decimal places (the fourth place is rounded off).
B=1.23456; A=FTOA(B, 10); The value of A is "1.2345600000". Indicated up to 10 decimal places (insufficient places are embedded with "0").

(Note 5) If the value exceeds the allowable range of the 32-bit integer type (-2147483648 to 2147483647) or the allowable range of the 64-bit floating point type (real number of 15 significant digits), an incorrect value is assigned.

Conditional Branch

Conditional branch commands are used in the macro program to branch according to the judgment of a condition. The conditional branch commands that can be used are listed below.

Item	Description

IF
ELSEIF
ELSE
ENDIF
	Enclose the condition expression with brackets () or [] following IF or ELSEIF. Add ENDIF without fail at the end. Up to 15 nesting levels are allowed.
	Example:
	IF(a==0)
	  GCSTextboxSetString(-1,"GTextBox00001","000");
	ELSEIF[a==1]
	  GCSTextboxSetString(-1,"GTextBox00001","001");
	ELSE
	  GCSTextboxSetString(-1,"GTextBox00001","111");
	ENDIF

Condition Expression

The condition expressions used for condition judgment in the IF statement are described in the following way.

Condition expression	Establishment condition
A == B	Established if A is equal to B.
A EQ B	
A > B	Established if A is larger than B.
A GT B	
A >= B	Established if A is equal to or larger than B.
A GE B	
A < B	Established if A is smaller than B.
A LT B	
A <= B	Established if A is equal to or smaller than B.
A LE B	
A != B	Established if A is not B.
A NE B	
A && B	Established if both condition A and B are established (logical product).
A || B	Established if either condition A or B is established (logical sum).

(Note 1) EQ, GT, GE, LT, LE and NE are reserved words. When using these words, make sure to enter a space before and after them.

(Note 2) When comparing between an integer value and a floating point value, a floating point value will be rounded down to an integer and then compared.

EX) When "@1 = 1" and "#1 = 1.1", they are compared as shown below since #1 is rounded down to an integer 1.

@1 == #1 … 1=1 Established

@1 != #1 … 1≠1 Not established

@1 <= #1 … 1≦1 Established

@1 >= #1 … 1≧1 Established

@1 < #1 … 1＜1 Not established

@1 > #1 … 1＞1 Not established

(Note 3) When an undefined variable (#) is used in a conditional expression such as "EQ", "==", "NE", "!=", it will be a compare condition with <empty>, however, if other reserved words are used for a conditional expression, it will be a compare condition with "0".

When #100 = <empty>		When #100 = 0	
#100 EQ #0	<Empty>=<Empty> Established	#100 EQ #0	0=<Empty> Not established
#100 NE #0	<Empty>≠<Empty> Not established	#100 NE #0	0≠<Empty> Established
#100 GE #0	<Empty>≧0 Established	#100 GE #0	0≧0 Established
#100 GT #0	<Empty> > 0 Not established	#100 GT #0	0>0 Not established
#100 LE #0	<Empty>≦0 Established	#100 LE #0	0≦0 Established
#100 LT #0	<Empty> < 0 Not established	#100 LT #0	0<0 Not established

Program Branch

In the macro program, program branch commands are used to control the flow of program execution. The following program branch commands can be used.

Command	Description
FOR, NEXT	Repeat the program between FOR and NEXT by the designated number of times. Up to eight nesting levels are allowed. Example: FOR(5) VAR_A = VAR_A + 1; NEXT

FOR, BREAK	Interrupt program repetition.
	Example: FOR(3)
	VAR_B= VAR_B + 1;
	IF(VAR_B > 10 && VAR_A == 1)
	BREAK;
	ENDIF
	NEXT

FOR, CONTINUE	Repeat a program.
	Example: FOR(3)
	IF(VAR_C == 1)
	CONTINUE;
	ENDIF
	VAR_C = VAR_C + 1;
	NEXT

WHILE, DOi, ENDi (i is the identification No. for "WHILE".)	While the conditional expression is established, a program will be repeated up to the END which has the same number as the identification number following D0. Add ";" (a semicolon) at the end of DOi and ENDi. Example: WHILE (VAR_D == 1) DO1; VAR_D = VAR_D + 1; END1; - The same identification number can be used a number of times. - 1 to 127 is available for the identification number and there is no rule for the description order. - Up to 27 nesting level can be used. - BREAK and CONTINUE commands are available between WHILE[conditional expression] DOi and ENDi. - ENDi must be designated after WHILE[conditional expression]D01;. - WHILE[conditional expression]D01 and ENDi must correspond on 1:1 (pairing) basis in a same macro. - Two WHILE[conditional expression] D01 must not intersect. - An infinite loop cannot be described in a style like WHILE(1).

Command	Description
GOTO n (n = sequence No.)	Setting a sequence No. after GOTO allows to branch to the sequence No. commanded in a macro. Example: GOTO 100; #100 = #10; N100; #100 = #20; - Description of IF[conditional expression]G0T0n; is same as IF[conditional expression] to ENDIF. IF ELSE, ELSEIF or ENDIF is added after G0T0, an error will occur when checking error. - GOTO statement cannot be used with ELSE, ELSEIF. - It is possible to branch out from WHILE[conditional expression] D0i with GOTO statement. - It is not possible to branch into WHILE[conditional expression] D0i with GOTO statement. - When executing a macro with a sequence No. described with # variable or @ variable but a sequence No. which corresponds to the # variable or @ variable does not exist, the running macro will be interrupted at where the GOTO is commanded. - When a variable or expression (such as GOTO#100) is set as a sequence No. but the sequence No. does not exist, the running macro will be interrupted.

The sequence No. is used as a label to show the destination of program branching command. The below explains the format of sequence No.

N	No.
Always add at the head	1 to 99999 in decimal

- When only writing the sequence No., make sure to add a semicolon ";" at the end.
  Example:N100;
- It is possible to add a processing after the sequence No. In this case, a space will be required between the sequence No. and a processing.
  Example:N100 #100=10;
- The sequence No. after the GOTO statement always needs to be added in the same macro. If not, "GOTO sentence error" error will occur.
- When the sequence number of the branch destination is searched in GOTO statement, the search is conducted from the GOTO statement to a macro end ($END). If the sequence No. is located before the GOTO statement, it cannot be searched. If there is multiple same sequence No., it branches to the first found sequence No. with this order.
- An error will occur when designating the common variable to the sequence No.
  EX:N#200

(Note) As the sequence No. format can be written in a same program with a variable name definition, an error will occur if the sequence No. whose name is the same as the variable name is defined.

Example:

```
SHORT N1000;
N1000=100;
N1000;      <-- This definition is a grammatical error.
IF[#100 EQ #1] GOTO N1000
```

WHILE statement description method

(1) Same identification No. can be used any number of times.

```
$GBasicControl00000-OnCreate
  WHILE[conditional expression] DO1;
  〉
  END1;
  WHILE[conditional expression] DO1;
  〉
  END1;
$End
```

(2) Any order may be used for the WHILE to DOi identification No.

```
$GBasicControl00001-OnKeyPress
  WHILE[conditional expression] DO3;
  〉
  END3;
  WHILE[conditional expression] DO1;
  〉
  END1;
$End
```

(3) Up to 27 nesting levels can be used.

```
$GBasicControl00002-OnKeyRelease
  WHILE[conditional expression] DO1;
    〉
    WHILE[conditional expression] DO2;
      〉
      WHILE[conditional expression] DO27;
      〉
      END27;
      〉
    END2;
    〉
  END1;
$End
```

(Note) For nesting, an identification No. which has been used once cannot be used again.

(4) Nesting levels cannot exceed 27.

```
$GBasicControl00003-OnPress
  WHILE[conditional expression] DO1;
    〉
    WHILE[conditional expression] DO2;
      〉
      WHILE[conditional expression] DO28;   Impossible
      〉
      END28;
      〉
    END2;
    〉
  END1;
$End
```

(Note) "Too many WHILE command" error will occur.

(5) ENDi must be designated after WHILE[conditional expression] DOi.

```
$GBasicControl00004-OnRelease
  END1;                                 Impossible
    〉
  WHILE[conditional expression] DO1;
$End
```

(Note) "WHILE DO sentence error" will occur.

(6) WHILE [conditional expression] DOi and ENDi must correspond on 1:1 (pairing) basis in a same program.

```
$GBasicControl00005-OnClick
  WHILE[conditional expression] DO1;
    〉
  WHILE[conditional expression] DO1;    Impossible
    〉
  END1;
$End
```

(Note) "WHILE DO sentence error" will occur.

(7) Two WHILE[conditional expression] DOi must not overlap.

```
$GBasicControl00006-OnDraw
  WHILE[conditional expression] DO1;
  〉
  WHILE[conditional expression] DO2;   Impossible
  〉
  END1;
  〉
  END2;
$End
```

(Note)"WHILE DO sentence error" will occur.

(8) It is not allowed to abbreviate WHILE[conditional expression], and to describe only D0 to END.

```
$GBasicControl00007-OnSetFocus
  DO1;                Impossible
  〉
  END1;
$End
```

(9) An infinite loop cannot be described in a style like WHILE(1).

```
$GBasicControl00008-OnKillFocus
  WHILE(1) DO1;       Impossible
  〉
  END1;
$End
```

(10) BREAK and CONTINUE commands can be used between DOn and END in WHILE[conditional expression].

```
$GBasicControl00009-OnKillFocus
  WHILE[conditional expression] DO1;
    〉
    BREAK;
    〉
  END1;
$End
```

GOTO statement description method

(1) When a condition is established with IF[conditional expression] GOTO n;, it will be branched. In this case, no need to add ENDIF.

```
$GTextBox00000-OnKeyPress
  IF(#1 == #2) GOTO 100;
  #3 = #3 + 1;
  N99;
  #10 = #5
  N100;
  #1 =#10;
$End
```

(2) It is possible to branch out from WHILE[conditional expression] D0i with GOTO statement.

```
$GTextBox00001-OnKeyPress
  WHILE[conditional expression] DO1;
      IF[#50 != #100] GOTO 100;
  END1;
  〉
  N100;
  #150 = 1.5;
$End
```

(3) It is not possible to branch into WHILE[conditional expression] D0i with GOTO statement.

```
$GTextBox00002-OnKeyPress
   IF[conditional expression] GOTO 100;
  WHILE[conditional expression] DO1;
    N100;
  END1;
$End
```

Impossible

```
$GTextBox00003-OnKeyPress
  WHILE[conditional expression] DO1;
    IF[conditional expression] GOTO 100;
  END1;
  WHILE[conditional expression] DO1;
    N100;
  END1;
$End
```

Impossible

```
$GTextBox00004-OnKeyPress
  WHILE[conditional expression] DO1;
    WHILE[conditional expression] DO2;
      IF[conditional expression] GOTO 100;
    END2;
    N100;
  END1;
$End
```

Impossible

(Note) "GOTO statemen inconsistent" error will occur.

(4) When the sequence No. of the branch destination is searched, the search is conducted from the GOTO statement to a macro end ($End).

```
$GTextBox00005-OnKeyPress
  〉
  N100;
  #100 = 100.1;
  IF[#10 == #20] GOTO 10;
  ELSE GOTO 100;
  ENDIF
  N10;
  #10 = 10.1;
  〉
$END
```

Impossible

(Note) If the sequence number of the branch destination is located before the GOTO statement, "WHILE DO sentence error" will occur.

(5) If there is multiple same sequence No., it branches to the first found sequence No.

```
$GTextBox00006-OnKeyPress
  〉
  IF[#110 == #120] GOTO 100;
  〉
  N100 @1 = 1;
  〉
  N100 #10 = 10.01;
  〉
$END
```

5. Function Details

The macro functions used for the macro programming are explained in this section.

5.1 Normal Command

SetCallback	Select call-back processing execution
Syntax	SetCallBack(nIsExec);
Argument	(i) LONG nIsExec : Whether or not to execute call-back processing
Return value	None
Details	Set whether or not to execute call-back processing. Set nIsExect to the value below. 0: Does not execute call-back processing after macro operation. 1: Executes call-back processing after macro operation.
Example	Set so that call-back processing is executed after macro operations. SetCallBack(1);

GMEMCreate	Create global memory
Syntax	GMEMCreate(strGmemName, nSize);
Argument	(i) STRING strGmemName : global memory name (i) LONG nSize : memory size
Return value	Global memory
Details	Creates a global memory by specifying a corresponding name.
Example	Creates a global memory "mem" that consists of 100 bytes of global memory named "ABC". GMEM mem; mem = GMEMCreate("ABC", 100);

GMEMSelect	Select global memory
Syntax	GMEMSelect(strGmemName);
Argument	(i) STRING strGmemName : global memory name
Return value	Global memory
Details	Gets the global memory with the specified name. Needs to specify the name of the global memory that has already been created in strGmemName.
Example	Gets the global memory named "ABC". GMEM mem; mem = GMEMSelect("ABC");

GMEMDelete	Delete global memory
Syntax	GMEMDelete(gmBuffer);
Argument	(i) GMEM gmBuffer : global memory
Return value	None
Details	Deletes the specified global memory.
Example	Deletes the global memory named "ABC". GMEM mem; mem = GMEMSelect ("ABC"); GMEMDelete(mem);

GMEMSetChar	Set 8bit integer value to global memory
Syntax	GMEMSetChar(gmBuffer, nOffset, nData);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position (i) CHAR nData : setting value
Return value	None
Details	Sets an 8bit integer value in the nOffset byte from the beginning of the specified global memory.
Example	Sets 8bit integer (10) in the "ABC" global memory's 4th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetChar(mem, 4, 10);

GMEMGetChar	Get 8bit integer value from global memory
Syntax	GMEMGetChar(gmBuffer, nOffset);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position
Return value	8bit integer value
Details	Gets an 8bit integer value in the nOffset byte from the beginning of the specified global memory.
Example	Gets 8bit integer value in the "ABC" global memory's 4th byte position from the top. CHAR Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetChar(mem, 4);

GMEMSetShort	Set 16bit integer value in global memory
Syntax	GMEMSetShort(gmBuffer, nOffset, nData);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position (i) SHORT nData : setting value
Return value	None
Details	Sets a 16bit integer value in the nOffset byte from the beginning of the specified global memory.
Example	Sets a 16bit integer (1000) in the "ABC" global memory's 4th byte position from the beginning. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetShort(mem, 4, 1000);

GMEMGetShort	Get 16bit integer value from global memory
Syntax	GMEMGetShort(gmBuffer, nOffset);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position
Return value	16bit integer value
Details	Gets a 16bit integer value in the nOffset byte from the beginning of the specified global memory.
Example	Gets 16bit integer value in the "ABC" global memory's 4th byte position from the top. SHORT Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetShort(mem, 4);

GMEMSetLong	Set 32bit integer value to global memory
Syntax	GMEMSetLong(gmBuffer, nOffset, nData);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position (i) LONG nData : setting value
Return value	None
Details	Sets a 32bit integer value in the nOffset byte from the beginning of the specified global memory.
Example	Sets 32bit integer value (100000) in the "ABC" global memory's 4th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetLong(mem, 4, 100000);

GMEMGetLong	Get 32bit integer value from global memory
Syntax	GMEMGetLong(gmBuffer, nOffset);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position
Return value	32bit integer value
Details	Gets 32bit integer value in the nOffset byte data from the beginning of the specified global memory.
Example	Gets a 32bit integer in the "ABC" global memory's 4th byte position from the beginning. LONG Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetLong(mem, 4);

GMEMSetDouble	Set 64bit floating point value in global memory
Syntax	GMEMSetDouble(gmBuffer, nOffset, fData);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position (i) DOUBLE fData : setting value
Return value	None
Details	Sets a 64bit floating point value in the nOffset byte from the beginning of the specified global memory.
Example	Sets 64bit floating point value (0.1234) in the "ABC" global memory's 8th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetDouble(mem, 8, 0.1234);

GMEMGetDouble	Get 64bit floating poing value from global memory
Syntax	GMEMGetDouble(gmBuffer, nOffset);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position
Return value	64bit floating point value
Details	Gets 64bit floating point value in the nOffset byte data from the beginning of the specified global memory.
Example	Gets a 64bit floating point value in the "ABC" global memory's 8th byte position from the beginning. DOUBLE Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetDouble(mem, 8);

GMEMSetString	Set character string in global memory
Syntax	GMEMSetString(gmBuffer, nOffset, strData);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position (i) STRING strData : character string to be set
Return value	None
Details	Sets a character string in the nOffset byte from the beginning of the specified global memory.
Example	Sets a character string ("adcdef") in the "ABC" global memory's 4th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetString(mem, 4, "abcdef");

GMEMGetString	Get character string from global memory
Syntax	GMEMGetString(gmBuffer, nOffset);
Argument	(i) GMEM gmBuffer : global memory (i) LONG nOffset : offset position
Return value	Character string
Details	Gets the character string in the nOffset byte from the beginning of the specified global memory.
Example	Gets the character string in the "ABC" global memory's 4th byte position from the top. STRING Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetString(mem, 4);

ExecDfc	Execute user definition process
Syntax	ExecDfc(dllName, funcName, dfcArg);
Argument	(i) STRING dllName : DLL name (i) STRING funcName : DFC function name (i) GMEM dfcArg : DFC argument global memory
Return value	None
Details	Executes upon passing the argument information to the function contained in DLL of the specified DLL name. (DLL name and DLL file name are the same, except that the latter has no extension.)
Example	Executes the function after passing a global memory "mem" as an argument to the function "dfcFunc01" which is included in the "DFC.dll" DLL. GMEM mem; mem = GMEMSelect("ABC"); ExecDfc("DFC", "dfcFunc01", mem);

NormalMethod	Call normal process of control
Syntax	NormalMethod ();
Argument	None
Return value	Return value of normal process of control.
Details	Carries out normal process of control.
Example	Calls normal process of control. LONG lRet; lRet = NormalMethod (); Return(lRet);

GCSGetLastPanelNumber	Get last displayed screen No.
Syntax	GCSGetLastPanelNumber (nFunctionNo);
Argument	(i) LONG nFunctionNo : function No.
Return value	Screen No.
Details	Displays the previously displayed custom screen when the screen is changed using a function key. Set nFunctionNo to one of the following values. 0: SFP key 1: F0 key 2: Window display key 3: Window selection key
Example	Gets the last displayed screen No. whose function No. is 0 LONG _lPanelNumber = GCSGetLastPanelNumber(0) GCSGEShowPanel(_lPanelNumber);

GCSUser	User process
Syntax	GCSUser(nWindowNo, nFunctionNo, gmParam);
Argument	(i) LONG nWindowNo : screen No. (i) LONG nFunctionNo : function No. (i/o)GMEM gmParam : setting value global memory
Return value	1: Succeed 0: Fail
Details	Executes the User function that corresponds to the function No. of the specified screen No.
Example	Executes the User function with the screen No. of 1027 and the function No. of 4193. GMEM mem; mem = GMEMCreate("TESTMEM", 70); GCSUser (1027, 4193, mem); GMEMDelete(mem);

GCSKeyPress	Key press process
Syntax	GCSKeyPress(nWindowNo, nKeyCode, lKeyStatus);
Argument	(i) LONG nWindowNo : screen No. (i) LONG nKeyCode : virtual key code (i) LONG lKeyStatus : key status
Return value	1: Succeed 0: Fail
Details	Executes the KeyPress function that corresponds to the function No. of the specified screen No.
Example	Executes the KeyPress function of the screen No. 1. GCSKeyPress (1, LLPARAM, LUPARAM);

GCSResourceLoadString	Get ID character string specified from resource data
Syntax	GCSResourceLoadString(usID, pString);
Argument	(i) LONG usID : character string resource ID (o) STRING pString : character string to be acquired
Return value	Getting result 0 : Abnormal 1 : Normal
Details	Gets the resource character string which has the specified ID.
Example	Sets the character strings at and subsequent to nOffset in the character string table for the GNXMenu00000 1st to 9th menu buttons in the self screen. SHORT nOffset ; 'Offset (0-) from the beginning of the character string table SHORT nMenuLoop ; STRING strMenu ; nOffset = 1 ; nMenuLoop = 1 ; FOR (9) GCSResourceLoadString((nOffset + nMenuLoop), strMenu) ; GCSMenuSetMenuButtonName_one(-1, "GNXMenu00000", 1, nMenuLoop, strMenu) ; nMenuLoop = nMenuLoop + 1 ; NEXT

GCSGetStandardColor	Acquiring the color information according to the theme color for the specified display part

Syntax	GCSGetStandardColor (usPartsKind);
Argument	(i) LONG usPartsKind: Kind of part color to be acquired
Return value	The color information (RGB888) according to the theme color
Details	Acquires the each part color used on the standard screen as the color information according to the theme color. Specify either of the following values to usPartsKind (kind of part color). When the color information cannot be acquired, -1 is returned to the return value. 0: Window background color 1: Window frame (inside) 2: Window title frame (upper) 3: Window title frame (lower) 4: Window title gradation (start) 5: Window title gradation (end) 6: Value part normal background color (active area) 7: Value part normal background color (inactive area) 8: Label background color 9: Cursor color 10: Scroll bar - bar color 11: Scroll bar - button color 12: Scroll bar - background color 13: Scroll bar - pinch color 14: Background color during switch-ON state 15: Entry area - background color at the time of disabled state

Example	Sets the GTextBox00000 normal background color in the self screen to the value part normal background color (active area) according to the parameter (＃11060 Screen theme color).
	
	$GTextBox00000-OnCreate
	LONG _gcColor;
	'Get the color information according to the theme color (value part normal background color).
	_gcColor = GCSGetStandardColor(6);
	IF(_gcColor != -1)
	'Set the acquired color information to the normal background color for the text box.
	GCSTextboxSetNormalColor(-1, "GTextBox00000", _gcColor);

5.2 Control Common Command

GCSSetBlinkEnableStatus	Set blink operation
Syntax	GCSSetBlinkEnableStatus(nWindowNo, strName, ucEnable);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucEnable : blink operation setting
Return value	None
Details	Sets the object blink operation. One of the following values is set for ucEnable: 0: disable 1: enable
Example	Sets the GButton00000 blink operation in the self screen to 1. GCSSetBlinkEnableStatus(-1, "GButton00000", 1);

GCSGetBlinkEnableStatus	Get blink operation
Syntax	GCSGetBlinkEnableStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : (Specify -1 fors self screen.) (i) STRING strName : control name
Return value	Setup of blink operation 0: disable 1: enable
Details	Gets the object blink operation.
Example	Gets the GButton00000 blink operation in the self screen in Stat. LONG Stat; Stat = GCSGetBlinkEnableStatus(-1, "GButton00000");

GCSSetBlinkStatus	Set blink status
Syntax	GCSSetBlinkStatus(nWindowNo, strName, ucBlinkStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucBlinkStatus : blink status
Return value	None
Details	Sets the object blinking status for the object. One of the following values is set for ucBlinkStatus: 0: OFF status 1: ON status
Example	Sets the GButton00000 blink operation in the self screen to 1.

GCSGetBlinkStatus	Get blink status
Syntax	GCSGetBlinkStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Blink status 0: OFF status 1: ON status
Details	Gets the object blinking status for the object.
Example	Gets the GButton00000 blink operation in the self screen in Stat. LONG Stat; Stat = GCSGetBlinkStatus(-1, "GButton00000");

GCSSetBlinkType	Set blink type
Syntax	GCSSetBlinkType(nWindowNo, strName, ucBlinkType);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucBlinkType : blink type
Return value	None
Details	Sets the object blink type. One of the following values is set for ucBlinkType: 0: display/non-display of character string 1: change character color 2: change whole color
Example	Sets the GButton00000 blink operation in the self screen to 1. GCSSetBlinkType(-1, "GButton00000", 1);

GCSGetBlinkType	Get blink type
Syntax	GCSGetBlinkType(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Blink type: 0: display/non-display of character string 1: change character color 2: change whole color
Details	Gets the object blink type.
Example	Gets the GButton00000 blink operation in the self screen in Stat. LONG Stat; Stat = GCSGetBlinkType(-1, "GButton00000");

GCSSetBounds	Set relative position within parent object
Syntax	GCSSetBounds(nWindowNo, strName, gmBounds);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmBounds : global memory for relative position
Return value	None
Details	Sets the relative position within the parent object.
Example	Sets the GButton00000 relative position in the self screen under the following conditions: upper left X coordinate : 10 upper left Y coordinate : 20 lower right X coordinate : 110 lower right Y coordinate : 120 mem = GMEMCreate("TESTMEM", 8); GMEMSetShort(mem, 0, 10); GMEMSetShort(mem, 2, 20); GMEMSetShort(mem, 4, 110); GMEMSetShort(mem, 6, 120); GCSSetBounds(-1, "GButton00000", mem); GMEMDelete(mem);

GCSGetBounds	Get relative position within parent object
Syntax	GCSGetBounds(nWindowNo, strName, gmBounds);
Argument	(i) LONG nWindowNo : screen No.(Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmBounds : global memory for relative position
Return value	Store the acquired relative position in gmBounds.
Details	Gets the relative position of the object within parent object.
Example	Sets the GButton00000 relative position in the self screen as follows: nXmin : upper left X coordinate nYmin : upper left Y coordinate nXmax : lower right X coordinate nYmax : lower right Y coordinate mem = GMEMCreate("TESTMEM", 8); GCSGetBounds(-1, "GButton00000", mem); GMEMGetShort(mem, 0); GMEMGetShort(mem, 2); GMEMGetShort(mem, 4); GMEMGetShort(mem, 6); GMEMDelete(mem);

GCSSetCaptionScrollEnable	Set scroll operation

Syntax	GCSSetCaptionScrollEnable(nWindowNo, strName, ucEnable);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucEnable : scroll operation setting
Return value	None
Details	Sets the object scroll operation. One of the following values is set for ucEnable: 0: disable 1: enable 2: enable when out of bounds
Example	Sets the GButton00000 scroll operation in the self screen to 1. GCSSetCaptionScrollEnable(-1, "GButton00000", 1);

GCSGetCaptionScrollEnable	Get scroll operation

Syntax	GCSGetCaptionScrollEnable(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Scroll operation 0: disable 1: enable 2: enable when out of bounds
Details	Gets the object scroll operation.
Example	Gets the GButton00000 scroll operation in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollEnable(-1, "GButton00000");

GCSStartCaptionScroll	Start scroll operation
Syntax	GCSStartCaptionScroll(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Starts the object scroll operation. Needs to set the scroll operation to 1 or 2 in advance with GCSSetCaptionScrollEnable.
Example	Starts the GButton00000 scroll operation in the self screen. GCSStartCaptionScroll(-1, "GButton00000");

GCSStopCaptionScroll	Stop scroll operation
Syntax	GCSStopCaptionScroll(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Stops the object scroll operation.
Example	Stops the GButton00000 scroll operation in the self screen. GCSStopCaptionScroll(-1, "GButton00000");

GCSPauseCaptionScroll	Pause scroll operation
Syntax	GCSPauseCaptionScroll(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Pauses the object scroll operation.
Example	Pauses the GButton00000 scroll operation in the self screen. GCSPauseCaptionScroll(-1, "GButton00000");

GCSRestartCaptionScroll	Restart scroll operation
Syntax	GCSRestartCaptionScroll(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Restart the object scroll operation.
Example	Restarts GButton00000 scroll operation in the self screen. GCSRestartCaptionScroll(-1, "GButton00000");

GCSIsCaptionOutOfBounds	Confirm overflowing of caption character string

Syntax	GCSIsCaptionOutOfBounds(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Out of bounds status 0: not out of bounds 1: out of bounds
Details	Confirms whether or not the caption character string set in the control goes out of the bounds of the control caption display area.
Example	Gets the status of the GButton00000 caption character string in the self screen in Stat as to whether or not it goes out of the bounds. LONG Stat; Stat = GCSIsCaptionOutOfBounds(-1, "GButton00000");

GCSSetCaptionScrollDelayTime	Set scroll delay time

Syntax	GCSSetCaptionScrollDelayTime(nWindowNo, strName, lTime);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lTime : delay time setting
Return value	None
Details	Sets the delay time for the object scroll starting time.
Example	Sets the GButton00000 scroll delay time in the self screen to 10. GCSSetCaptionScrollDelayTime(-1, "GButton00000", 10);

GCSGetCaptionScrollDelayTime	Get scroll delay time

Syntax	GCSGetCaptionScrollDelayTime(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Delay time
Details	Gets the delay time for the object scroll starting time.
Example	Gets the GButton00000 scroll delay time in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollDelayTime(-1, "GButton00000");

GCSSetCaptionScrollRefreshTime	Set scroll refresh time

Syntax	GCSSetCaptionScrollRefreshTime(nWindowNo, strName, lTime);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lTime : refresh time setting
Return value	None
Details	Sets the object scroll refresh time.
Example	Sets the GButton00000 scroll refresh time in the self screen to 10. GCSSetCaptionScrollRefreshTime(-1, "GButton00000", 10);

GCSGetCaptionScrollRefreshTime	Get scroll refresh time

Syntax	GCSGetCaptionScrollRefreshTime(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Refresh time
Details	Gets the object scroll refresh time.
Example	Gets the GButton00000 scroll refresh time in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollRefreshTime(-1, "GButton00000");

GCSSetCaptionScrollMovementValue	Set scroll movement value

Syntax	GCSSetCaptionScrollMovementValue(nWindowNo, strName, usMovementValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usMovementValue : movement value setting
Return value	None
Details	Sets the object scroll movement value.
Example	Sets the GButton00000 scroll movement value in the self screen to 10. GCSSetCaptionScrollMovementValue(-1, "GButton00000", 10);

GCSGetCaptionScrollMovementValue	Get scroll movement value

Syntax	GCSGetCaptionScrollMovementValue(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Movement value
Details	Gets the object scroll movement value.
Example	Gets the GButton00000 scroll movement value in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollMovementValue(-1, "GButton00000");

GCSSetCaptionScrollStartPosition	Set scroll start position

Syntax	GCSSetCaptionScrollStartPosition(nWindowNo, strName, ucStartPosition);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucStartPosition : start position setting
Return value	None
Details	Sets the object scroll start position. One of the following values is set for ucStartPositio: 0: Follow current caption setting. 1: From right.
Example	Sets the GButton00000 scroll start position in the self screen to 1. GCSSetCaptionScrollStartPosition(-1, "GButton00000", 1);

GCSGetCaptionScrollStartPosition	Get scroll start position

Syntax	GCSGetCaptionScrollStartPosition(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Start position 0: Follow current caption setting. 1: From right.
Details	Gets the object scroll start position.
Example	Gets the GButton00000 scroll start position in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollStartPosition(-1, "GButton00000");

GCSSetCaptionScrollPosition	Set total scroll movement value

Syntax	GCSSetCaptionScrollPosition(nWindowNo, strName, lPosition);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lPosition : total movement value setting
Return value	None
Details	Sets the total scroll movement value from the start position for the object.
Example	Sets the GButton00000 scroll total movement value in the self screen to 100. GCSSetCaptionScrollPosition(-1, "GButton00000", 100);

GCSGetCaptionScrollPosition	Get total scroll movement value

Syntax	GCSGetCaptionScrollPosition(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Total movement value
Details	Gets the total scroll movement value from the start position for the object.
Example	Gets the GButton00000 scroll total movement value in the self screen in Stat. LONG Stat; Stat = GCSGetCaptionScrollPosition(-1, "GButton00000");

GCSIsCaptionScrollFinish	Confirm caption scroll completion

Syntax	GCSIsCaptionScrollFinish(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Scroll finish status 0: scrolling 1: scrolling finish
Details	Gets the scroll finish status for the caption character string that is set in control.
Example	Gets the GButton00000 scroll finish status in the self screen in Stat. LONG Stat; Stat = GCSIsCaptionScrollFinish(-1, "GButton00000");

GCSGEShowPanel	Change display panels
Syntax	GCSGEShowPanel(nWindowNo);
Argument	(i) LONG nWindowNo : screen No.
Return value	None
Details	Change to the panel with the specified screen No.
Example	Changes to the self screen panel. GCSGEShowPanel(-1);

GCSPrevPage	Switch to previous screen
Syntax	GCSPrevPage(nShowWindow);
Argument	(i) LONG nShowWindow : switch window displays
Return value	None
Details	According to the screen history, switch to the previous screen. One of the following values is set for nShowWindow. 0: does not switch 1: switches (Note) This function is valid only when the the previous screen is the custom screen.
Example	Switches to the previous screen. GCSPrevPage(1);

GCSNextPage	Switch to next screen
Syntax	GCSNextPage(nShowWindow);
Argument	(i) LONG nShowWindow : switch window displays
Return value	None
Details	According to the screen history, switch to the next screen. One of the following values is set for nShowWindow: 0: does not switch 1: switches
Example	Switches to the next screen. GCSNextPage(1);

GCSCreateGWindow	Display window
Syntax	GCSCreateGWindow(nWindowNo);
Argument	(i) LONG nWindowNo : screen No.
Return value	None
Details	Displays the window with the specified screen No. (Note) When using GCSCreateGWindow, write it at the end of the macro process.
Example	Show the screen No.20 window. $Button-OnPress NormalMethod(); GCSCloseForceGWindow(10); GCSCreateGWindow(20); $End

GCSSetVisibleStatus	Switch show/hide of the control

Syntax	GCSSetVisibleStatus(nWindowNo, strName, Status);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) Long Status : Show/hide
Return value	None
Details	Switches the control is displayed/hidden. One of the following values is set for Status: 0: Not displayed 1: Displayed
Example	Hides the Gbutton0009.GCSSetVisibleStatus(-1,"GButton00009",0);

GCSShowPanel	Switch flame

Syntax	GCSShowPanel(nWindowNo, strName, Status);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) Long Status : frame page No.
Return value	None
Details	Switches page in the frane.
Example	Switches the Frame00001 frame to the frame page 1. GCSShowPanel(-1,"Frame00001",1);

GCSCloseForceGWindow	Close window
Syntax	GCSCloseForceGWindow (windowNo);
Argument	(i)LONG windowNo : screen No.
Return value	None
Details	Closes the window of the specified screen No. When the window is closed, the events registered up until that point are abandoned.
Example	Closes the window of the screen No. 1. GCSCloseForceWindow(1);

GCSCloseGWindow	Close window
Syntax	GCSCloseGWindow(windowNo);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) Long controlID : 0 fixed (i) Long lProcessID : process ID
Return value	None
Details	Closes the window of the designated screen No.
Example	Closes the window of the screen No. 1. GCSCloseGWindow(1);

GCSChangeActiveFocus	Change focus
Syntax	GCSChangeActiveFocus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Changes the focus.
Example	Changes the focus to GButton00001. GCSChangeActiveFocus(-1, "GButton00001");

GCSSetSystemNumber	Set part system number
Syntax	GCSSetSystemNumber(nWindowNo, strName, lSystem);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lSystem : part system number (1 or later)
Return value	None
Details	Sets the part system number. After the part system number is set, the control area is registered as a redraw area.
Example	Sets the GNXCounter00000 part system number in the self screen to 1. GCSSetSystemNumber(-1, "GNXCounter00000", 1) ;

GCSGetSystemNumber	Get part system number
Syntax	GCSGetSystemNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Part system number
Details	Gets the part system number.
Example	Gets the GNXCounter00000 part system number in the self screen in Stat. LONG Stat; Stat = GCSGetSystemNumber(-1, "GNXCounter00000") ;

5.3 Button

GCSButtonSetAction	Set button operations
Syntax	GCSButtonSetAction(nWindowNo, strName, ucAction);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucAction : button operation
Return value	None
Details	Sets the button operation. One of the following values is set for ucAction: 0: no operation 1: momentary operation 2: alternative operation When a value outside of the range is set, there is no forced change of the setting.
Example	Sets the GButton00000 operations in the self screen to the momentary mode. GCSButtonSetAction(-1, "GButton00000", 1);

GCSButtonGetAction	Get button operations
Syntax	GCSButtonGetAction(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: no operation 1: momentary operation 2: alternative operation Besides the above: illegal operation setting
Details	Gets the button operation setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GButton00000 operations in the self screen in Stat. LONG Stat; Stat = GCSButtonGetAction(-1, "GButton00000");

GCSButtonSetDisplay	Set button display
Syntax	GCSButtonSetDisplay(nWindowNo, strName, ucDisplay);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucDisplay : button display
Return value	None
Details	Sets the button display. One of the following values is set for ucDisplay: 0: rectangle display 1: image display 3: round shape display When a value outside of the range is set, there is no forced change of the setting. After the button display is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 in the self screen to be displayed in a rectangle. GCSButtonSetDisplay(-1, "GButton00000", 0);

GCSButtonGetDisplay	Get button display
Syntax	GCSButtonGetDisplay(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: rectangle display 1: image display 3: round shape display Besides the above: illegal display setting
Details	Gets the button display setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GButton00000 display setting in the self screen in Stat. LONG Stat; Stat = GCSButtonGetDisplay(-1, "GButton00000");

GCSButtonSetBorderID	Set 3D border resource ID
Syntax	GCSButtonSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 3D border resource ID in the self screen to 1. GCSButtonSetBorderID(-1, "GButton00000", 1);

GCSButtonGetBorderID	Get 3D border resource ID
Syntax	GCSButtonGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GButton00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSButtonGetBorderID(-1, "GButton00000");

GCSButtonSetOnDesign	Set ON status design
Syntax	GCSButtonSetOnDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for ON status. After the design for ON status is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 ON status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetLong(mem, 12, 2); GCSButtonSetOnDesign(-1, "GButton00000", mem); GMEMDelete(mem);

GCSButtonGetOnDesign	Get ON status design
Syntax	GCSButtonGetOnDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for ON status.

Example	Gets the GButton00000 ON status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSButtonGetOnDesign(-1, "GButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSButtonSetOffDesign	Set OFF status design
Syntax	GCSButtonSetOffDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for OFF status. After the design for OFF status is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 OFF status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSButtonSetOffDesign(-1, "GButton00000", mem); GMEMDelete(mem);

GCSButtonGetOffDesign	Get OFF status design
Syntax	GCSButtonGetOffDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for OFF status.

Example	Gets the GButton00000 OFF status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSButtonGetOffDesign(-1, "GButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSButtonSetFocusDesign	Set FOCUS status design

Syntax	GCSButtonSetFocusDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for FOCUS status. After the design for FOCUS status is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 FOCUS status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSButtonSetFocusDesign(-1, "GButton00000", mem); GMEMDelete(mem);

GCSButtonGetFocusDesign	Get FOCUS status design

Syntax	GCSButtonGetFocusDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for FOCUS status.
Example	Gets the GButton00000 FOCUS status design in the self screen as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSButtonGetFocusDesign(-1, "GButton00000", mem);
nPat = GMEMGetShort(mem, 0);
nFCol = GMEMGetLong(mem, 4);
nBCol = GMEMGetLong(mem, 8);
nImg = GMEMGetShort(mem, 12);
GMEMDelete(mem);
```

GCSButtonSetDisableDesign	Set Disable status design

Syntax	GCSButtonSetDisableDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for Disable status. After the design for Disable status is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 Disable status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSButtonSetDisableDesign(-1, "GButton00000", mem); GMEMDelete(mem);

GCSButtonGetDisableDesign	Get Disable status design

Syntax	GCSButtonGetDisableDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for Disable status.
Example	Gets the GButton00000 Disable status design in the self screen as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSButtonGetDisableDesign(-1, "GButton00000", mem);
nPat = GMEMGetShort(mem, 0);
nFCol = GMEMGetLong(mem, 4);
nBCol = GMEMGetLong(mem, 8);
nImg = GMEMGetShort(mem, 12);
GMEMDelete(mem);
```

GCSButtonSetFontID	Set font resource ID
Syntax	GCSButtonSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 font resource ID in the self screen to 1. GCSButtonSetFontID(-1, "GButton00000", 1);

GCSButtonGetFontID	Get font resource ID
Syntax	GCSButtonGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GButton00000 font resource ID in the self screen in Stat. LONG Stat; Stat=GCSButtonGetFontID(-1,"GButton0000");

GCSButtonSetStringID	Set caption character string resource ID
Syntax	GCSButtonSetStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 caption character string resource ID in the self screen to 1. GCSButtonSetStringID(-1, "GButton00000", 1);

GCSButtonGetStringID	Get caption character string resource ID
Syntax	GCSButtonGetStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
Example	Gets the GButton00000 caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSButtonGetStringID(-1, "GButton00000");

GCSButtonSetCaption	Set caption information
Syntax	GCSButtonSetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmCaption : global memory for caption information
Return value	None
Details	Sets caption information. After the caption information is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 caption information in the self screen as follows: Color : White (0xffffff) Horizontal position : 0 Vertical position : 1 Left margin : 10 Right margin : 0 Top margin : 0 Bottom margin : 0 mem = GMEMCreate("TESTMEM", 14); GMEMSetLong(mem, 0, HFFFFFF); GMEMSetChar(mem, 4, 0); GMEMSetChar(mem, 5, 1); GMEMSetShort(mem, 6, 10); GMEMSetShort(mem, 8, 0); GMEMSetShort(mem, 10, 0); GMEMSetShort(mem, 12, 0); GCSButtonSetCaption(-1, "GButton00000", mem); GMEMDelete(mem);

GCSButtonGetCaption	Get caption information
Syntax	GCSButtonGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Store the acquired caption information in gmCaption.
Details	Gets the caption information.

Example	Gets the GButton00000 caption information in the self screen as follows:
	nCol : Color
	nHPos : Horizontal position
	nVPos : Vertical position
	nLMgn : Left margin
	nRMgn : Right margin
	nTMgn : Top margin
	nBMgn : Bottom margin
	mem = GMEMCreate("TESTMEM", 14);
	GCSButtonGetCaption(-1, "GButton00000", mem);
	nCol = GMEMGetLong(mem, 0);
	nHPos = GMEMGetChar(mem, 4);
	nHPos = GMEMGetChar(mem, 5);
	nLMgn = GMEMGetShort(mem, 6);
	nRMgn = GMEMGetShort(mem, 8);
	nTMgn = GMEMGetShort(mem, 10);
	nBMgn = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSButtonSetFocusEffect	Set effect during focus
Syntax	GCSButtonSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 effect during focus in the self screen to 1. GCSButtonSetFocusEffect(-1, "GButton00000", 1);

GCSButtonGetFocusEffect	Get effect during focus
Syntax	GCSButtonGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GButton00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSButtonGetFocusEffect(-1, "GButton00000");

GCSButtonSetStatus	Set object status
Syntax	GCSButtonSetStatus(nWindowNo, strName, ucStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen) (i) STRING strName : control name (i) LONG ucStatus : object status
Return value	None
Details	Sets the GCButton object ON/OFF status. One of the following values is set for ucStatus: 0: OFF status 1: ON status After the GCButton object status is set, the control area is registered as a redraw area.
Example	Sets the GButton00000 ON/OFF status in the self screen to 1. GCSButtonSetStatus(-1, "GButton00000", 1);

GCSButtonGetStatus	Get object status
Syntax	GCSButtonGetStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen) (i) STRING strName : control name
Return value	Object status 0: OFF status 1: ON status
Details	Gets the GCButton object status.
Example	Gets the GButton00000 ON/OFF status in the self screen in Stat. LONG Stat; Stat = GCSButtonGetStatus(-1, "GButton00000");

5.4 CheckBox

GCSCheckboxSetBorderID	Set 3D border resource ID
Syntax	GCSCheckboxSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 3D border resource ID in the self screen to 1. GCSCheckboxSetBorderID(-1, "GCheckBox00000", 1);

GCSCheckboxGetBorderID	Get 3D border resource ID
Syntax	GCSCheckboxGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GCheckBox00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetBorderID(-1, "GCheckBox00000");

GCSCheckboxSetFontID	Set font resource ID
Syntax	GCSCheckboxSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 font resource ID in the self screen to 1. GCSCheckboxSetFontID(-1, "GCheckBox00000", 1);

GCSCheckboxGetFontID	Get font resource ID
Syntax	GCSCheckboxGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GCheckBox00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetFontID(-1, "GCheckBox00000");

GCSCheckboxSetStringID	Set caption character string resource ID
Syntax	GCSCheckboxSetStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 caption character string font resource ID in the self screen to 1. GCSCheckboxSetStringID(-1, "GCheckBox00000", 1);

GCSCheckboxGetStringID	Get caption character string resource ID
Syntax	GCSCheckboxGetStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID.
Details	Gets the caption character string resource ID.
Example	Gets the GCheckBox00000 caption character string font resource ID in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetStringID(-1, "GCheckBox00000");

GCSCheckboxSetCaption	Set caption information

Syntax

GCSCheckboxSetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmCaption : global memory for caption information

Return value

None

Details

Sets the caption information
All except character color are invalid.
After the caption information is set, the control area is registered as a redraw area.

Example

Sets the GCheckBox00000 caption information in the self screen as follows:
Color : White (0xffffff)
Horizontal position : 0
Vertical position : 1
Left margin : 10
Right margin : 0
Top margin : 0
Bottom margin : 0

```
mem = GMEMCreate("TESTMEM", 14);
GMEMSetLong(mem, 0, HFFFFFF);
GMEMSetChar(mem, 4, 0);
GMEMSetChar(mem, 5, 1);
GMEMSetShort(mem, 6, 10);
GMEMSetShort(mem, 8, 0);
GMEMSetShort(mem, 10, 0);
GMEMSetShort(mem, 12, 0);
GCSCheckboxSetCaption(-1, "GCheckBox00000", mem);
GMEMDelete(mem);
```

GCSCheckboxGetCaption	Get caption information
Syntax	GCSCheckboxGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Store the acquired caption information in gmCaption.
Details	Gets the caption information setting. All settings except character color are invalid.

Example	Gets the GCheckBox00000 caption information in the self screen as follows:
	nCol : Color
	nHPos : Horizontal position
	nVPos : Vertical position
	nLMgn : Left margin
	nRMgn : Right margin
	nTMgn : Top margin
	nBMgn : Bottom margin
	mem = GMEMCreate("TESTMEM", 14);
	GCSCheckboxGetCaption(-1, "GCheckBox00000", mem);
	nCol = GMEMGetLong(mem, 0);
	nHPos = GMEMGetChar(mem, 4);
	nHPos = GMEMGetChar(mem, 5);
	nLMgn = GMEMGetShort(mem, 6);
	nRMgn = GMEMGetShort(mem, 8);
	nTMgn = GMEMGetShort(mem, 10);
	nBMgn = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSCheckboxSetFocusEffect	Set effect during focus

Syntax	GCSCheckboxSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 effect during focus in the self screen to 1. GCSCheckboxSetFocusEffect(-1, "GCheckBox00000", 1);

GCSCheckboxGetFocusEffect	Get effect during focus

Syntax	GCSCheckboxGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GCheckBox00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetFocusEffect(-1, "GCheckBox00000");

GCSCheckboxSetStatus	Set object status
Syntax	GCSCheckboxSetStatus(nWindowNo, strName, ucStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucStatus : set object status
Return value	None
Details	Sets the ON/OFF status for the GCCheckBox object. One of the following values is set for ucStatus: 0: OFF status 1: ON status After the object status is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 ON/OFF status in the self screen to 1. GCSCheckboxSetStatus(-1, "GCheckBox00000", 1);

GCSCheckboxGetStatus	Get object status
Syntax	GCSCheckboxGetStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)) (i) STRING strName : control name
Return value	Object status 0: OFF status 1: ON status
Details	Gets status for GCCheckBox object.
Example	Gets the GCheckBox00000 ON/OFF status in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetStatus(-1, "GCheckBox00000");

GCSCheckboxSetBoxSize	Set box size
Syntax	GCSCheckboxSetBoxSize(nWindowNo, strName, lBoxSize);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lBoxSize : check box size
Return value	None
Details	Sets the box size. After the box size is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 box size in the self screen to 10. GCSCheckboxSetBoxSize(-1, "GCheckBox00000", 10);

GCSCheckboxGetBoxSize	Get box size
Syntax	GCSCheckboxGetBoxSize(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box size
Details	Gets the box size settings.
Example	Gets the GCheckBox00000 box size in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetBoxSize(-1, "GCheckBox00000");

GCSCheckboxSetBoxColor	Set box color

Syntax	GCSCheckboxSetBoxColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : box color
Return value	None
Details	Sets the box color. After the box color is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 box color in the self screen to white (0xffffff). GCSCheckboxSetBoxColor(-1,"GCheckBox00000",HFFFFFF);

GCSCheckboxGetBoxColor	Get box color

Syntax	GCSCheckboxGetBoxColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box color
Details	Gets the box color.
Example	Gets the GCheckBox00000 box color in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetBoxColor(-1, "GCheckBox00000");

GCSCheckboxSetFocusColor	Set background color during focus

Syntax	GCSCheckboxSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : background color during focus
Return value	None
Details	Sets the background color in the case where the effect during focus is "change color". After the background color is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 background color during focus in the self screen to white (0xffffff). GCSCheckboxSetFocusColor(-1, "GCheckBox00000", HFFFFFF);

GCSCheckboxGetFocusColor	Get background color during focus

Syntax	GCSCheckboxGetFocusColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Background color during focus
Details	Gets the background color in the case where the effect during focus is "change color".
Example	Gets the GCheckBox00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSCheckboxGetFocusColor(-1, "GCheckBox00000");

GCSCheckboxSetDisableCaptionColor	Set character color when disabled

Syntax	GCSCheckboxSetDisableCaptionColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : character color when disabled
Return value	None
Details	Sets the character color during disabled control. After the character color is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 character color when disabled in the self screen to white (0xffffff). GCSCheckboxSetDisableCaptionColor(-1, "GCheckBox00000", HFFFFFF);

GCSCheckboxGetDisableCaptionColor	Get character color when disabled

Syntax	GCSCheckboxGetDisableCaptionColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Character color when disabled
Details	Gets the character color during disabled control.
Example	Gets the GCheckBox00000 character color when disabled in the self screen in Stat. LONG Stat; Stat=GCSCheckboxGetDisableCaptionColor(-1, "GCheckBox00000");

GCSCheckboxSetDisableBoxColor	Set box color when disabled

Syntax	GCSCheckboxSetDisableBoxColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : box color
Return value	None
Details	Sets the box color during disabled control. After the box color is set, the control area is registered as a redraw area.
Example	Sets the GCheckBox00000 box color when disabled in the self screen to white (0xffffff). GCSCheckboxSetDisableBoxColor(-1, "GCheckBox00000", HFFFFFF);

GCSCheckboxGetDisableBoxColor	Get box color when disabled

Syntax	GCSCheckboxGetDisableBoxColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box color when disabled
Details	Gets the box color during disabled control.
Example	Gets the GCheckBox00000 box color when disabled in the self screen in Stat. LONG Stat; Stat=GCSCheckboxGetDisableBoxColor(-1, "GCheckBox00000");

5.5 Edit

GCSEditSetBorderID	Set 3D border resource ID
Syntax	GCSEditSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No.(Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GEdit00000 3D border resource ID in the self screen to 1. GCSEditSetBorderID(-1, "GEdit00000", 1);

GCSEditGetBorderID	Get 3D border resource ID
Syntax	GCSEditGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No.(Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
Example	Gets the GEdit00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSEditGetBorderID(-1, "GEdit00000");

GCSEditSetFontID	Set font resource ID
Syntax	GCSEditSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GEdit00000 font resource ID in the self screen to 1. GCSEditSetFontID(-1, "GEdit00000", 1);

GCSEditGetFontID	Get font resource ID
Syntax	GCSEditGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GEdit00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSEditGetFontID(-1, "GEdit00000");

GCSEditAddString	Add character string
Syntax	GCSEditAddString(nWindowNo, strName, GTCHAR* pszString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszString : character string to be added
Return value	0: processing failed 1: processing succeeded
Details	Adds the character string to the end.
Example	Adds the character string "ABCD" to the end of GEdit00000 in the self screen. LONG Stat; Stat = GCSEditAddString(-1, "GEdit00000", "ABCD");

GCSEditSetTextColor	Set character display color
Syntax	GCSEditSetTextColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for character display color
Return value	None
Details	Sets the character display color.
Example	Sets the GEdit00000 character display color in the self screen to 0xffffff. GCSEditSetTextColor(-1, "GEdit00000", HFFFFFF);

GCSEditGetTextColor	Get character display color
Syntax	GCSEditGetTextColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for character display color
Details	Gets the character display color.
Example	Gets the GEdit00000 character display color in the self screen in Stat. LONG Stat; Stat = GCSEditGetTextColor(-1, "Gedit00000");

GCSEditSetBackColor	Set background color
Syntax	GCSEditSetBackColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color
Return value	None
Details	Sets the background color.
Example	Sets the GEdit00000 background color in the self screen to 0xffffff. GCSEditSetBackColor(-1, "GEdit00000", HFFFFFF);

GCSEditGetBackColor	Get background color
Syntax	GCSEditGetBackColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color
Details	Gets the background color.
Example	Gets the GEdit00000 background color in the self screen in Stat. LONG Stat; Stat = GCSEditGetBackColor(-1, "GEdit00000");

GCSEditSetBrush	Set brush
Syntax	GCSEditSetBrush(nWindowNo, strName, gmBrush);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmBrush : global memory for brush
Return value	1: processing succeeded 0: processing failed
Details	Sets the brush.
Example	Sets the GEdit00000 brush in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 mem = GMEMCreate("TESTMEM", 12); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GCSEditSetBrush(-1, "GEdit00000", mem); GMEMDelete(mem);

GCSEditGetBrush	Get brush
Syntax	GCSEditGetBrush(nWindowNo, strName, gmBrush);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmBrush : global memory for brush
Return value	1: processing succeeded 0: processing failed
Details	Gets the brush.

Example	Gets the GEdit00000 brush in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	mem = GMEMCreate("TESTMEM", 12);
	GCSEditGetBrush(-1, "GEdit00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	GMEMDelete(mem);

GCSEditDeleteString	Delete character string
Syntax	GCSEditDeleteString(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Deletes all the character strings.
Example	Deletes all the GEdit00000 character strings in the self screen. GCSEditDeleteString(-1, "GEdit00000");

GCSEditAddLine	Add line
Syntax	GCSEditAddLine(nWindowNo, strName, pszLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszLine : line to be added
Return value	0: processing failed 1: processing succeeded
Details	Adds the character string as a new line at the end.
Example	Adds the line "ABCD" to the end of GEdit00000 in the self screen. GCSEditAddLine(-1, "GEdit00000", "ABCD");

GCSEditInsertLine	Insert line
Syntax	GCSEditInsertLine(nWindowNo, strName, pszLine, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszLine : line to be inserted (i) LONG ulLine : line No.
Return value	0 : processing failed Other than 0 : number of lines inserted
Details	Inserts the character string as a new line at specified position.
Example	Inserts the line "ABCD" into the GEdit00000 line No.2 in the self screen. GCSEditInsertLine(-1, "GEdit00000", "ABCD", 2);

GCSEditDeleteLine	Delete line
Syntax	GCSEditDeleteLine(nWindowNo, strName, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulLine : line No.
Return value	0: processing failed 1: processing succeeded
Details	Deletes the specified line.
Example	Deletes GEdit00000 line No.2 in the self screen. GCSEditDeleteLine(-1, "GEdit00000", "ABCD", 2);

GCSEditGetLineString	Get line character string
Syntax	GCSEditGetLineString(nWindowNo, strName, ulLine, GTCHAR* pszString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulLine : line No. (o) STRING pszString : character string to be acquired
Return value	0: processing failed 1: processing succeeded
Details	Gets the maximum of 256byte character string for the specified line.
Example	Gets the contents of the GEdit00000 line No.2 in strStat. STRING strStat; GCSEditGetLineString(-1, "GEdit00000", 2, strStat);

GCSEditSetLineFeedCode	Set line feed character string

Syntax	GCSEditSetLineFeedCode(nWindowNo, strName, ucLineFeedCode);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usLineFeedType : line feed character type
Return value	None
Details	Sets the line feed character string type. One of the following values is set for usLineFeedType: 0:CR+LF 1:LF 2:CR
Example	Sets the GEdit00000 linefeed character string type in the self screen to CR+LF. STRING strStat; GCSEditSetLineFeedCode(-1, "GEdit00000", 0);

GCSEditGetLineFeedCode	Get line feed character string

Syntax	GCSEditGetLineFeedCode(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Line feed character string type 0:CR+LF 1:LF 2:CR
Details	Gets the line feed character string type.
Example	Gets the GEdit00000 line feed character string type in the self screen in Stat. LONG Stat; Stat = GCSEditGetLineFeedCode(-1, "GEdit00000");

GCSEditSetCursor	Set cursor position
Syntax	GCSEditSetCursor(nWindowNo, strName, ulLine, ulIndex);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulLine : line No. (i) LONG ulindex : index No.
Return value	0: processing failed 1: processing succeeded
Details	Sets the cursor position to the specified position.
Example	Sets the GEdit00000 cursor position in the self screen to line No.2 and index No.3. GCSEditSetCursor(-1, "GEdit00000", 2, 3);

GCSEditGetCursor	Get cursor position
Syntax	GCSEditGetCursor(nWindowNo, strName, gmCurPos);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCurPos : global memory for cursor position
Return value	0: processing failed 1: processing succeeded
Details	Gets the cursor position.

Example	Gets the GEdit00000 cursor position in the self screen as follows:
	nLine : Line No.
	nIndex : Index No.
	LONG nLine;
	LONG nIndex;
	mem = GMEMCreate("TESTMEM", 8);
	GCSEditGetCursor(-1, "GEdit00000", mem);
	nLine = GMEMGetLong(mem, 0);
	nIndex = GMEMGetLong(mem, 4);
	GMEMDelete(mem);

GCSEditSetForeColor	Set foreground color
Syntax	GCSEditSetForeColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for foreground color
Return value	None
Details	Sets the foreground color. After the foreground color is set, the control area is registered as a redraw area.
Example	Sets the GEdit00000 foreground color in the self screen to 0xffffff. GCSEditSetForeColor(-1, "GEdit00000", HFFFFFF);

GCSEditGetForeColor	Get foreground color
Syntax	GCSEditGetForeColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for foreground color
Details	Gets the foreground color.
Example	Gets the GEdit00000 foreground color in the self screen in Stat. LONG Stat; Stat = GCSEditGetForeColor(-1, "GEdit00000");

GCSEditSetFillPattern	Set fill pattern
Syntax	GCSEditSetFillPattern(nWindowNo, strName, nFillPattern);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nFillPattern : fill pattern
Return value	None
Details	Sets the fill pattern. After the fill pattern is set, the control area is registered as a redraw area.
Example	Sets the GEdit00000 fill pattern in the self screen to 2. GCSEditSetFillPattern(-1, "GEdit00000", 2);

GCSEditGetFillPattern	Get fill pattern
Syntax	GCSEditGetFillPattern(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Fill pattern
Details	Gets the fill pattern.
Example	Gets the GEdit00000 fill pattern in the self screen in Stat. LONG Stat; Stat = GCSEditGetFillPattern(-1, "GEdit00000");

GCSEditSetInsertMode	Set insert/overwrite mode
Syntax	GCSEditSetInsertMode(nWindowNo, strName, nIsInsert);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nIsInsert : mode setting
Return value	None
Details	Sets the insert/overwrite mode. One of the following values is set for nIsInsert: 1: insert mode 0: overwrite mode
Example	Sets the GEdit00000 insert/overwrite mode in the self screen to "overwrite" mode. GCSEditSetInsertMode(-1, "GEdit00000", 1);

GCSEditGetInsertMode	Get insert/overwrite mode
Syntax	GCSEditGetInsertMode(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	1: insert mode 0: overwrite mode
Details	Gets the insert/overwrite mode.
Example	Gets the GEdit00000 insert/overwrite mode in the self screen in Stat. LONG Stat; Stat = GCSEditGetInsertMode(-1, "GEdit00000");

GCSEditSetLineBrush	Set line brush
Syntax	GCSEditSetLineBrush(nWindowNo, strName, gmBrush, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmBrush : global memory for brush (i) LONG ulLine : line No.
Return value	None
Details	Sets the brush for the specified line.
Example	Sets the GEdit00000 line No.5 brush in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 mem = GMEMCreate("TESTMEM", 12); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GCSEditSetLineBrush(-1, "GEdit00000", mem, 5); GMEMDelete(mem);

GCSEditGetLineBrush	Get line brush
Syntax	GCSEditGetLineBrush(nWindowNo, strName, gmBrush, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmBrush : global memory for brush (i) LONG ulLine : line No.
Return value	1: processing succeeded 0: processing failed
Details	Gets the brush for the specified line.

Example	Gets the GEdit00000 line No.5 brush in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	mem = GMEMCreate("TESTMEM", 12);
	GCSEditGetLineBrush(-1, "GEdit00000", mem, 5);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	GMEMDelete(mem);

GCSEditSetLineTextColor	Set line text display color
Syntax	GCSEditSetLineTextColor(nWindowNo, strName, gcColor, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for text display color (i) LONG ulLine : line No.
Return value	None
Details	Sets the text display color for the specified line.
Example	Sets the GEdit00000 line No.5 text display color in the self screen to 0xffffff. GCSEditSetLineTextColor(-1, "GEdit00000", HFFFFFF, 5);

GCSEditGetLineTextColor	Get line text display color
Syntax	GCSEditGetLineTextColor(nWindowNo, strName, ulLine);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for text display color for specified line
Details	Gets the text display color for the specified line.
Example	Gets the GEdit00000 line No.5 text display color in the self screen in Stat. LONG Stat; Stat = GCSEditGetLineTextColor(-1, "GEdit00000", 5);

GCSEditGetLength	Get character length
Syntax	GCSEditGetLength(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Number of characters
Details	Gets the total current character length.
Example	Gets the GEdit00000 total character length in the self screen in Stat. LONG Stat; Stat = GCSEditGetLength(-1, "GEdit00000");

GCSEditFind	Search forward

Syntax

GCSEditFind(nWindowNo, strName, pszString, ulStartLine, ulStartIndex, gmFindPos);

Argument

(i) LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i) STRING strName	: control name
(i) STRING pszString	: character string for search
(i) LONG ulStartLine	: starting line for search
(i) LONG ulStartIndex	: starting position for search
(o) GMEM gmFindPos	: search result

Return value

0: search failed
1: search succeeded

Details

Searchs the character string for search from the search starting position toward the end. Returns the position of the character string found first.

Example

Searches GEdit00000 in the self screen under the following conditions:
 starting line for search : 3
 starting position for search : 5
 character string for search : "ABCD"
Gets the result in:
 nStat : search succeeded/failed
 nLine : searched line
 nIndex : searched position

```
LONG nLine;
LONG nIndex;
LONG Stat;
mem = GMEMCreate("TESTMEM", 8);
Stat = GCSEditFind(-1, "GEdit00000", "ABCD", 3, 5, mem);
nLine  = GMEMGetLong(mem, 0);
nIndex = GMEMGetLong(mem, 4);
GMEMDelete(mem);
```

GCSEditReverseFind	Search backward

Syntax

GCSEditReverseFind(nWindowNo, strName, pszString, ulStartLine, ulStartIndex, gmFindPos);

Argument

(i) LONG nWindowNo	: search No. (Specify -1 for self screen.)
(i) STRING strName	: control name
(i) STRING pszString	: character string for search
(i) LONG ulStartLine	: starting line for search
(i) LONG ulStartIndex	: starting position for search
(o) GMEM gmFindPos	: search result

Return value

0: search failed
1: search succeeded

Details

Searchs the character string for search from the search starting position toward the beginning.
Returns the position of the character string found first.

Example

Searches GEdit00000 in the self screen under the following conditions:
 starting line for search : 3
 starting position for search : 5
 character string for search : "ABCD"
Gets the result in:
 nStat : search succeeded/failed
 nLine : searched line
 nIndex : searched position

```
LONG nLine;
LONG nIndex;
LONG Stat;
mem = GMEMCreate("TESTMEM", 8);
Stat = GCSEditReverseFind(-1, "GEdit00000", "ABCD", 3, 5,
mem);
nLine  = GMEMGetLong(mem, 0);
nIndex = GMEMGetLong(mem, 4);
GMEMDelete(mem);
```

GCSEditReplace	Replace character string

Syntax

GCSEditReplace(nWindowNo, strName, pszString, ulStartLine, ulStartIndex, ulSize);

Argument

(i) LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i) STRING strName	: control name
(i) STRING pszString	: replacement character string
(i) LONG ulStartLine	: starting line for replacement
(i) LONG ulStartIndex	: starting position for replacement
(i) LONG ulSize	: replacement size

Return value

0: replacement failed
1: replacement succeeded

Details

Replaces the character string for the amount of replacement size from the replacement starting position.

Example

Executes a replacement process for GEdit00000 in the self screen under the following conditions:
 replacement character string : "ABCD"
 starting line for replacement : 3
 starting position for replacement : 5
 replacement size : 8
Gets the replacement result in Stat.

```
LONG Stat;
Stat = GCSEditReplace(-1, "GEdit00000", "ABCD", 3, 5, 8);
```

GCSEditReplaceAll	Replaces all character strings
Syntax	GCSEditReplaceAll(nWindowNo, strName, pszReplace, pszFind);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszReplace : replacement character string (i) STRING pszFind : character string for search
Return value	Number of replacements
Details	Searches the character string for search starting from the character string in the buffer and replaces it with the replacement character string.
Example	Executes an entire replacement process for GEdit00000 in the self screen under the following conditions: character string for search : "ABC" replacement character string : "12345" Gets the number of replacements in Stat. LONG Stat; Stat =GCSEditReplaceAll(-1, "GEdit00000", "12345", "ABC");

GCSEditInsertString	Insert character string
Syntax	GCSEditInsertString(nWindowNo, strName, pszString, ulLine, ulIndex);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszString : character string to be inserted (i) LONG ulLine : line for insertion (i) LONG ulIndex : insertion position
Return value	0: processing failed 1: processing succeeded
Details	Inserts character string.
Example	Executes a character string insertion process for GEdit00000 in the self screen under the following conditions: character string to be inserted : "ABCD" line for insertion : 3 insertion position : 5 Gets the insertion result in Stat. LONG Stat; Stat = GCSEditInsertString(-1, "GEdit00000", "ABCD", 3, 5);

GCSEditGetLineCount	Get number of lines
Syntax	GCSEditGetLineCount(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Number of lines
Details	Gets the current number of lines.
Example	Gets the number of GEdit00000 lines in the self screen in Stat. LONG Stat; Stat = GCSEditGetLineCount(-1, "GEdit00000");

GCSEditRemoveString	Delete character string
Syntax	GCSEditRemoveString(nWindowNo, strName, ulRemoveSize);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulRemoveSize : size of character string for deletion
Return value	0: processing failed 1: processing succeeded
Details	Deletes character string from the end.
Example	Deletes a 4-character-long character string from the end of GEdit0000 in the self screen, then gets the result in Stat. LONG Stat; Stat = GCSEditRemoveString(-1, "GEdit00000", 4);

5.6 HtmlBrowser

GCSHtmlbrowserSetBorderID	Set 3D border resource ID

Syntax	GCSHtmlbrowserSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 3D border resource ID in the self screen to 1. GCSHtmlbrowserSetBorderID(-1, "GHtmlBrowser00000", 1);

GCSHtmlbrowserGetBorderID	Get 3D border resource ID

Syntax	GCSHtmlbrowserGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
Example	Gets the GHtmlBrowser00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetBorderID(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetFontID	Set font resource ID

Syntax	GCSHtmlbrowserSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 font resource ID in the self screen to 1. GCSHtmlbrowserSetFontID(-1, "GHtmlBrowser00000", 1);

GCSHtmlbrowserGetFontID	Get font resource ID

Syntax	GCSHtmlbrowserGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font the resource ID
Details	Gets the font resource ID.
Example	Gets the GHtmlBrowser00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetFontID(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetScrollBarWidth	Set scroll bar width

Syntax	GCSHtmlbrowserSetScrollBarWidth(nWindowNo, strName, ucScroolBarWidth);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucScrollBarWidth : scroll bar width
Return value	None
Details	Sets the scroll bar width. After the scroll bar width is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 scroll bar width in the self screen to 20. GCSHtmlbrowserSetScrollBarWidth(-1, "GHtmlBrowser00000", 20);

GCSHtmlbrowserGetScrollBarWidth	Get scroll bar width

Syntax	GCSHtmlbrowserGetScrollBarWidth(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Scroll bar width
Details	Gets the scroll bar width.
Example	Gets the GHtmlBrowser00000 scroll bar width in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetScrollBarWidth(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetTextColor	Set text display color

Syntax	GCSHtmlbrowserSetTextColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for text display color
Return value	None
Details	Sets the text display color. After the text display color is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 text color in the self screen to white (0xffffff). GCSHtmlbrowserSetTextColor(-1, "GHtmlBrowser00000", HFFFFFF);

GCSHtmlbrowserGetTextColor	Get text display color

Syntax	GCSHtmlbrowserGetTextColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for text display color
Details	Gets the text display color.
Example	Gets the GHtmlBrowser00000 text color in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetTextColor(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetBackColor	Set background color

Syntax	GCSHtmlbrowserSetBackColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color
Return value	None
Details	Sets the background color. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 background color in the self screen to white (0xffffff). GCSHtmlbrowserSetBackColor(-1, "GHtmlBrowser00000", HFFFFFF);

GCSHtmlbrowserGetBackColor	Get background color

Syntax	GCSHtmlbrowserGetBackColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color
Details	Gets the background color.
Example	Gets the GHtmlBrowser00000 background color in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetBackColor(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetLinkColor	Set link color

Syntax	GCSHtmlbrowserSetLinkColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for link color
Return value	None
Details	Sets the link color. After the link color is set, the control area is registered as a redraw area.
Example	Sets the GHtmlBrowser00000 link color in the self screen to white (0xffffff). GCSHtmlbrowserSetLinkColor(-1, "GHtmlBrowser00000", HFFFFFF);

GCSHtmlbrowserGetLinkColor	Get link color

Syntax	GCSHtmlbrowserGetLinkColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for link color
Details	Gets the link color.
Example	Gets the GHtmlBrowser00000 link color in the self screen in Stat. LONG Stat; Stat = GCSHtmlbrowserGetLinkColor(-1, "GHtmlBrowser00000");

GCSHtmlbrowserSetHtmlFileName	Set HTML file name

Syntax	GCSHtmlbrowserSetHtmlFileName(nWindowNo, strName, FileName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING FileName : HTML file name
Return value	None
Details	Sets the HTML file name. After the HTML file name is set, the read complete flag is cleared to 0 for reading again.
Example	Sets the GHtmlBrowser00000 HTML file name in the self screen in "index.htm". GCSHtmlbrowserSetHtmlFileName(-1, "GHtmlBrowser00000", "index.htm");

GCSHtmlbrowserGetHtmlFileName	Get HTML file name

Syntax	GCSHtmlbrowserGetHtmlFileName(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self sreen.) (i) STRING strName : control name
Return value	HTML file name
Details	Gets the HTML file name.
Example	Gets the GHtmlBrowser00000 HTML file name in the self screen in strStat. STRING strStat; strStat = GCSHtmlbrowserGetHtmlFileName(-1, "GHtmlBrowser00000");

5.7 Label

GCSLabelSetFontID	Set font resource ID
Syntax	GCSLabelSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GLabel00000 font resource ID in the self screen to 1. GCSLabelSetFontID(-1, "GLabel00000", 1);

GCSLabelGetFontID	Get font resource ID
Syntax	GCSLabelGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GLabel00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSLabelGetFontID(-1, "GLabel00000");

GCSLabelSetStringID	Set caption character string resource ID
Syntax	GCSLabelSetStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GLabel00000 caption character string resource ID in the self screen to 1. GCSLabelSetStringID(-1, "GLabel00000", 1);

GCSLabelGetStringID	Get caption character string resource ID
Syntax	GCSLabelGetStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
Example	Gets the GLabel00000 caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSLabelGetStringID(-1, "GLabel00000");

GCSLabelSetCaption	Set caption information
Syntax	GCSLabelSetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmCaption : global memory for caption information
Return value	None
Details	Set caption information. After the caption information is set, the control area is registered as a redraw area.
Example	Sets the GLabel00000 caption information in the self screen as follows: Color : White (0xffffff) Horizontal position : 0 Vertical position : 1 Left margin : 10 Right margin : 0 Top margin : 0 Bottom margin : 0 mem = GMEMCreate("TESTMEM", 14); GMEMSetLong(mem, 0, HFFFFFF); GMEMSetChar(mem, 4, 0); GMEMSetChar(mem, 5, 1); GMEMSetShort(mem, 6, 10); GMEMSetShort(mem, 8, 0); GMEMSetShort(mem, 10, 0); GMEMSetShort(mem, 12, 0); GCSLabelSetCaption(-1, "GLabel00000", mem); GMEMDelete(mem);

GCSLabelGetCaption	Get caption information
Syntax	GCSLabelGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Stores the acquired caption information in gmCaption.
Details	Gets the caption information.

Example	Gets the GLabel00000 caption information in the self screen as follows:
	nCol : Color
	nHPos : Horizontal position
	nVPos : Vertical position
	nLMgn : Left margin
	nRMgn : Right margin
	nTMgn : Top margin
	nBMgn : Bottom margin
	mem = GMEMCreate("TESTMEM", 14);
	GCSLabelGetCaption(-1, "GLabel00000", mem);
	nCol = GMEMGetLong(mem, 0);
	nHPos = GMEMGetChar(mem, 4);
	nHPos = GMEMGetChar(mem, 5);
	nLMgn = GMEMGetShort(mem, 6);
	nRMgn = GMEMGetShort(mem, 8);
	nTMgn = GMEMGetShort(mem, 10);
	nBMgn = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

5.8 List

GCSListSetBorderID	Set 3D border resource ID
Syntax	GCSListSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GList00000 3D border resource ID in the self screen to 1. GCSListSetBorderID(-1, "GList00000", 1);

GCSListGetBorderID	Get 3D border resource ID
Syntax	GCSListGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID
Example	Gets the GList00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSListGetBorderID(-1, "GList00000");

GCSListSetFontID	Set font resource ID
Syntax	GCSListSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GList00000 font resource ID in the self screen to 1. GCSListSetFontID(-1, "GList00000", 1);

GCSListGetFontID	Get font resource ID
Syntax	GCSListGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GList00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSListGetFontID(-1, "GList00000");

GCSListSetCaption	Set caption information

Syntax	GCSListSetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmCaption : global memory for caption information
Return value	None
Details	Set caption information. After the caption information is set, the control area is registered as a redraw area.
Example	Sets the GList00000 caption information in the self screen as follows: Color : White (0xffffff) Horizontal position : 0 Vertical position : 1 Left margin : 10 Right margin : 0 Top margin : 0 Bottom margin : 0

```
mem = GMEMCreate("TESTMEM", 14);
GMEMSetLong(mem, 0, HFFFFFF);
GMEMSetChar(mem, 4, 0);
GMEMSetChar(mem, 5, 1);
GMEMSetShort(mem, 6, 10);
GMEMSetShort(mem, 8, 0);
GMEMSetShort(mem, 10, 0);
GMEMSetShort(mem, 12, 0);
GCSListSetCaption(-1, "GList00000", mem);
GMEMDelete(mem);
```

GCSListGetCaption	Get caption information
Syntax	GCSListGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Stores the acquired caption information in gmCaption.
Details	Gets the caption information setting.

Example	Gets the GList00000 caption information in the self screen as follows:
	nCol : Color
	nHPos : Horizontal position
	nVPos : Vertical position
	nLMgn : Left margin
	nRMgn : Right margin
	nTMgn : Top margin
	nBMgn : Bottom margin
	mem = GMEMCreate("TESTMEM", 14);
	GCSListGetCaption(-1, "GList00000", mem);
	nCol = GMEMGetLong(mem, 0);
	nHPos = GMEMGetChar(mem, 4);
	nHPos = GMEMGetChar(mem, 5);
	nLMgn = GMEMGetShort(mem, 6);
	nRMgn = GMEMGetShort(mem, 8);
	nTMgn = GMEMGetShort(mem, 10);
	nBMgn = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSListSetFocusColor	Set background color during focus
Syntax	GCSListSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color during
Return value	None
Details	Sets the background color during focus. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 background color during focus in the self screen to white (0xffffff). GCSListSetFocusColor(-1, "GList00000", HFFFFFF);

GCSListGetFocusColor	Get background color during focus
Syntax	GCSListGetFocusColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color during focus
Details	Gets the background color during focus.
Example	Gets the GList00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSListGetFocusColor(-1, "GList00000");

GCSListSetScrollButtonColor	Set scroll button color

Syntax	GCSListSetScrollButtonColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for scroll button color
Return value	None
Details	Sets the scroll button color. After the scroll button color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 scroll button color in the self screen to white (0xffffff). GCSListSetScrollButtonColor(-1, "GList00000", HFFFFFF);

GCSListGetScrollButtonColor	Get scroll button color

Syntax	GCSListGetScrollButtonColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code
Details	Gets the scroll button color.
Example	Gets the GList00000 scroll button color in the self screen in Stat. LONG Stat; Stat = GCSListGetScrollButtonColor(-1, "GList00000");

GCSListSetScrollBarColor	Set scroll bar color
Syntax	GCSListSetScrollBarColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for scroll bar color
Return value	None
Details	Sets the scroll bar color. After the scroll bar color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 scroll bar color in the self screen to white (0xffffff). GCSListSetScrollBarColor(-1, "GList00000", HFFFFFF);

GCSListGetScrollBarColor	Get scroll bar color
Syntax	GCSListGetScrollBarColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code
Details	Gets the scroll bar color.
Example	Gets the GList00000 scroll bar color in the self screen in Stat. LONG Stat; Stat = GCSListGetScrollBarColor(-1, "GList00000");

GCSListSetNormalColor	Set normal background color
Syntax	GCSListSetNormalColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the normal background color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 normal background color in the self screen to white (0xffffff). GCSListSetNormalColor(-1, "GList00000", HFFFFFF);

GCSListGetNormalColor	Get normal background color
Syntax	GCSListGetNormalColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for normal background color
Details	Gets the normal background color.
Example	Gets the GList00000 normal background color in the self screen in Stat. LONG Stat; Stat = GCSListGetNormalColor(-1, "GList00000");

GCSListSetDisableColor	Set background color when disabled
Syntax	GCSListSetDisableColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 background color when disabled in the self screen to white (0xffffff). GCSListSetDisableColor(-1, "GList00000", HFFFFFF);

GCSListGetDisableColor	Get background color when disabled
Syntax	GCSListGetDisableColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
Example	Gets the GList00000 background color when disabled in the self screen in Stat. LONG Stat; Stat = GCSListGetDisableColor(-1, "GList00000");

GCSListSetScrollBarWidth	Set scroll bar width
Syntax	GCSListSetScrollBarWidth(nWindowNo, strName, ucScroolBarWidth);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucScrollBarWidth : scroll bar width
Return value	None
Details	Sets the scroll bar width. After the scroll bar width is set, the control area is registered as a redraw area.
Example	Sets the GList00000 scroll bar width in the self screen to 20. GCSListSetScrollBarWidth(-1, "GList00000", 20);

GCSListGetScrollBarWidth	Get scroll bar width
Syntax	GCSListGetScrollBarWidth(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Scroll bar width
Details	Gets the scroll bar width.
Example	Gets the GList00000 scroll bar width in the self screen in Stat. LONG Stat; Stat = GCSListGetScrollBarWidth(-1, "GList00000");

GCSListSetSelectBarColor	Set selection bar color

Syntax	GCSListSetSelectBarColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for selection bar color
Return value	None
Details	Sets the selection bar color. After the selection bar color is set, the control area is registered as a redraw area.
Example	Sets the GList00000 select bar color in the self screen to white (0xffffff) GCSListSetSelectBarColor(-1, "GList00000", HFFFFFF);

GCSListGetSelectBarColor	Get selection bar color

Syntax	GCSListGetSelectBarColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for selection bar
Details	Gets the selection bar color.
Example	Gets the GList00000 select bar in the self screen in Stat. LONG Stat; Stat = GCSListGetSelectBarColor(-1, "GList00000");

GCSListSetMaxListLines	Set maximum number of lines in list
Syntax	GCSListSetMaxListLines(nWindowNo, strName, usMaxListLines);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usMaxListLines : maximum number 0 to 32767
Return value	None
Details	Sets the maximum number of lines for list control. The range of settings is 0 to 32767. When the maximum number of lines for display that is set is less than the number of lines in a particular list already registered, the character strings in the list that are over the setting are deleted.
Example	Sets the maximum number of GList00000 lines in the self screen to 20. GCSListSetMaxListLines(-1, "GList00000", 20);

GCSListGetMaxListLines	Get maximum number of lines in list
Syntax	GCSListGetMaxListLines(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Maximum number of lines in list
Details	Gets the maximum number of lines for list control.
Example	Gets the maximum number of GList00000 lines in the self screen in Stat. LONG Stat; Stat = GCSListGetMaxListLines(-1, "GList00000");

GCSListAddString	Add list character string
Syntax	GCSListAddString(nWindowNo, strName, pszString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszString : character string to be added
Return value	-1 : addition failed 0 or more : line added
Details	Adds a character string to the end of a list.
Example	Adds the character string "ABCD" to the end of GList00000 in the self screen, and gets the result in Stat. LONG Stat; GCSListAddString(-1, "GList00000", "ABCD");

GCSListInsertString	Insert list character string
Syntax	GCSListInsertString(nWindowNo, strName, nIndex, pszString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nIndex : line No. for insertion (-1: add at the end) (i) STRING pszString : character string to be added
Return value	-1 : addition failed 0 or more : line added
Details	Adds a character string to the specified line in a list. Specifies a line No. to be added to nIndex. When -1 is specified in nIndex, it is added to the end of the list. When the line specified by nIndex does not exist, it is an addition failure.
Example	Adds "ABCD" to the GList00000 list's line No.5 in the self screen, and gets the result in Stat. LONG Stat; Stat = GCSListInsertString(-1, "GList00000", 5, "ABCD");

GCSListGetLineCount	Get number of lines in list
Syntax	GCSListGetLineCount(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Current number of lines in list
Details	Gets the current number of lines in a list.
Example	Gets the number of GList00000 lists in the self screen in Stat. LONG Stat; Stat = GCSListGetLineCount(-1, "GList00000");

GCSListRemoveString	Delete list character string
Syntax	GCSListRemoveString(nWindowNo, strName, nIndex);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) short nIndex : line No. for deletion
Return value	0: deletion failed 1: deletion succeeded
Details	Deletes the character string for the specified line in a list.
Example	Deletes the contents of GList00000 list's line No.5 in the self screen, and gets the result in Stat. LONG Stat; Stat = GCSListRemoveString(-1, "GList00000", 5);

GCSListGetListString	Get list character string
Syntax	GCSListGetListString(nWindowNo, strName, nIndex, pszString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nIndex : acquired number of lines (i) STRING pszString : character string to be acquired
Return value	None
Details	Gets the maximum of 256byte character string for the specified line in the list.
Example	Gets the contents of GList00000 line No.2 in the self screen in strStat. STRING strStat; GCSListGetListString(-1, "GList00000", 2, strStat);

GCSListRemoveAllStrings	Delete all list character strings
Syntax	GCSListRemoveAllStrings(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Deletes all the character strings in a list.
Example	Deletes all the GList00000 list character string in the self screen. GCSListRemoveAllStrings(-1, "GList00000");

GCSListSetCurrentSelect	Set selected line
Syntax	GCSListSetCurrentSelect(nWindowNo, strName, nSel);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nSel : selected line setting
Return value	0: setting failed (illegal setting) 1: setting succeeded
Details	Sets the selected line in a list. When -1 is set in nSel, there is no selected line. When the maximum number of lines is set, 0 is returned as a return value.
Example	Sets 3 for the GList00000 currently selected line in the screen display No.10. GCSListSetCurrentSelect(-1, "GList00000", 3);

GCSListGetCurrentSelect	Get selected line
Syntax	GCSListGetCurrentSelect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0 or more : current selected line -1 : no selection
Details	Gets the selected line in a list.
Example	Gets the GList00000 currently selected line in the self screen in Stat. LONG Stat; Stat = GCSListGetCurrentSelect(-1, "GList00000");

GCSListSetVisibleScrollBarStatus	Set scroll bar display/non-display status

Syntax	GCSListSetVisibleScrollBarStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) short fStatus : display/non-display status
Return value	None
Details	Sets the display/non-display status for the scroll bar. When switching the display/non-display, the object area is registered as a redraw area. One of the following values is set for fStatus: 0: non-display 1: display
Example	Sets 1 for the GList00000 scroll bar display/non-display status in the self screen. GCSListSetVisibleScrollBarStatus(-1, "GList00000", 1);

GCSListGetVisibleScrollBarStatus	Get scroll bar display/non-display status

Syntax	GCSListGetVisibleScrollBarStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: non-display 1: display
Details	Gets the display/non-display status for the scroll bar.
Example	Gets the GList00000 scroll bar display/non-display status in the self screen in Stat. LONG Stat; Stat = GCSListGetVisibleScrollBarStatus(-1, "GList00000");

GCSListSetEnableScrollBarStatus	Set scroll bar controllable or uncontrollable status

Syntax	GCSListSetEnableScrollBarStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : controllable or not controllable status
Return value	None
Details	Sets the controllable or uncontrollable status for the scroll bar. Controllability with mouse input or key input is determined according to this flag. One of the following values is set for fStatus: 0: uncontrollable 1: controllable
Example	Sets 1 for the GList00000 scroll bar controllable/uncontrollable status in the self screen. GCSListSetEnableScrollBarStatus(-1, "GList00000", 1);

GCSListGetEnableScrollBarStatus	Get scroll bar controllable or uncontrollable status

Syntax	GCSListGetEnableScrollBarStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: uncontrollable 1: controllable
Details	Gets the controllable or uncontrollable status for the scroll bar.
Example	Gets the GList00000 scroll bar controllable/uncontrollable status in the self screen in Stat. LONG Stat; Stat = GCSListGetEnableScrollBarStatus(-1, "GList00000");

GCSListSetTopLine	Set list top line
Syntax	GCSListSetTopLine(nWindowNo, strName, lIndex);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lIndex : top line No.
Return value	None
Details	Sets the top line of the list. When a value outside of the range is set, it is changed to the smallest line or the largest line.
Example	Sets 3 for the GList00000 list top line in the self screen. GCSListSetTopLine(-1, "GList00000", 3);

GCSListGetTopLine	Get list top line
Syntax	GCSListGetTopLine(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Top line No.
Details	Gets the current first line in a list.
Example	Gets the GList00000 list top line in the self screen in Stat. LONG Stat; Stat = GCSListGetTopLine(-1, "GList00000");

5.9 Picture

GCSPictureSetDisplay	Set picture display
Syntax	GCSPictureSetDisplay(nWindowNo, strName, ucDisplay);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucDisplay : picture display
Return value	None
Details	Sets the picture display. One of the following values is set for ucDisplay: 0: rectangle display 1: (oval) circle shape display 2: image display When a value outside of the range is set, there is no forced change of the setting. After the picture display is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 display setting in the self screen to 0. GCSPictureSetDisplay(-1, "GPicture00000", 0);

GCSPictureGetDisplay	Get picture display
Syntax	GCSPictureGetDisplay(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: rectangle display 1: (oval) circle shape display 2: image display Besides the above : illegal display setting
Details	Gets the picture display setting. When a setting outside the range is made, that setting value is returned.
Example	Sets the GPicture00000 display setting in the self screen in Stat. LONG Stat; Stat = GCSPictureGetDisplay(-1, "GPicture00000");

GCSPictureSetBorderID	Set 3D border resource ID
Syntax	GCSPictureSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 3D border resource ID in the self screen to 1. GCSPictureSetBorderID(-1, "GPicture00000", 1);

GCSPictureGetBorderID	Get 3D border resource ID
Syntax	GCSPictureGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GPicture00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSPictureGetBorderID(-1, "GPicture00000");

GCSPictureSetFontID	Set font resource ID
Syntax	GCSPictureSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 font resource ID in the self screen to 1. GCSPictureSetFontID(-1, "GPicture00000", 1);

GCSPictureGetFontID	Get font resource ID
Syntax	GCSPictureGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GPicture00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSPictureGetFontID(-1, "GPicture00000");

GCSPictureSetStringID	Set caption character string resource ID
Syntax	GCSPictureSetStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 caption character string resource ID in the self screen to 1. GCSPictureSetStringID(-1, "GPicture00000", 1);

GCSPictureGetStringID	Get caption character string resource ID
Syntax	GCSPictureGetStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
Example	Gets the GPicture00000 caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSPictureGetStringID(-1, "GPicture00000");

GCSPictureSetCaption	Set caption information
Syntax	GCSPictureSetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmCaption : global memory for caption information
Return value	None
Details	Sets the caption information. After the caption information is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 caption information in the self screen as follows: Color : White (0xffffff) Horizontal position : 0 Vertical position : 1 Left margin : 10 Right margin : 0 Top margin : 0 Bottom margin : 0 mem = GMEMCreate("TESTMEM", 14); GMEMSetLong(mem, 0, HFFFFFF); GMEMSetChar(mem, 4, 0); GMEMSetChar(mem, 5, 1); GMEMSetShort(mem, 6, 10); GMEMSetShort(mem, 8, 0); GMEMSetShort(mem, 10, 0); GMEMSetShort(mem, 12, 0); GCSPictureSetCaption(-1, "GPicture00000", mem); GMEMDelete(mem);

GCSPictureGetCaption	Get caption information
Syntax	GCSPictureGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Stores acquired caption information in gmCaption.
Details	Gets the caption information.

Example	Gets the GPicture00000 caption information in the self screen as follows:
	nCol : Color
	nHPos : Horizontal position
	nVPos : Vertical position
	nLMgn : Left margin
	nRMgn : Right margin
	nTMgn : Top margin
	nBMgn : Bottom margin
	mem = GMEMCreate("TESTMEM", 14);
	GCSPictureGetCaption(-1, "GPicture00000", mem);
	nCol = GMEMGetLong(mem, 0);
	nHPos = GMEMGetChar(mem, 4);
	nHPos = GMEMGetChar(mem, 5);
	nLMgn = GMEMGetShort(mem, 6);
	nRMgn = GMEMGetShort(mem, 8);
	nTMgn = GMEMGetShort(mem, 10);
	nBMgn = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSPictureSetStatus	Set object display status
Syntax	GCSPictureSetStatus(nWindowNo, strName, ucStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucStatus : object status setting
Return value	None
Details	Sets the GCPicture object display status. After the object display status is set, the control area is registered as a redraw area.
Example	Sets the GPicture00000 display status in the self screen to 3. GCSPictureSetStatus(-1, "GPicture00000", 3);

GCSPictureGetStatus	Get object display status
Syntax	GCSPictureGetStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	None
Details	Gets the GCPicture object display status.
Example	Gets the GPicture00000 display status in the self screen in Stat. LONG Stat; Stat = GCSPictureGetStatus(-1, "GPicture00000");

5.10 ProgressBar

GCSProgressbarSetBackGroundDesign	Set background design

Syntax	GCSProgressbarSetBackGroundDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the background design for the progress bar. After the background design is set, the control area is registered as a redraw area.
Example	Sets the GProgressBar00000 background design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2

```
mem = GMEMCreate("TESTMEM", 16);
GMEMSetShort(mem, 0, 4);
GMEMSetLong(mem, 4, HFFFFFF);
GMEMSetLong(mem, 8, H000000);
GMEMSetLong(mem, 12, 2);
GCSProgressbarSetBackGroundDesign(-1,
"GProgressBar00000", mem);
GMEMDelete(mem);
```

GCSProgressbarGetBackGroundDesign	Get background design

Syntax	GCSProgressbarGetBackGroundDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for desgin
Return value	Stores acquired design in gmDesign.
Details	Gets the background design for the progress bar.
Example	Gets the GProgressBar00000 background design in the self screen as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSProgressbarGetBackGroundDesign(-1,
"GProgressBar00000", mem);
nPat = GMEMGetShort(mem, 0);
nFCol = GMEMGetLong(mem, 4);
nBCol = GMEMGetLong(mem, 8);
nImg = GMEMGetShort(mem, 12);
GMEMDelete(mem);
```

GCSProgressbarSetBarDesign	Set bar design

Syntax

GCSProgressbarSetBarDesign(nWindowNo, strName, gmDesign);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmDesign : global memory for design

Return value

None

Details

Sets the design for the progress bar.
After the bar design is set, the control area is registered as a redraw area.

Example

Sets the GProgressBar00000 design in the self screen as follows:
Fill pattern : 4
Foreground color : 0xffffff
Background color : 0x000000
Image resource ID : 2

```
mem = GMEMCreate("TESTMEM", 16);
GMEMSetShort(mem, 0, 4);
GMEMSetLong(mem, 4, HFFFFFF);
GMEMSetLong(mem, 8, H000000);
GMEMSetLong(mem, 12, 2);
GCSProgressbarSetBarDesign(-1, "GProgressBar00000", mem);
GMEMDelete(mem);
```

GCSProgressbarGetBarDesign	Get bar design

Syntax	GCSProgressbarGetBarDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Stores acquired design in gmDesign.
Details	Gets the design for the progress bar.

Example	Gets the GProgressBar00000 design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSProgressbarGetBarDesign(-1, "GProgressBar00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSProgressbarSetRange	Set progress bar range

Syntax	GCSProgressbarSetRange(nWindowNo, strName, lMin, lMax);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG lMin : minimum value (i) LONG lMax : maximum value
Return value	None
Details	Sets the range for the progress bar. When the smallest value and largest value settings are inverted, they are reversed.
Example	Sets the GProgressBar00000's minimum value to 0 and the maximum value to 10 in the self screen. GCSProgressbarSetRange(-1, "GProgressBar00000", 0, 10);

GCSProgressbarGetRange	Get progress bar range

Syntax	GCSProgressbarGetRange(nWindowNo, strName, gmRange);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmRange : global memory for range
Return value	None
Details	Gets the range for the progress bar.

Example	Gets the GProgressBar00000 range in the self screen as follows:
	nMin : Minimum value
	nMax : Maximum value
	SHORT nMin;
	SHORT nMax;
	mem = GMEMCreate("TESTMEM", 4);
	GCSProgressbarGetRange(-1, "GProgressBar00000", mem);
	nMin = GMEMGetShort(mem, 0);
	nMax = GMEMGetShort(mem, 2);
	GMEMDelete(mem);

GCSProgressbarSetDirection	Set fill direction

Syntax

GCSProgressbarSetDirection(nWindowNo, strName, ucDirection);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG ucDirection : fill direction

Return value

None

Details

Sets the fill direction for the progress bar.
One of the followings is set for ucDirection:
0: left to right
1: right to left
2: top to bottom
3: bottom to top
When a value outside of the range is set, there is no forced change of the setting.
After the fill direction is set, the control area is registered as a redraw area.

Example

Sets the GProgressBar00000 fill direction in the self screen to 0.

```
GCSProgressbarSetDirection(-1, "GProgressBar00000", 0);
```

GCSProgressbarGetDirection	Get fill direction

Syntax

GCSProgressbarGetDirection(nWindowNo, strName);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name

Return value

0: left to right
1: right to left
2: top to bottom
3: bottom to top
Besides the above: illegal display setting

Details

Gets the fill direction setting.

Example

Gets the GProgressBar00000 fill direction in the self screen in Stat.

```
Stat = GCSProgressbarGetDirection(-1, "GProgressBar00000");
```

GCSProgressbarSetValue	Set current progress bar value
Syntax	GCSProgressbarSetValue(nWindowNo, strName, nValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nValue : current value
Return value	None
Details	Sets the current value for the progress bar. The permissible range for the setting is the range specified by SetRange, and values outside of the range become the maximum value. or the minimum value.
Example	Sets 10 for the GProgressBar00000 current value in the self screen. GCSProgressbarSetValue(-1, "GProgressBar00000", 10);

GCSProgressbarGetValue	Get current progress bar value
Syntax	GCSProgressbarGetValue(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Current value for the progress bar
Details	Gets the current value for the progress bar.
Example	Gets the GProgressBar00000 current value in the self screen in Stat. GCSProgressbarGetValue(-1, "GProgressBar00000", 10);

5.11 RadioButton

GCSRadiobuttonSetBorderID	Set 3D border resource ID

Syntax	GCSRadiobuttonSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 3D border resource ID in the self screen to 1. GCSRadiobuttonSetBorderID(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetBorderID	Get 3D border resource ID

Syntax	GCSRadiobuttonGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
Example	Gets the GRadioButton00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetBorderID(-1, "GRadioButton00000");

GCSRadiobuttonSetFontID	Set font resource ID
Syntax	GCSRadiobuttonSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 font resource ID in the self screen to 1. GCSRadiobuttonSetFontID(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetFontID	Get font resource ID
Syntax	GCSRadiobuttonGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GRadioButton00000 font resource ID in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetFontID(-1, "GRadioButton00000");

GCSRadiobuttonSetStringID	Set caption character string resource ID

Syntax	GCSRadiobuttonSetStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 caption character string resource ID in the self screen to 1. GCSRadiobuttonSetStringID(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetStringID	Get caption character string resource ID

Syntax	GCSRadiobuttonGetStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
Example	Gets the GRadioButton00000 caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetStringID(-1, "GRadioButton00000");

GCSRadiobuttonSetCaption	Set caption information

Syntax

GCSRadiobuttonSetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmCaption : global memory for caption information

Return value

None

Details

Sets the caption information.
All except character color are disabled.
After the caption information is set, the control area is registered as a redraw area.

Example

Sets the GRadioButton00000 caption information in the self screen as follows:
Color : White (0xffffff)
Horizontal position : 0
Vertical position : 1
Left margin : 10
Rihgt margin : 0
Top margin : 0
Bottom margin : 0

```
mem = GMEMCreate("TESTMEM", 14);
GMEMSetLong(mem, 0, HFFFFFF);
GMEMSetChar(mem, 4, 0);
GMEMSetChar(mem, 5, 1);
GMEMSetShort(mem, 6, 10);
GMEMSetShort(mem, 8, 0);
GMEMSetShort(mem, 10, 0);
GMEMSetShort(mem, 12, 0);
GCSRadiobuttonSetCaption(-1, "GRadioButton00000", mem);
GMEMDelete(mem);
```

GCSRadiobuttonGetCaption	Get caption information

Syntax

GCSRadiobuttonGetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(o) GMEM gmCaption : global memory for caption information

Return value

Stores acquired caption information in gmCaption.

Details

Gets the caption information setting.
All settings except character color are disabled.

Example

Gets the GRadioButton00000 caption information in the self screen as follows:

```
 nCol : Color
 nHPos : Horizontal position
 nVPos : Vertical position
 nLMgn : Left margin
 nRMgn : Right margin
 nTMgn : Top margin
 nBMgn : Bottom margin
mem = GMEMCreate("TESTMEM", 14);
GCSRadiobuttonGetCaption(-1, "GRadioButton00000", mem);
nCol = GMEMGetLong(mem, 0);
nHPos = GMEMGetChar(mem, 4);
nHPos = GMEMGetChar(mem, 5);
nLMgn = GMEMGetShort(mem, 6);
nRMgn = GMEMGetShort(mem, 8);
nTMgn = GMEMGetShort(mem, 10);
nBMgn = GMEMGetShort(mem, 12);
GMEMDelete(mem);
```

GCSRadiobuttonSetFocusEffect	Set effect during focus

Syntax	GCSRadiobuttonSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect is set, the control area is registered as a redraw
Example	Sets the GRadioButton00000 effect during focus in the self screen to 1. GCSRadiobuttonSetFocusEffect(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetFocusEffect	Get effect during focus

Syntax	GCSRadiobuttonGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GRadioButton00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetFocusEffect(-1, "GRadioButton00000");

GCSRadiobuttonSetStatus	Set object status
Syntax	GCSRadiobuttonSetStatus(nWindowNo, strName, ucStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucStatus : set object status
Return value	None
Details	Sets the GCRadioButton object status. One of the following values is set for ucStatus: 0: OFF status 1: ON status When ON status is set, turn the control in the same screen that sets the same group No. using the other radio button control to OFF status. After the object status is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 ON/OFF status in the self screen to 1. GCSRadiobuttonSetStatus(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetStatus	Get object status
Syntax	GCSRadiobuttonGetStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Object status 0: OFF status 1: ON status
Details	Gets the GCRadioButton object status.
Example	Gets the GRadioButton00000 ON/OFF status in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetStatus(-1, "GRadioButton00000");

GCSRadiobuttonSetBoxSize	Set box size

Syntax	GCSRadiobuttonSetBoxSize(nWindowNo, strName, ucBoxSize);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucBoxSize : box size
Return value	None
Details	Sets the box size. After the box size is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 box size in the self screen to 10. GCSRadiobuttonSetBoxSize(-1, "GRadioButton00000", 10);

GCSRadiobuttonGetBoxSize	Get box size

Syntax	GCSRadiobuttonGetBoxSize(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box size
Details	Gets the box size settings.
Example	Gets the GRadioButton00000 box size in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetBoxSize(-1, "GRadioButton00000");

GCSRadiobuttonSetBoxColor	Set box color

Syntax	GCSRadiobuttonSetBoxColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : box color
Return value	None
Details	Sets the box color. After the box color is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 box color in the self screen to white (0xffffff). GCSRadiobuttonSetBoxColor(-1, "GRadioButton00000", HFFFFFF);

GCSRadiobuttonGetBoxColor	Get box color

Syntax	GCSRadiobuttonGetBoxColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box color
Details	Gets the box color.
Example	Gets the GRadioButton00000 box color in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetBoxColor(-1, "GRadioButton00000");

GCSRadiobuttonSetFocusColor	Set background color during focus

Syntax	GCSRadiobuttonSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : background color during focus
Return value	None
Details	Sets the background color in the case where the effect during focus is "change design". After the background color is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 background color during focus in the self screen to white (0xffffff). GCSRadiobuttonSetFocusColor(-1, "GRadioButton00000", HFFFFFF);

GCSRadiobuttonGetFocusColor	Get background color during focus

Syntax	GCSRadiobuttonGetFocusColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Background color during focus
Details	Gets the background color in the case where the effect during focus is "change design".
Example	Gets the GRadioButton00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetFocusColor(-1, "GRadioButton00000");

GCSRadiobuttonSetDisableCaptionColor	Set character color when disabled

Syntax	GCSRadiobuttonSetDisableCaptionColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GColor gcColor : character color when disabled
Return value	None
Details	Sets the character color during disabled control. After the character color is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 character color when disabled in the self screen to white (0xffffff). GCSRadiobuttonSetDisableCaptionColor(-1, "GRadioButton00000", HFFFFFF);

GCSRadiobuttonGetDisableCaptionColor	Get character color when diabled

Syntax	GCSRadiobuttonGetDisableCaptionColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Character color when disabled
Details	Gets the character color during disabled control.
Example	Gets the GRadioButton00000 character color when disabled in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetDisableCaptionColor(-1, "GRadioButton00000");

GCSRadiobuttonSetDisableBoxColor	Set box color when disabled

Syntax	GCSRadiobuttonSetDisableBoxColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : box color
Return value	None
Details	Sets the box color during disabled control. After the box color is set, the control area is registered as a redraw area.
Example	Sets the GRadioButton00000 box color when disabled in the self screen to white (0xffffff). GCSRadiobuttonSetDisableBoxColor(-1, "GRadioButton00000", HFFFFFF);

GCSRadiobuttonGetDisableBoxColor	Get box color when disabled

Syntax	GCSRadiobuttonGetDisableBoxColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Box color when disabled
Details	Gets the box color during disabled control.
Example	Gets the GRadioButton00000 box color when disabled in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetDisableBoxColor(-1, "GRadioButton00000");

GCSRadiobuttonSetRadioGroup	Set radio group No.
Syntax	GCSRadiobuttonSetRadioGroup(nWindowNo, strName, usRadioGroup);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usRadioGroup : radio group No.
Return value	None
Details	Sets the group No. for radio buttons. The radio button control in the screen having the same group No. carries out an exclusion operation.
Example	Sets the GRadioButton00000 group No. in the self screen to 1. GCSRadiobuttonSetRadioGroup(-1, "GRadioButton00000", 1);

GCSRadiobuttonGetRadioGroup	Get radio group No.
Syntax	GCSRadiobuttonGetRadioGroup(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Radio group No.
Details	Gets the group No. for radio buttons.
Example	Gets the GRadioButton00000 group No. in the self screen in Stat. LONG Stat; Stat = GCSRadiobuttonGetRadioGroup(-1, "GRadioButton00000");

5.12 ScrollBar

GCSScrollbarexSetDisplay	Set display type
Syntax	GCSScrollbarexSetDisplay(nWindowNo, strName, ucDisplay);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucDisplay : display type
Return value	None
Details	Sets the display type. One of the following values is set for ucDisplay: 0: normal display 1: image display When a value outside of the range is set, there is no forced change of the setting. After the display type is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 display type in the self screen to 1. GCSScrollbarexSetDisplay(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexGetDisplay	Get display type
Syntax	GCSScrollbarexGetDisplay(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display 1: image display Besides the above: illegal display setting
Details	Gets the display type setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GScrollBarEx00000 display type in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetDisplay(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetDirection	Set scroll bar direction

Syntax	GCSScrollbarexSetDirection(nWindowNo, strName, ucDirection);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucDirection : scroll bar direction
Return value	None
Details	Sets the scroll bar direction. One of the following values is set for ucDisplay: 0: vertical direction 1: horizontal direction When a value outside of the range is set, there is no forced change of the setting. After the scroll bar direction is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 direction in the self screen to 1. GCSScrollbarexSetDirection(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexGetDirection	Get scroll bar direction

Syntax	GCSScrollbarexGetDirection(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: vertical direction 1: horizontal direction Besides the above: illegal display setting
Details	Gets the scroll bar direction setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GScrollBarEx00000 direction in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetDirection(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetPageSize	Set page size

Syntax	GCSScrollbarexSetPageSize(nWindowNo, strName, ulPageSize);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulPageSize : page size
Return value	None
Details	Sets the page size. After the page size is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 page size in the self screen to 10. GCSScrollbarexSetPageSize(-1, "GScrollBarEx00000", 10) ;

GCSScrollbarexGetPageSize	Get page size

Syntax	GCSScrollbarexGetPageSize(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Page size
Details	Gets the page size.
Example	Gets the GScrollBarEx00000 page size in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetPageSize(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetScrollArrowColor	Set scroll arrow color

Syntax	GCSScrollbarexSetScrollArrowColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for scroll arrow color
Return value	None
Details	Sets the scroll bar arrow color. After the scroll bar arrow color is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 scroll arrow color in the self screen to white (0xffffff). GCSScrollbarexSetScrollArrowColor(-1, "GScrollBarEx00000", HFFFFFF) ;

GCSScrollbarexGetScrollArrowColor	Get scroll arrow color

Syntax	GCSScrollbarexGetScrollArrowColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code
Details	Gets the scroll bar arrow color.
Example	Gets the GScrollBarEx00000 scroll arrow color in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetScrollArrowColor(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetScrollPosition	Set scroll bar current position
Syntax	GCSScrollbarexSetScrollPosition(nWindowNo, strName, nPosition);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nPosition : scroll bar current position
Return value	None
Details	Sets the scroll bar current position. When a value that exceeds the range of the property "Scroll maximum" and "Scroll minimum" is set, the set value is set in the property "Scroll maximum" and "Scroll minimum".
Example	Sets the GScrollBarEx00000 scroll bar current position in the self screen to 50. GCSScrollbarexSetScrollPosition(-1, "GScrollBarEx00000", 50);

GCSScrollbarexGetScrollPosition	Get scroll bar current position
Syntax	GCSScrollbarexGetScrollPosition(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Current position
Details	Gets the scroll bar current position.
Example	Gets the GScrollBarEx00000 scroll bar current position in the self screen in nPosition. LONG nPosition; nPosition = GCSScrollbarexGetScrollPosition(-1,"GScrollBarEx00000 ");

GCSScrollbarexSetPrevBtnOnImgID	Set design (upper) resource ID when button is ON

Syntax	GCSScrollbarexSetPrevBtnOnImgID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : design (upper) resource ID when button is ON
Return value	None
Details	Sets the design (upper) resource ID when the button is ON. After the design resource ID is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 design (upper) resource ID in the self screen when the button is ON to 1. GCSScrollbarexSetPrevBtnOnImgID(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexSetPrevBtnOffImgID	Set design (upper) resource ID when button is OFF

Syntax	GCSScrollbarexSetPrevBtnOffImgID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : design (upper) resource ID when button is OFF
Return value	None
Details	Sets the design (upper) resource ID when the button is OFF. After the design resource ID is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 design (upper) resource ID in the self screen when the button is OFF to 1. GCSScrollbarexSetPrevBtnOffImgID(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexGetPrevBtnOnImgID	Get design (upper) resource ID when button is ON

Syntax	GCSScrollbarexGetPrevBtnOnImgID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Design (upper) resource ID when button is ON
Details	Gets the design (upper) resource ID setting when the button is ON.
Example	Gets the GScrollBarEx00000 design (upper) resource ID in the self screen when the button is ON in Stat. LONG Stat; Stat = GCSScrollbarexGetPrevBtnOnImgID(-1, "GScrollBarEx00000") ;

GCSScrollbarexGetPrevBtnOffImgID	Get design (upper) resource ID when button is OFF

Syntax	GCSScrollbarexGetPrevBtnOffImgID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Design (upper) resource ID when button is OFF
Details	Gets the design (upper) resource ID setting when the button is OFF.
Example	Gets the GScrollBarEx00000 design (upper) resource ID in the self screen when the button is OFF in Stat. LONG Stat; Stat = GCSScrollbarexGetPrevBtnOffImgID(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetNextBtnOnImgID	Set design (lower) resource ID when button is ON

Syntax	GCSScrollbarexSetNextBtnOnImgID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : design (lower) resource ID when button is ON
Return value	None
Details	Sets the design (lower) resource ID when the button is ON. After the design resource ID is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 design (lower) resource ID in the self screen when the button is ON to 1. GCSScrollbarexSetNextBtnOnImgID(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexSetNextBtnOffImgID	Set design (lower) resource ID when button is OFF

Syntax	GCSScrollbarexSetNextBtnOffImgID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : design (lower) resource ID when button is OFF
Return value	None
Details	Sets the design (lower) resource ID when the button is OFF. After the design resource ID is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 design (lower) resource ID in the self screen when the button is OFF to 1. GCSScrollbarexSetNextBtnOffImgID(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexGetNextBtnOnImgID	Get design (lower) resource ID when button is ON

Syntax	GCSScrollbarexGetNextBtnOnImgID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Design (lower) resource ID when button is ON
Details	Gets the design (lower) resource ID setting when the button is ON.
Example	Gets the GScrollBarEx00000 design (lower) resource ID in the self screen when the button is ON in Stat. LONG Stat; Stat = GCSScrollbarexGetNextBtnOnImgID(-1, "GScrollBarEx00000") ;

GCSScrollbarexGetNextBtnOffImgID	Get design (lower) resource ID when button is OFF

Syntax	GCSScrollbarexGetNextBtnOffImgID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Design (lower) resource ID when button is OFF
Details	Gets the design (lower) resource ID setting when the button is OFF.
Example	Gets the GScrollBarEx00000 design (lower) resource ID in the self screen when the button is OFF in Stat. LONG Stat; Stat = GCSScrollbarexGetNextBtnOffImgID(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetBackGroundColor	Set background color

Syntax	GCSScrollbarexSetBackGroundColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color
Return value	None
Details	Sets the background color. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GScrollBarEx00000 backgroud color in the self screen to white (0xffffff). GCSScrollbarexSetBackGroundColor(-1, "GScrollBarEx00000", HFFFFFF) ;

GCSScrollbarexGetBackGroundColor	Get background color

Syntax	GCSScrollbarexGetBackGroundColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code
Details	Gets the background color.
Example	Gets the GScrollBarEx00000 backgroud color in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetBackGroundColor(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetOutLineBorderID	Set outline 3D border

Syntax	GCSScrollbarexSetOutLineBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the outline 3D border.
Example	Sets the GScrollBarEx00000 3D border resource ID in the self screen to 1. GCSScrollbarexSetOutLineBorderID(-1, "GscrollBarEx00000", 1) ;

GCSScrollbarexGetOutLineBorderID	Get outline 3D border

Syntax	GCSScrollbarexGetOutLineBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Outline 3D border resource ID
Details	Gets outline 3D border resource ID.
Example	Gets the GScrollBarEx00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetOutLineBorderID(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetPinchWidth	Set pinch width

Syntax	GCSScrollbarexSetPinchWidth(nWindowNo, strName, nWidth);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nWidth : pinch width
Return value	None
Details	Sets the pinch width.
Example	Sets the GScrollBarEx00000 pinch width in the self screen to 5. GCSScrollbarexSetPinchWidth(-1, "GScrollBarEx00000", 5) ;

GCSScrollbarexGetPinchWidth	Get pinch width

Syntax	GCSScrollbarexGetPinchWidth(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Pinch width
Details	Gets the pinch width.
Example	Gets the GScrollBarEx00000 pinch width in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetPinchWidth(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetPinchColor	Set pinch color

Syntax	GCSScrollbarexSetPinchColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : pinch color
Return value	None
Details	Sets the pinch color.
Example	Sets the GScrollBarEx00000 pinch color in the self screen to white (0xffffff). GCSScrollbarexSetPinchColor(-1, "GScrollBarEx00000", HFFFFFF) ;

GCSScrollbarexGetPinchColor	Get pinch color

Syntax	GCSScrollbarexGetPinchColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Pinch color
Details	Gets the pinch color.
Example	Gets the GScrollBarEx00000 pinch color in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetPinchColor(-1, "GScrollBarEx00000") ;

GCSScrollbarexSetBarEnableStatus	Set bar display/non-display

Syntax	GCSScrollbarexSetBarEnableStatus(nWindowNo, strName, nStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) short nStatus : display setting
Return value	None
Details	Sets the display/non-display for the bar. One of the following values is set for nStatus: 0: display 1: non-display
Example	Sets the GScrollBarEx00000 bar display/non-display status in the self screen to 1. GCSScrollbarexSetBarEnableStatus(-1, "GScrollBarEx00000", 1) ;

GCSScrollbarexGetBarEnableStatus	Get bar display/non-display

Syntax	GCSScrollbarexGetBarEnableStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: non-display 1: display
Details	Gets the display/non-display setting for the bar.
Example	Gets the GScrollBarEx00000 bar display/non-display status in the self screen in Stat. LONG Stat; Stat = GCSScrollbarexGetBarEnableStatus(-1, "GScrollBarEx00000") ;

5.13 TextBox

GCSTextboxSetBorderID	Set 3D border resource ID
Syntax	GCSTextboxSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : Screen No. (Specify –1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 3D border resource ID in the self screen to 1. GCSTextboxSetBorderID(-1, "GTextBox00000", 1) ;

GCSTextboxGetBorderID	Get 3D border resource ID
Syntax	GCSTextboxGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GTextBox00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetBorderID(-1, "GTextBox00000") ;

GCSTextboxSetFontID	Set font resource ID
Syntax	GCSTextboxSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 font resource ID in the self screen to 1. GCSTextboxSetFontID(-1, "GTextBox00000", 1) ;

GCSTextboxGetFontID	Get font resource ID
Syntax	GCSTextboxGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GTextBox00000 font resource ID in the self screen in Stat. Stat = GCSTextboxGetFontID(-1, "GTextBox00000") ;

GCSTextboxSetCaption	Set caption information
Syntax	GCSTextboxSetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmCaption : global memory for caption information
Return value	None
Details	Set the caption information. After the caption information is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 caption information in the self screen as follows: Color : White (0xffffff) Horizontal position : 0 Vertical position : 1 Left margin : 10 Right margin : 0 Top margin : 0 Bottom margin : 0 mem = GMEMCreate("TESTMEM", 14) ; GMEMSetLong(mem, 0, HFFFFFF) ; GMEMSetChar(mem, 4, 0) ; GMEMSetChar(mem, 5, 1) ; GMEMSetShort(mem, 6, 10) ; GMEMSetShort(mem, 8, 0) ; GMEMSetShort(mem, 10, 0) ; GMEMSetShort(mem, 12, 0) ; GCSTextboxSetCaption(-1, "GTextBox00000", mem) ; GMEMDelete(mem);

GCSTextboxGetCaption	Get caption information
Syntax	GCSTextboxGetCaption(nWindowNo, strName, gmCaption);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmCaption : global memory for caption information
Return value	Stores acquired caption information in gmCaption.
Details	Gets the caption information.
Example	Gets the GTextBox00000 caption information in the self screen as follows: nCol : Color nHPos : Horizontal position nVPos : Vertical position nLMgn : Left margin nRMgn : Right margin nTMgn : Top margin nBMgn : Bottom margin mem = GMEMCreate("TESTMEM", 14) ; GCSTextboxGetCaption(-1, "GTextBox00000", mem) ; nCol = GMEMGetLong(mem, 0) ; nHPos = GMEMGetChar(mem, 4) ; nHPos = GMEMGetChar(mem, 5) ; nLMgn = GMEMGetShort(mem, 6) ; nRMgn = GMEMGetShort(mem, 8) ; nTMgn = GMEMGetShort(mem, 10) ; nBMgn = GMEMGetShort(mem, 12) ; GMEMDelete(mem);

GCSTextboxSetFocusEffect	Set effect during focus

Syntax	GCSTextboxSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 effect during focus in the self screen to 1. GCSTextboxSetFocusEffect(-1, "GTextBox00000", 1) ;

GCSTextboxGetFocusEffect	Get effect during focus

Syntax	GCSTextboxGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GTextBox00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetFocusEffect(-1, "GTextBox00000") ;

GCSTextboxSetFocusColor	Set background color during focus

Syntax	GCSTextboxSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color during
Return value	None
Details	Sets the background color during focus. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 background color during focus in the self screen to white (0xffffff). GCSTextboxSetFocusColor(-1, "GTextBox00000", HFFFFFF);

GCSTextboxGetFocusColor	Get background color during focus

Syntax	GCSTextboxGetFocusColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color during focus
Details	Gets the background color during focus.
Example	Gets the GTextBox00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetFocusColor(-1, "GTextBox00000") ;

GCSTextboxSetNormalColor	Set normal background color

Syntax	GCSTextboxSetNormalColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 normal background color in self screen to white (0xffffff). GCSTextboxSetNormalColor(-1, "GTextBox00000", HFFFFFF) ;

GCSTextboxGetNormalColor	Get normal background color

Syntax	GCSTextboxGetNormalColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for normal background color
Details	Gets the normal background color.
Example	Gets the GTextBox00000 normal background color in self screen in Stat. LONG Stat; Stat = GCSTextboxGetNormalColor(-1, "GTextBox00000") ;

GCSTextboxSetDisableColor	Set background color when disabled

Syntax	GCSTextboxSetDisableColor(nWindowNo, strName, gcColor);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG gcColor : color code for background color when
Return value	None
Details	Sets the background color when disabled. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 background color when disabled in the self screen to white (0xffffff). GCSTextboxSetDisableColor(-1, "GTextBox00000", HFFFFFF) ;

GCSTextboxGetDisableColor	Get background color when disabled

Syntax	GCSTextboxGetDisableColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
Example	Gets the GTextBox00000 background color when disabled in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetDisableColor(-1, "GTextBox00000") ;

GCSTextboxSetTextType	Set character string type
Syntax	GCSTextboxSetTextType(nWindowNo, strName, ucType);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucType : character string type (Setting value of the "Type" property)
Return value	None
Details	Sets the character string type. One of the following values is set for ucType: GTEXT_TYPE_STRING(0) : character string GTEXT_TYPE_SHORT(1) : short value GTEXT_TYPE_USHORT(2) : unsigned short value GTEXT_TYPE_LONG(3) : long value GTEXT_TYPE_ULONG(4) : unsigned long value GTEXT_TYPE_FLOAT(5) : float value GTEXT_TYPE_DOUBLE(6) : double value After the character string type is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 display character string type in the self screen to 1. GCSTextboxSetTextType(-1, "GTextBox00000", 1) ;

(Note) When the character string type is changed, also change the "Display Format" property to suit the character string type.
If the character string type does not match the "Display Format" property, the character string type is not applied.

GCSTextboxGetTextType	Get character string type
Syntax	GCSTextboxGetTextType(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	GTEXT_TYPE_STRING(0) : character string GTEXT_TYPE_SHORT(1) : short value GTEXT_TYPE_USHORT(2) : unsigned short value GTEXT_TYPE_LONG(3) : long value GTEXT_TYPE_ULONG(4) : unsigned long value GTEXT_TYPE_FLOAT(5) : float value GTEXT_TYPE_DOUBLE(6) : double value
Details	Gets the display character string type (the "Type" property).
Example	Gets the GTextBox00000 display character string type in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetTextType(-1, "GTextBox00000") ;

GCSTextboxSetFormatID	Set character string display format resource ID
Syntax	GCSTextboxSetFormatID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : display format character string resource
Return value	None
Details	Sets the character string resource ID used as the format when display in a text box. After the character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 character string resource ID in the self screen to 1. GCSTextboxSetFormatID(-1, "GTextBox00000", 1) ;

(Note) If the display format is changed, also change the character string type to the appropriate value.
If the character string type does not match the "Display Format" property, the display format is not applied.

GCSTextboxGetFormatID	Get character string display format resource ID
Syntax	GCSTextboxGetFormatID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Character string resource ID
Details	Gets the character string resource ID used as the display format.
Example	Gets the GTextBox00000 character string resource ID (The "Display Format" property) in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetFormatID(-1, "GTextBox00000") ;

GCSTextboxSetPasswordStatus	Set password display status

Syntax	GCSTextboxSetPasswordStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : password display status
Return value	None
Details	Sets whether the password will be displayed for text boxes or not. One of the following values is set for fStatus: 0: normal display status 1: password display status After the password display status is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 password display status in the self screen to 1. GCSTextboxSetPasswordStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetPasswordStatus	Get password display status

Syntax	GCSTextboxGetPasswordStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: password display status
Details	Gets whether the test box display status is normal display status or the password display status.
Example	Gets the GTextBox00000 password display status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetPasswordStatus(-1, "GTextBox00000") ;

GCSTextboxSetCommaStatus	Set comma display status

Syntax	GCSTextboxSetCommaStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : comma display status
Return value	None
Details	Sets whether commas will be displayed for text boxes or not. One of the following values is set for fStatus: 0: normal display status 1: comma display status After the comma display status is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 comma display status in the self screen to 1. GCSTextboxSetCommaStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetCommaStatus	Get comma display status

Syntax	GCSTextboxGetCommaStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: comma display status
Details	Gets the setting as to whether the test box display status is normal display status or the comma display status.
Example	Gets the GTextBox00000 comma display status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetCommaStatus(-1, "GTextBox00000") ;

GCSTextboxSetRefuseInputNumberStatus	Set numerical input disabled status

Syntax	GCSTextboxSetRefuseInputNumberStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : number input status
Return value	None
Details	Sets whether numerical input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GTextBox00000 numerical input disabled status in the self screen to 1. GCSTextboxSetRefuseInputNumberStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetRefuseInputNumberStatus	Get numerical input disabled status

Syntax	GCSTextboxGetRefuseInputNumberStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets the setting as to whether numerical input will be rejected for text boxes or not.
Example	Gets the GTextBox00000 numerical input disabled status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputNumberStatus(-1, "GTextBox00000") ;

GCSTextboxSetRefuseInputSmallLetterStatus	Set one-byte lower case character input disabled status

Syntax	GCSTextboxSetRefuseInputSmallLetterStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : one-byte lower case character input
Return value	None
Details	Sets whether one-byte lower case character input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GTextBox00000 one-byte small letter input disabled status in the self screen to 1. GCSTextboxSetRefuseInputSmallLetterStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetRefuseInputSmallLetterStatus	Get one-byte lower case character input disabled status

Syntax	GCSTextboxGetRefuseInputSmallLetterStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets the setting as to whether one-byte lower case character input will be rejected for text boxes or not.
Example	Gets the GTextBox00000 one-byte small letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputSmallLetterStatus(-1, "GTextBox00000") ;

GCSTextboxSetRefuseInputCapitalLetterStatus	Set one-byte upper case character input disabled status

Syntax	GCSTextboxSetRefuseInputCapitalLetterStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : one-byte upper case character input
Return value	None
Details	Sets whether one-byte upper case character input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GTextBox00000 one-byte capital letter input disabled status in the self screen to 1. GCSTextboxSetRefuseInputCapitalLetterStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetRefuseInputCapitalLetterStatus	Get one-byte upper case character input disabled status

Syntax	GCSTextboxGetRefuseInputCapitalLetterStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets whether one-byte upper case character input will be rejected for text boxes or not.
Example	Gets the GTextBox00000 one-byte capital letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputCapitalLetterStatus(-1, "GTextBox00000") ;

GCSTextboxSetRefuseInputSymbolLetterStatus	Set one-byte symbol input disabled status

Syntax	GCSTextboxSetRefuseInputSymbolLetterStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : one-byte symbol input status
Return value	None
Details	Sets whether one-byte symbol input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GTextBox00000 one-byte symbol letter input disabled status in the self screen to 1. GCSTextboxSetRefuseInputSymbolLetterStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetRefuseInputSymbolLetterStatus	Get one-byte symbol input disabld status

Syntax	GCSTextboxGetRefuseInputSymbolLetterStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets whether one-byte symbol input will be rejected for text boxes or not.
Example	Gets the GTextBox00000 one-byte symbol letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputSymbolLetterStatus(-1, "GTextBox00000") ;

GCSTextboxSetRefuseInputFullLetterStatus	Set two-byte character input disabled status

Syntax	GCSTextboxSetRefuseInputFullLetterStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : two-byte character input status
Return value	None
Details	Sets whether two-byte character input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GTextBox00000 two-byte letter input disabled status in the self screen to 1. GCSTextboxSetRefuseInputFullLetterStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetRefuseInputFullLetterStatus	Get two-byte character input disabled status

Syntax	GCSTextboxGetRefuseInputFullLetterStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets setting as to whether two-byte character input will be rejected for text boxes or not.
Example	Gets the GTextBox00000 two-byte letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputFullLetterStatus(-1, "GTextBox00000") ;

GCSTextboxSetCheckMaxStatus	Set maximum value check status

Syntax	GCSTextboxSetCheckMaxStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : setting for maximum value check status
Return value	None
Details	Sets the status of the maximum value check during input/output for the text box. One of the following values is set for fStatus: 0: maximum value check disabled 1: maximum value check enabled
Example	Sets the GTextBox00000 maximum value check status in the self screen to 1. GCSTextboxSetCheckMaxStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetCheckMaxStatus	Get maximum value check status

Syntax	GCSTextboxGetCheckMaxStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: check disabled 1: check enabled
Details	Gets the status of the maximum value check during input/output from the text box.
Example	Gets the GTextBox00000 maximum value check status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetCheckMaxStatus(-1, "GTextBox00000") ;

GCSTextboxSetCheckMinStatus	Set minimum value check status

Syntax	GCSTextboxSetCheckMinStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : setting for minimum value check status
Return value	None
Details	Sets the status of the minimum value check during input/output for the text box. One of the following values is set for fStatus: 0: minimum value check disabled 1: minimum value check enabled
Example	Sets the GTextBox00000 minimum value check status in the self screen to 1. GCSTextboxSetCheckMinStatus(-1, "GTextBox00000", 1) ;

GCSTextboxGetCheckMinStatus	Get minimum value check status

Syntax	GCSTextboxGetCheckMinStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: check disabled 1: check enabled
Details	Gets the status of the minimum value check during input/output from the text box.
Example	Gets the GTextBox00000 minimum value check status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetCheckMinStatus(-1, "GTextBox00000") ;

GCSTextboxSetMaxGValue	Set maximum value

Syntax	GCSTextboxSetMaxGValue(nWindowNo, strName, gvValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gvValue : maximum value
Return value	None
Details	Sets the value of the maximum value check during input/output for the text box.
Example	Sets the value of GTextBox00000 maximum value check in the self screen to 100. GCSTextboxSetMaxGValue(-1, "GTextBox00000", 100) ;

GCSTextboxGetMaxGValue	Get maximum value

Syntax	GCSTextboxGetMaxGValue(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Maximum value (GValue)
Details	Gets the value of the maximum value check during input/output from the text box.
Example	Gets the value of GTextBox00000 maximum value check in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetMaxGValue(-1, "GTextBox00000") ;

GCSTextboxSetMinGValue	Set minimum value
Syntax	GCSTextboxSetMinGValue(nWindowNo, strName, gvValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gvValue : minimum value
Return value	None
Details	Sets the value of the minimum value check during input/output for the text box.
Example	Sets the value of GTextBox00000 minimum value check in the self screen to 1. GCSTextboxSetMinGValue(-1, "GTextBox00000", 1) ;

GCSTextboxGetMinGValue	Get minimum value
Syntax	GCSTextboxGetMinGValue(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Minimum value (GValue)
Details	Gets the value of the minimum value check during input/output from the text box.
Example	Gets the value of GTextBox00000 minimum value check in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetMinGValue(-1, "GTextBox00000") ;

GCSTextboxSetString	Set display character string
Syntax	GCSTextboxSetString(nWindowNo, strName, pString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pString : display character string
Return value	0: setting failed 1: setting succeeded
Details	Sets the display character string.
Example	Sets the character string "ABCD" in GtextBox00000 in the self screen. GCSTextboxSetString(-1, "GTextBox00000", "ABCD") ;

GCSTextboxGetString	Get display character string
Syntax	GCSTextboxGetString(nWindowNo, strName, pString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) STRING pString : display character string
Return value	0: acquisition failed 1: acquisition succeeded
Details	Stores the display character string in pString.
Example	Gets the GTextBox00000 display character string in the self screen in Stat. STRING strStat; GCSTextboxGetString(-1, "GTextBox00000", strStat) ;

GCSTextboxSetGValue	Set display value

Syntax

GCSTextboxSetGValue(nWindowNo, strName, gvValue);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gvValue : global memory for setting value

Return value

0: normal valu
1: illegal value
2: less than minimum value
3: greater than maximum value
4: over maximum number of characters
5: illegal type

Details

Sets the value displayed in a text box.
Numbers are converted to character strings according to the display format.
At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value.
After the display value is set, the control area is registered as a redraw area.
(Note 1) If the number of characters after the conversion is over the value of the "maximum number of characters" of the properties, the "over maximum number of characters" will be returned. Even when the "over maximum number of characters" occurs, a character string which is limited to "maximum number of characters" is displayed.
(Note 2) Use the function "GCSCTextboxSetString", if float or double is set to the "type" of the properties.
(Note 3) If "Display Format" does not match the "Type" property, the value cannot be set.

Example

Sets the LONG integer value 100 as a value to be displayed in the GTextBox00000 in the self screen.

```
GMEM mem;
GCSTextboxSetTextType(-1, "GTextBox00000", 3) ;
mem = GMEMCreate("TESTMEM", 4) ;
GMEMSetLong(mem, 0, 100) ;
GCSTextboxSetGValue(-1, "GTextBox00000", mem) ;
GMEMDelete(mem);
```

GCSTextboxGetGValue	Get display value

Syntax

GCSTextboxGetGValue(nWindowNo, strName, gvValue);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(o) GMEM gvValue : global memory to store numerical value

Return value

0: normal value
1: illegal value
2: less than minimum value
3: greater than maximum value
5: illegal type

Details

Gets the value displayed in a text box.
Character strings are converted to numbers according to the display format.
At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value.
(Note 1) Use the function "GCSCTextboxSetString", if float or double is set to the "type" of the properties.
(Note 2) If "Display Format" does not match the "Type" property, the value cannot be set.

Example

Gets the numerical value being displayed in GtextBox00000 in the self screen as nVal : LONG integer value.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSTextboxGetGValue(-1, "GTextBox00000", mem) ;
nVal = GMEMGetLong(mem, 4)
GMEMDelete(mem);
```

GCSTextboxSetBackGroundPattern	Set background fill status

Syntax	GCSTextboxSetBackGroundPattern(nWindowNo, strName, nBackGroundPattern);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. One of the following values is set for nBackGroundPattern: -1: no fill -2: with background fill
Example	Sets the GTextBox00000 background fill status in the self screen to -2. GCSTextboxSetBackGroundPattern(-1, "GTextBox00000", -2) ;

GCSTextboxGetBackGroundPattern	Get background fill status

Syntax	GCSTextboxGetBackGroundPattern(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Background fill status -1: no fill -2: with background fill
Details	Gets the background color fill status.
Example	Gets the GTextBox00000 background fill status in the self screen in Stat. LONG Stat; Stat = GCSTextboxGetBackGroundPattern(-1, "GTextBox00000") ;

5.14 NCPLCButton

GCSNCPLCButtonSetAction	Set PLC button operations

Syntax	GCSNCPLCButtonSetAction(nWindowNo, strName, ucAction);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucAction : button operation
Return value	None
Details	Sets the PLC button operation. One of the following values is set for ucAction: 0: no operation 1: momentary operation 2: alternative operation When a value outside of the range is set, there is no forced change of the setting.
Example	Sets the GNCPLCButton00000 operations in the self screen to the momentary mode. GCSNCPLCButtonSetAction(-1, "GNCPLCButton00000", 1);

GCSNCPLCButtonGetAction	Get PLC button operations

Syntax	GCSNCPLCButtonGetAction(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: no operation 1: momentary operation 2: alternative operation Besides the above: illegal operation setting
Details	Gets the PLC button operation setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GNCPLCButton00000 operations in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetAction (-1, "GNCPLCButton00000");

GCSNCPLCButtonSetDisplay	Set PLC button display

Syntax	GCSNCPLCButtonSetDisplay(nWindowNo, strName, ucDisplay);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucDisplay : button display
Return value	None
Details	Sets the PLC button display. One of the following values is set for ucDisplay: 0: rectangle display 1: image display 3: round shape display When a value outside of the range is set, there is no forced change of the setting. After the PLC button display is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 in the self screen to be displayed in a rectangle. GCSNCPLCButtonSetDisplay(-1, "GNCPLCButton00000", 0);

GCSNCPLCButtonGetDisplay	Get PLC button display

Syntax	GCSNCPLCButtonGetDisplay(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: rectangle display 1: image display 3: round shape display Besides the above: illegal display setting
Details	Gets the PLC button display setting. When a setting outside the range is made, that setting value is returned.
Example	Gets the GNCPLCButton00000 display setting in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetDisplay(-1, "GNCPLCButton00000");

GCSNCPLCButtonSetBorderID	Set 3D border resource ID

Syntax	GCSNCPLCButtonSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 3D border resource ID in the self screen to 1. GCSNCPLCButtonSetBorderID(-1, "GNCPLCButton00000", 1);

GCSNCPLCButtonGetBorderID	Get 3D border resource ID

Syntax	GCSNCPLCButtonGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GNCPLCButton00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetBorderID(-1, "GNCPLCButton00000");

GCSNCPLCButtonSetOnDesign	Set ON status design

Syntax

GCSNCPLCButtonSetOnDesign(nWindowNo, strName, gmDesign);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmDesign : global memory for design

Return value

None

Details

Sets the design for ON status.
After the design for ON status is set, the control area is registered as a redraw area.

Example

Sets the GNCPLCButton00000 ON status design in the self screen as follows:
Fill pattern : 4
Foreground color : 0xffffff
Background color : 0x000000
Image resource ID : 2

```
mem = GMEMCreate("TESTMEM", 16);
GMEMSetShort(mem, 0, 4);
GMEMSetLong(mem, 4, HFFFFFF);
GMEMSetLong(mem, 8, H000000);
GMEMSetLong(mem, 12, 2);
GCSNCPLCButtonSetOnDesign(-1, "GNCPLCButton00000", mem);
GMEMDelete(mem);
```

GCSNCPLCButtonGetOnDesign	Get ON status design

Syntax	GCSNCPLCButtonGetOnDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for ON status.

Example	Gets the GNCPLCButton00000 ON status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetOffDesign	Set OFF status design

Syntax

GCSNCPLCButtonSetOffDesign(nWindowNo, strName, gmDesign);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmDesign : global memory for design

Return value

None

Details

Sets the design for OFF status.
After the design for OFF status is set, the control area is registered as a redraw area.

Example

Sets the GNCPLCButton00000 OFF status design in the self screen as follows:

```
 Fill pattern : 4
 Foreground color : 0xffffff
 Background color : 0x000000
 Image resource ID : 2
mem = GMEMCreate("TESTMEM", 16);
GMEMSetShort(mem, 0, 4);
GMEMSetLong(mem, 4, HFFFFFF);
GMEMSetLong(mem, 8, H000000);
GMEMSetShort(mem, 12, 2);
GCSNCPLCButtonSetOffDesign(-1, "GNCPLCButton00000",
mem);
GMEMDelete(mem);
```

GCSNCPLCButtonGetOffDesign	Get OFF status design

Syntax	GCSNCPLCButtonGetOffDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for OFF status.

Example	Gets the GNCPLCButton00000 OFF status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOffDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetOnOnDesign	Set ONON status design

Syntax	GCSNCPLCButtonSetOnOnDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for ONON status.
Example	Sets the GNCPLCButton00000 ONON status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSNCPLCButtonSetOnOnDesign(-1, "GNCPLCButton00000", mem); GMEMDelete(mem);

GCSNCPLCButtonGetOnOnDesign	Get ONON status design

Syntax	GCSNCPLCButtonGetOnOnDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for ONON status.

Example	Gets the GNCPLCButton00000 ONON status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnOnDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetOnOffDesign	Set ONOFF status design

Syntax	GCSNCPLCButtonSetOnOffDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for ONOFF status.
Example	Sets the GNCPLCButton00000 ONOFF status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSNCPLCButtonSetOnOffDesign(-1, "GNCPLCButton00000", mem); GMEMDelete(mem);

GCSNCPLCButtonGetOnOffDesign	Get ONOFF status design

Syntax	GCSNCPLCButtonGetOnOffDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for ONOFF status.

Example	Gets the GNCPLCButton00000 ONOFF status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnOffDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetFocusDesign	Set FOCUS status design

Syntax	GCSNCPLCButtonSetFocusDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for FOCUS status. After the design for FOCUS status is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 FOCUS status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSNCPLCButtonSetFocusDesign(-1, "GNCPLCButton00000", mem); GMEMDelete(mem);

GCSNCPLCButtonGetFocusDesign	Get FOCUS status design

Syntax	GCSNCPLCButtonGetFocusDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for FOCUS status.

Example	Gets the GNCPLCButton00000 FOCUS status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetFocusDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetDisableDesign	Set Disable status design

Syntax	GCSNCPLCButtonSetDisableDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmDesign : global memory for design
Return value	None
Details	Sets the design for Disable status. After the design for Disable status is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 Disable status design in the self screen as follows: Fill pattern : 4 Foreground color : 0xffffff Background color : 0x000000 Image resource ID : 2 mem = GMEMCreate("TESTMEM", 16); GMEMSetShort(mem, 0, 4); GMEMSetLong(mem, 4, HFFFFFF); GMEMSetLong(mem, 8, H000000); GMEMSetShort(mem, 12, 2); GCSNCPLCButtonSetDisableDesign(-1, "GNCPLCButton00000", mem); GMEMDelete(mem);

GCSNCPLCButtonGetDisableDesign	Get Disable status design

Syntax	GCSNCPLCButtonGetDisableDesign(nWindowNo, strName, gmDesign);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmDesign : global memory for design
Return value	Store the acquired design in gmDesign.
Details	Gets the design for Disable status.

Example	Gets the GNCPLCButton00000 Disable status design in the self screen as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetDisableDesign(-1, "GNCPLCButton00000", mem);
	nPat = GMEMGetShort(mem, 0);
	nFCol = GMEMGetLong(mem, 4);
	nBCol = GMEMGetLong(mem, 8);
	nImg = GMEMGetShort(mem, 12);
	GMEMDelete(mem);

GCSNCPLCButtonSetFontID	Set font resource ID

Syntax	GCSNCPLCButtonSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : Font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 font resource ID in the self screen to 1. GCSNCPLCButtonSetFontID(-1, "GNCPLCButton00000", 1);

GCSNCPLCButtonGetFontID	Get font resource ID

Syntax	GCSNCPLCButtonGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GNCPLCButton00000 font resource ID in the self screen in Stat. LONG Stat; Stat=GCSNCPLCButtonGetFontID(-1,"GNCPLCButton0000");

GCSNCPLCButtonSetOnStringID	Set ON status caption character string resource ID

Syntax	GCSNCPLCButtonSetOnStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : caption character string resource ID
Return value	None
Details	Sets the ON status caption character string resource ID. After the ON status caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 ON status caption character string resource ID in the self screen to 1. GCSNCPLCButtonSetOnStringID(-1, " GNCPLCButton00000", 1);

GCSNCPLCButtonGetOnStringID	Get ON status caption character string resource ID

Syntax	GCSNCPLCButtonGetOnStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the ON status caption character string resource ID.
Example	Gets the GNCPLCButton00000 ON status caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnStringID(-1, " GNCPLCButton00000");

GCSNCPLCButtonSetOffStringID	Set OFF status caption character string resource ID

Syntax	GCSNCPLCButtonSetOffStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : Font resource ID
Return value	None
Details	Sets the OFF status caption character string resource ID. After the OFF status caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 OFF status caption character string resource ID in the self screen to 1. GCSNCPLCButtonSetOffStringID(-1, " GNCPLCButton00000", 1);

GCSNCPLCButtonGetOffStringID	Get OFF status caption character string resource ID

Syntax	GCSNCPLCButtonGetOffStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the OFF status caption character string resource ID.
Example	Gets the GNCPLCButton00000 OFF status caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOffStringID(-1, " GNCPLCButton00000");

GCSNCPLCButtonSetOnOnStringID	Set ONON status caption character string resource ID

Syntax	GCSNCPLCButtonSetOnOnStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : Font resource ID
Return value	None
Details	Sets the ONON status caption character string resource ID. After the ONON status caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 ONON status caption character string resource ID in the self screen to 1. GCSNCPLCButtonSetOnOnStringID(-1, " GNCPLCButton00000", 1);

GCSNCPLCButtonGetOnOnStringID	Get ONON status caption character string resource ID

Syntax	GCSNCPLCButtonGetOnOnStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the ONON status caption character string resource ID.
Example	Gets the GNCPLCButton00000 ONON status caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnOnStringID(-1, " GNCPLCButton00000");

GCSNCPLCButtonSetOnOffStringID	Set ONOFF status caption character string resource ID

Syntax	GCSNCPLCButtonSetOnOffStringID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : Font resource ID
Return value	None
Details	Sets the ONOFF status caption character string resource ID. After the ONOFF status caption character string resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 ONOFF status caption character string resource ID in the self screen to 1. GCSNCPLCButtonSetOnOffStringID(-1, " GNCPLCButton00000", 1);

GCSNCPLCButtonGetOnOffStringID	Get ONOFF status caption character string resource ID

Syntax	GCSNCPLCButtonGetOnOffStringID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Caption character string resource ID
Details	Gets the ONOFF status caption character string resource ID.
Example	Gets the GNCPLCButton00000 ONOFF status caption character string resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnOffStringID(-1, " GNCPLCButton00000");

GCSNCPLCButtonSetCaption	Set caption information

Syntax

GCSNCPLCButtonSetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmCaption : global memory for caption information

Return value

None

Details

Sets the caption information.
After the caption information is set, the control area is registered as a redraw area.

Example

Sets the GNCPLCButton00000 caption information in the self screen as follows:
Color : White (0xffffff)
Horizontal position : 0
Vertical position : 1
Left margin : 10
Right margin : 0
Top margin : 0
Bottom margin : 0

```
mem = GMEMCreate("TESTMEM", 14);
GMEMSetLong(mem, 0, HFFFFFF);
GMEMSetChar(mem, 4, 0);
GMEMSetChar(mem, 5, 1);
GMEMSetShort(mem, 6, 10);
GMEMSetShort(mem, 8, 0);
GMEMSetShort(mem, 10, 0);
GMEMSetShort(mem, 12, 0);
GCSNCPLCButtonSetCaption(-1, "GNCPLCButton00000", mem);
GMEMDelete(mem);
```

GCSNCPLCButtonGetCaption	Get caption information

Syntax

GCSNCPLCButtonGetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(o) GMEM gmCaption : global memory for caption information

Return value

Store the acquired caption information in gmCaption.

Details

Gets the caption information.

Example

Gets the GNCPLCButton00000 caption information in the self screen as follows:

```
 nCol : Color
 nHPos : Horizontal position
 nVPos : Vertical position
 nLMgn : Left margin
 nRMgn : Right margin
 nTMgn : Top margin
 nBMgn : Bottom margin
mem = GMEMCreate("TESTMEM", 14);
GCSNCPLCButtonGetCaption(-1, "GNCPLCButton00000", mem);
nCol = GMEMGetLong(mem, 0);
nHPos = GMEMGetChar(mem, 4);
nHPos = GMEMGetChar(mem, 5);
nLMgn = GMEMGetShort(mem, 6);
nRMgn = GMEMGetShort(mem, 8);
nTMgn = GMEMGetShort(mem, 10);
GMEMDelete(mem);

nBMgn = GMEMGetShort(mem, 12);
```

GCSNCPLCButtonSetFocusEffect	Set effect during focus

Syntax	GCSNCPLCButtonSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect during focus is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 effect during focus in the self screen to 1. GCSNCPLCButtonSetFocusEffect(-1, "GNCPLCButton00000", 1);

GCSNCPLCButtonGetFocusEffect	Get effect during focus

Syntax	GCSNCPLCButtonGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GNCPLCButton00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetFocusEffect(-1, "GNCPLCButton00000");

GCSNCPLCButtonSetStatus	Set object status

Syntax	GCSNCPLCButtonSetStatus(nWindowNo, strName, ucStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucStatus : Object status
Return value	None
Details	Sets the GCNCPLCButton object ON/OFF status. One of the following values is set for ucStatus: 0: OFF status 1: ON status After the GCNCPLCButton object ON/OFF is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCButton00000 ON/OFF status in the self screen to 1. GCSNCPLCButtonSetStatus(-1, "GNCPLCButton00000", 1);

GCSNCPLCButtonGetStatus	Get object status

Syntax	GCSNCPLCButtonGetStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Object status 0: OFF status 1: ON status
Details	Gets the GCNCPLCButton object status.
Example	Gets the GNCPLCButton00000 ON/OFF status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCButtonGetStatus(-1, "GNCPLCButton00000");

GCSNCPLCButtonSetDeviceType	PLC device operation

Syntax

GCSNCPLCButtonSetDeviceType(nWindowNo, strName, ucIndex ,ucType);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG ucIndex : PLC device index No.
0 : PLC device 1
1 : PLC device 2
(i) LONG ucType : PLC device operation
0 : No setting
1 : Reading
2 : Writing

Return value

Setting result
0 : Abnormal (index No. illegal)
1: Normal

Details

Sets the PLC device operation to allocate to the control.

Example

Sets PLC device 1 operation to the reading in GNCPLCButton00000 in the self screen.

```
LONG Stat;
Stat = GCSNCPLCButtonSetDeviceType(-1,
"GNCPLCButton00000", 0, 1);
```

GCSNCPLCButtonGetDeviceType	Get PLC device operations

Syntax

GCSNCPLCButtonGetDeviceType(nWindowNo, strName, ucIndex, gmType);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG ucIndex : PLC device index No.
0 : PLC device 1
1 : PLC device 2
(o) GMEM gmType : global memory to store PLC device operations
0 : No setting
1 : Reading
2 : Writing

Return value

Getting result
0 : Abnormal (index No. illegal)
1: Normal

Details

Gets the PLC device operation allocated to the control.

Example

Gets PLC device 1 operation in GNCPLCButton00000 in the self screen as nVal : LONG integer value.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSNCPLCButtonGetDeviceType(-1, "GNCPLCButton00000", 0, mem) ;
nVal = GMEMGetLong(mem, 0) ;
GMEMDelete(mem);
```

GCSNCPLCButtonSetDevice	Set PLC device

Syntax

GCSNCPLCButtonSetDevice(nWindowNo, strName, ucIndex, pszDevice);

Argument

(i) LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i) STRING strName	: control name
(i) LONG ucIndex	: PLC device index No. 0 : PLC device 1 1 : PLC device 2
(i) STRING pszDevice	: PLC device character string

Return value

Setting result
0: Abnormal
1: Normal

Details

Sets the PLC device address to allocate to the control.

Example

Sets "X0" to PLC device of PLC device 1 in GNCPLCButton00000 in the self screen.

```
LONG Stat;
Stat = GCSNCPLCButtonSetDevice(-1, "GNCPLCButton00000", 0, "X0");
```

GCSNCPLCButtonGetDevice	Get PLC device

Syntax	GCSNCPLCButtonGetDevice(nWindowNo, strName, ucIndex, pszDevice);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucIndex : PLC device index No. 0 : PLC device 1 1 : PLC device 2 (o) STRING pszDevice : PLC device
Return value	Getting result 0: Abnormal 1: Normal
Details	Gets the PLC device address allocated to the control.
Example	Sets PLC device of PLC device 1 to strStat in GNCPLCButton00000 in the self screen. STRING strStat; GCSNCPLCButtonGetDevice(-1, "GNCPLCButton00000", 0, strStat);

GCSNCPLCButtonSetDeviceVal	Set PLC device value

Syntax

GCSNCPLCButtonSetDeviceVal(nWindowNo, strName, ucIndex, ucVal);

Argument

(i) LONG nWindowNo	: screen No. (Specify -1 for self screen.)	
(i) STRING strName	: control name	
(i) LONG ucIndex	: PLC device index No.	
	0 : PLC device 1	
	1 : PLC device 2	
(i) LONG ucVal	: PLC device value	

Return value

Setting result
0x5002 : Index illegal
0x5003 : Writing failure
0x0 : Normal

Details

Sets the PLC device value to allocate to the control.

Example

Sets 1 to PLC device value of PLC device 1 in GNCPLCButton00000 in the self screen.

```
LONG Stat;
Stat = GCSNCPLCButtonSetDeviceVal(-1,
"GNCPLCButton00000", 0, 1);
```

GCSNCPLCButtonGetDeviceVal	Get PLC device value

Syntax

GCSNCPLCButtonGetDeviceVal(nWindowNo, strName, ucIndex, gmVal);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG ucIndex : PLC device index No.
0 : PLC device 1
1 : PLC device 2
(o) GMEM gmVal : global memory to store PLC device value

Return value

Getting result
0x5002 : Index illegal
0x5003 : Reading failure
0x0 : Normal

Details

Gets the PLC device value allocated to the control.

Example

Gets PLC device value of PLC device 1 in GNCPLCButton00000 in the self screen as nVal : LONG integer value.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSNCPLCButtonGetDeviceVal(-1, " GNCPLCButton00000", 0, mem);
nVal = GMEMGetLong(mem, 0) ;
GMEMDelete(mem);
```

5.15 NCPLCTextbox

GCSNCPLCTextboxSetBorderID	Set 3D border resource ID

Syntax	GCSNCPLCTextboxSetBorderID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 3D border resource ID in the self screen to 1. GCSNCPLCTextboxSetBorderID(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetBorderID	Get 3D border resource ID

Syntax	GCSNCPLCTextboxGetBorderID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
Example	Gets the GNCPLCTextBox00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetBorderID(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetFontID	Set font resource ID

Syntax	GCSNCPLCTextboxSetFontID(nWindowNo, strName, usID);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG usID : Font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 font resource ID in the self screen to 1. GCSNCPLCTextboxSetFontID(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetFontID	Get font resource ID

Syntax	GCSNCPLCTextboxGetFontID(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GNCPLCTextBox00000 font resource ID in the self screen in Stat. Stat = GCSNCPLCTextboxGetFontID(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetCaption	Set caption information

Syntax

GCSNCPLCTextboxSetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmCaption : global memory for caption information

Return value

None

Details

Sets the caption information.
After the caption information is set, the control area is registered as a redraw area.

Example

Sets the GNCPLCTextBox00000 caption information in the self screen as follows:

```
 Color : White (0xffffff)
 Horizontal position : 0
 Vertical position : 1
 Left margin : 10
 Right margin : 0
 Top margin : 0
 Bottom margin : 0
mem = GMEMCreate("TESTMEM", 14) ;
GMEMSetLong(mem, 0, HFFFFFF) ;
GMEMSetChar(mem, 4, 0) ;
GMEMSetChar(mem, 5, 1) ;
GMEMSetShort(mem, 6, 10) ;
GMEMSetShort(mem, 8, 0) ;
GMEMSetShort(mem, 10, 0) ;
GMEMSetShort(mem, 12, 0) ;
GCSNCPLCTextboxSetCaption(-1, "GNCPLCTextBox00000",
mem) ;
GMEMDelete(mem);
```

GCSNCPLCTextboxGetCaption	Get caption information

Syntax

GCSNCPLCTextboxGetCaption(nWindowNo, strName, gmCaption);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(o) GMEM gmCaption : global memory for caption information

Return value

Store the acquired caption information in gmCaption.

Details

Gets the caption information setting.

Example

Gets the GNCPLCTextBox00000 caption information in the self screen as follows:

```
 nCol : Color
 nHPos : Horizontal position
 nVPos : Vertical position
 nLMgn : Left margin
 nRMgn : Right margin
 nTMgn : Top margin
 nBMgn : Bottom margin
mem = GMEMCreate("TESTMEM", 14) ;
GCSNCPLCTextboxGetCaption(-1, "GNCPLCTextBox00000",
mem) ;
nCol = GMEMGetLong(mem, 0) ;
nHPos = GMEMGetChar(mem, 4) ;
nHPos = GMEMGetChar(mem, 5) ;
nLMgn = GMEMGetShort(mem, 6) ;
nRMgn = GMEMGetShort(mem, 8) ;
nTMgn = GMEMGetShort(mem, 10) ;
nBMgn = GMEMGetShort(mem, 12) ;
GMEMDelete(mem);
```

GCSNCPLCTextboxSetFocusEffect	Set effect during focus

Syntax	GCSNCPLCTextboxSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 0: design change 1: no effect After the effect during focus is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 effect during focus in the self screen to 1. GCSNCPLCTextboxSetFocusEffect(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetFocusEffect	Get effect during focus

Syntax	GCSNCPLCTextboxGetFocusEffect(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: design change 1: no effect Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GNCPLCTextBox00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetFocusEffect(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetFocusColor	Set background color during focus

Syntax	GCSNCPLCTextboxSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : Color code for background color during focus
Return value	None
Details	Sets the background color during focus. After the background color during focus is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 background color during focus in the self screen to white (0xffffff). GCSNCPLCTextboxSetFocusColor(-1, "GNCPLCTextBox00000", HFFFFFF);

GCSNCPLCTextboxGetFocusColor	Get background color during focus

Syntax	GCSNCPLCTextboxGetFocusColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color during focus
Details	Gets the background color during focus.
Example	Gets the GNCPLCTextBox00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetFocusColor(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetNormalColor	Set normal background color

Syntax	GCSNCPLCTextboxSetNormalColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : Color code for normal background color
Return value	None
Details	Sets the normal background color. After the normal background color is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 normal background color in self screen to white (0xffffff). GCSNCPLCTextboxSetNormalColor(-1, "GNCPLCTextBox00000", HFFFFFF) ;

GCSNCPLCTextboxGetNormalColor	Get normal background color

Syntax	GCSNCPLCTextboxGetNormalColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for normal background color
Details	Gets the normal background color.
Example	Gets the GNCPLCTextBox00000 normal background color in self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetNormalColor(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetDisableColor	Set background color when disabled

Syntax	GCSNCPLCTextboxSetDisableColor(nWindowNo, strName, gcColor);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color when disabled is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 background color when disabled in the self screen to white (0xffffff). GCSNCPLCTextboxSetDisableColor(-1, "GNCPLCTextBox00000", HFFFFFF) ;

GCSNCPLCTextboxGetDisableColor	Get background color when disabled

Syntax	GCSNCPLCTextboxGetDisableColor(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
Example	Gets the GNCPLCTextBox00000 background color when disabled in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetDisableColor(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetTextType	Set character string type

Syntax

GCSNCPLCTextboxSetTextType(nWindowNo, strName, ucType);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG ucType : character string type

Return value

None

Details

Sets the character string type.
One of the following values is set for ucType:

GPLCTEXT_TYPE_DEC(1) : Signed decimal integer
GPLCTEXT_TYPE_UDEC(2) : Unsigned decimal integer
GPLCTEXT_TYPE_HEX(3) : Hexadecimal integer
GPLCTEXT_TYPE_FLOAT(4) : Real number (float)
GPLCTEXT_TYPE_DOUBLE(5) : Real number (double)
GPLCTEXT_TYPE_STRING(6) : Character sequence

After the character string type is set, the control area is registered as a redraw area.

Example

Sets the GNCPLCTextBox00000 display character string type in the self screen to 1.

```
GCSNCPLCTextboxSetTextType(-1, "GNCPLCTextBox00000",
1) ;
```

GCSNCPLCTextboxGetTextType	Get character string type

Syntax

GCSNCPLCTextboxGetTextType(nWindowNo, strName);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name

Return value

GPLCTEXT_TYPE_DEC(1) : Signed decimal integer
GPLCTEXT_TYPE_UDEC(2) : Unsigned decimal integer
GPLCTEXT_TYPE_HEX(3) : Hexadecimal integer
GPLCTEXT_TYPE_FLOAT(4) : Real number (float)
GPLCTEXT_TYPE_DOUBLE(5) : Real number (double)
GPLCTEXT_TYPE_STRING(6) : Character sequence

Details

Gets the display character string type.

Example

Gets the GNCPLCTextBox00000 display character string type in the self screen in Stat.

```
LONG Stat;
Stat = GCSNCPLCTextboxGetTextType(-1,
"GNCPLCTextBox00000") ;
```

GCSNCPLCTextboxSetTextSize	Set size

Syntax	GCSNCPLCTextboxSetTextSize(nWindowNo, strName, ucSize);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucSize : PLC device size 2 : 2 byte 4 : 4 byte
Return value	None
Details	Sets the PLC device size to allocate to the control. After the PLC device size is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 PLC device size in the self screen to 2 byte. GCSNCPLCTextboxSetTextSize(-1, "GNCPLCTextBox00000", 2);

GCSNCPLCTextboxGetTextSize	Get size

Syntax	GCSNCPLCTextboxGetTextSize(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	PLC device size 2 : 2 byte 4 : 4 byte
Details	Gets the PLC device size allocated to the control.
Example	Gets the GNCPLCTextBox00000 PLC device size in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetTextSize(-1, "GNCPLCTextBox00000");

GCSNCPLCTextboxSetPasswordStatus	Set password display status

Syntax	GCSNCPLCTextboxSetPasswordStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : password display status
Return value	None
Details	Sets whether the password will be displayed for text boxes or not. One of the following values is set for fStatus: 0: normal display status 1: password display status After the password display status is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 password display status in the self screen to 1. GCSNCPLCTextboxSetPasswordStatus(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetPasswordStatus	Get password display status

Syntax	GCSNCPLCTextboxGetPasswordStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: password display status
Details	Gets whether the test box display status is normal display status or the password display status.
Example	Gets the GNCPLCTextBox00000 password display status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetPasswordStatus(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetCommaStatus	Set comma display status

Syntax	GCSNCPLCTextboxSetCommaStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : comma display status
Return value	None
Details	Sets whether commas will be displayed or not for text boxes. One of the following values is set for fStatus: 0: normal display status 1: comma display status After the comma display status is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 comma display status in the self screen to 1. GCSNCPLCTextboxSetCommaStatus(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetCommaStatus	Get comma display status

Syntax	GCSNCPLCTextboxGetCommaStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: comma display status
Details	Gets the setting as to whether the test box display status is normal display status or the comma display status.
Example	Gets the GNCPLCTextBox00000 comma display status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCommaStatus(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetCheckMaxStatus	Set maximum value check status

Syntax	GCSNCPLCTextboxSetCheckMaxStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : setting for maximum value check status
Return value	None
Details	Sets the status of the maximum value check during input/output for the text box. One of the following values is set for fStatus: 0: Maximum value check disabled 1: Maximum value check enabled
Example	Sets the GNCPLCTextBox00000 maximum value check status in the self screen to 1. GCSNCPLCTextboxSetCheckMaxStatus(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetCheckMaxStatus	Get maximum value check status

Syntax	GCSNCPLCTextboxGetCheckMaxStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: check disabled 1: check enabled
Details	Gets the status of the maximum value check during input/output from the text box.
Example	Gets the GNCPLCTextBox00000 maximum value check status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCheckMaxStatus(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetCheckMinStatus	Set minimum value check status

Syntax	GCSNCPLCTextboxSetCheckMinStatus(nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : setting for minimum value check status
Return value	None
Details	Sets the status of the minimum value check during input/output for the text box. One of the following values is set for fStatus: 0: Minimum value check disabled 1: Minimum value check enabled
Example	Sets the GNCPLCTextBox00000 minimum value check status in the self screen to 1. GCSNCPLCTextboxSetCheckMinStatus(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetCheckMinStatus	Get minimum value check status

Syntax	GCSNCPLCTextboxGetCheckMinStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: check disabled 1: check enabled
Details	Gets the status of the minimum value check during input/output from the text box.
Example	Gets the GNCPLCTextBox00000 minimum value check status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCheckMinStatus(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetMaxGValue	Set maximum value

Syntax	GCSNCPLCTextboxSetMaxGValue(nWindowNo, strName, gmValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmValue : maximum value
Return value	None
Details	Sets the value of the maximum value check during input/output for the text box.
Example	Sets the value of GNCPLCTextBox00000 maximum value check in the self screen to 100. GCSNCPLCTextboxSetMaxGValue(-1, "GNCPLCTextBox00000", 100) ;

GCSNCPLCTextboxGetMaxGValue	Get maximum value

Syntax	GCSNCPLCTextboxGetMaxGValue(nWindowNo, strName, gmValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmValue : maximum value
Return value	Store the maximum value in gmValue.
Details	Gets the value of the maximum value check during input/output from the text box.

Example	Gets the value of GNCPLCTextBox00000 maximum value check in the self screen in nMax.
	LONG nMax;
	mem = GMEMCreate("TESETMEM", 4);
	GCSNCPLCTextboxGetMaxGValue(-1, "GNCPLCTextBox00000", mem) ;
	nMax = GMEMGetLong(mem,0);
	GMEMDelete(mem);

GCSNCPLCTextboxSetMinGValue	Set minimum value

Syntax	GCSNCPLCTextboxSetMinGValue(nWindowNo, strName, gmValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmValue : minimum value
Return value	None
Details	Sets the value of the minimum value check during input/output for the text box.
Example	Sets the value of GNCPLCTextBox00000 minimum value check in the self screen to 1. GCSNCPLCTextboxSetMinGValue(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetMinGValue	Get minimum value

Syntax	GCSNCPLCTextboxGetMinGValue(nWindowNo, strName, gmValue);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmValue : minimum value
Return value	Store the minimum value in gmValue.
Details	Gets the value of the minimum value check during input/output from the text box.

Example	Gets the value of GNCPLCTextBox00000 minimum value check in the self screen in nMix.
	LONG nMix;
	mem = GMEMCreate("TESETMEM", 4);
	GCSNCPLCTextboxGetMinGValue(-1, "GNCPLCTextBox00000", mem) ;
	nMix = GMEMGetLong(mem,0);
	GMEMDelete(mem);

GCSNCPLCTextboxSetString	Set display character string

Syntax

GCSNCPLCTextboxSetString(nWindowNo, strName, pString);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) STRING pString : display character string

Return value

0: setting succeeded
0x6002: illegal value
0x6003: less than minimum value
0x6004: greater than maximum value
0x6005: writing failure
0x6007: failed to reserve memory
0x6008: over than maximum number of characters
0x600A: interlocked

Details

Sets the display character string.
(Note 1) By executing the display character string setting, data will be written in the PLC.
(Note 2) Character strings can be set even when the focus is not placed on the PLC text box.
(Note 3) Controls are not specified during interlock and an error "0x600A (interlocked)" is returned in the return value.

Example

Sets the character string "ABCD" in GNCPLCTextBox00000 in the self screen.

```
GCSNCPLCTextboxSetString(-1, "GNCPLCTextBox00000",
"ABCD") ;
```

GCSNCPLCTextboxGetString	Get display character string

Syntax	GCSNCPLCTextboxGetString(nWindowNo, strName, pString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) STRING pString : display character string
Return value	0: acquisition succeeded Other than 0: acquisition failed
Details	Stores the display character string in pString. (Note 1) "Get display character string" will acquire data from the PLC. (Note 2) When the focus is placed on the text box, the data displayed in the PLC text box and the PLC data may differ. In that case, PLC data will be acquired.
Example	Gets the GNCPLCTextBox00000 display character string in the self screen in Stat. STRING strStat; GCSNCPLCTextboxGetString(-1, "GNCPLCTextBox00000", strStat) ;

GCSNCPLCTextboxSetGValue	Set PLC device value

Syntax

GCSNCPLCTextboxSetGValue(nWindowNo, strName, gmValue);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) GMEM gmValue : global memory for setting value

Return value

0: normal value
0x6003: less than minimum value
0x6004: greater than maximum value
0x6005: writing failure
0x6007: failed to reserve memory
0x6008: over than maximum number of characters
0x6009: illegal type
0x600A: interlocked

Details

Sets the PLC device value to allocate to the control.
After the PLC device value is set, the control area is registered as a redraw area.
(Note 1) The character string, which is converted from PLC device value according to the display format, is displayed in the PLC text box.
(Note 2) If the number of characters after the conversion is over the value of the "maximum number of characters" of the properties, the "over maximum number of characters" will be returned. Even when it is over the "maximum number of characters", it sets the value onto the PLC device, a character string after the conversion which is limited to the "maximum number of characters"is displayed.
(Note 3) Use the function "GCSNCPLCTextboxSetString", if a real number (float) or a real number (double) is set to the "type" of the properties.
(Note 4) If "Character sequence" is set to the "type" of the properties, "illegal type" will be returned.
(Note 5) Controls are not specified during interlock and an error "0x600A (interlocked)" is returned in the return value.

Example	Sets the LONG integer value 100 to the PLC device set to the GNCPLCTextBox00000 in the self screen. GMEM mem; GCSNCPLCTextboxSetTextType (-1, "GNCPLCTextBox00000", 3) ; mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSNCPLCTextboxSetGValue(-1, "GNCPLCTextBox00000", mem) ; GMEMDelete(mem);

GCSNCPLCTextboxGetGValue	Get PLC device value

Syntax

GCSNCPLCTextboxGetGValue(nWindowNo, strName, gmValue);

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(o) GMEM gmValue : global memory to store numerical value

Return value

0: normal value
0x6002: illegal value
0x6003: less than minimum value
0x6004: more than maximum value
0x6009: illegal type

Details

Gets the PLC device value allocated to the control.
(Note 1) The character string, which is converted from PLC device value according to the display format, is displayed in the PLC text box.
(Note 2) Use the function "GCSNCPLCTextboxSetString", if a real number (float) or a real number (double) is set to the "type" of the properties.
(Note 3) If "Character sequence" is set to the "type" of the properties, "illegal type" will be returned.

Example

Gets the PLC device value as nVal : LONG integer value from the PLC device set on GNCPLCTextBox00000 in the screen No.10.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSNCPLCTextboxGetGValue(10, "GNCPLCTextBox00000",
mem) ;
nVal = GMEMGetLong(mem, 0) ;
GMEMDelete(mem);
```

GCSNCPLCTextboxSetBackGroundPattern	Set background fill status

Syntax	GCSNCPLCTextboxSetBackGroundPattern(nWindowNo, strName, nBackGroundPattern);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. One of the following values is set for nBackGroundPattern: -1: no fill -2: with background fill
Example	Sets the GNCPLCTextBox00000 background fill status in the self screen to -2. GCSNCPLCTextboxSetBackGroundPattern(-1, "GNCPLCTextBox00000", -2) ;

GCSNCPLCTextboxGetBackGroundPattern	Get background fill status

Syntax	GCSNCPLCTextboxGetBackGroundPattern(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Background fill status -1: no fill -2: with background fill
Details	Gets the background color fill status.
Example	Gets the GNCPLCTextBox00000 background fill status in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetBackGroundPattern(-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetIntegerWidth	Set integer part width

Syntax	GCSNCPLCTextboxSetIntegerWidth(nWindowNo, strName, ucWidth);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucWidth : The number of integer part figures
Return value	None
Details	Sets the number of integer part figures.
Example	Sets the GNCPLCTextBox00000 the number of integer part figures in the self screen to 2. GCSNCPLCTextboxSetIntegerWidth(-1, "GNCPLCTextBox00000", 2);

GCSNCPLCTextboxGetIntegerWidth	Get integer part width

Syntax	GCSNCPLCTextboxGetIntegerWidth(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	The number of integer part figures
Details	Gets the number of integer part figures.
Example	Gets the GNCPLCTextBox00000 the number of integer part figures in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetIntegerWidth(-1, "GNCPLCTextBox00000");

GCSNCPLCTextboxSetDecimalWidth	Set decimal part width

Syntax	GCSNCPLCTextboxSetDecimalWidth(nWindowNo, strName, ucWidth);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ucWidth : The number of decimal part figures
Return value	None
Details	Sets the number of decimal part figures.
Example	Sets the GNCPLCTextBox00000 the number of decimal part figures in the self screen to 2. GCSNCPLCTextboxSetDecimalWidth(-1, "GNCPLCTextBox00000", 2);

GCSNCPLCTextboxGetDecimalWidth	Get decimal part width

Syntax	GCSNCPLCTextboxGetDecimalWidth(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	The number of decimal part figures
Details	Gets the number of decimal part figures.
Example	Gets the GNCPLCTextBox00000 the number of decimal part figures in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetDecimalWidth(-1, "GNCPLCTextBox00000");

GCSNCPLCTextboxSetScale	Set scale

Syntax	GCSNCPLCTextboxSetScale(nWindowNo, strName, gmScale);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) GMEM gmScale : global memory for scale
Return value	None
Details	Sets the scale.
Example	Sets the scale of GNCPLCTextBox00000 in the self screen as DOUBLE integer value : 2.5. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetDouble(mem, 0, 2.5) ; GCSNCPLCTextboxSetScale(-1, " GNCPLCTextBox00000", mem) ; GMEMDelete(mem);

GCSNCPLCTextboxGetScale	Get scale

Syntax	GCSNCPLCTextboxGetScale(nWindowNo, strName, gmScale);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) GMEM gmScale : global memory to store scale
Return value	None
Details	Gets the scale.
Example	Gets the scale of GNCPLCTextBox00000 in the self screen as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSNCPLCTextboxGetScale(-1, "GNCPLCTextBox00000", mem) ; nVal = GMEMGetLong(mem, 0) ; GMEMDelete(mem);

GCSNCPLCTextboxSetOffset	Set offset

Syntax	GCSNCPLCTextboxSetOffset(nWindowNo, strName, ulOffset);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG ulOffset : offset
Return value	None
Details	Sets the offset.
Example	Sets the GNCPLCTextBox00000 offset in the self screen to 100. GCSNCPLCTextboxSetOffset(-1, "GNCPLCTextBox00000", 100);

GCSNCPLCTextboxGetOffset	Get offset

Syntax	GCSNCPLCTextboxGetOffset(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Offset
Details	Gets the offset.
Example	Gets the GNCPLCTextBox00000 offset in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetOffset(-1, "GNCPLCTextBox00000");

GCSNCPLCTextboxSetDevice	Set PLC device

Syntax	GCSNCPLCTextboxSetDevice(nWindowNo, strName, pszDevice);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszDevice : PLC device character string
Return value	Setting result 0: Abnormal 1: Normal
Details	Sets the PLC device address to allocate to the control.
Example	Sets "X0" to GNCPLCTextBox00000 PLC device in the self screen. LONG Stat; Stat = GCSNCPLCTextboxSetDevice(-1, "GNCPLCTextBox00000", "X0");

GCSNCPLCTextboxGetDevice	Get PLC device

Syntax	GCSNCPLCTextboxGetDevice(nWindowNo, strName, pszDevice);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) STRING pszDevice : PLC device
Return value	Getting result 0: Abnormal 1: Normal
Details	Gets the PLC device address allocated to the control.
Example	Gets the GNCPLCTextBox00000 PLC device in the self screen in strStat. STRING strStat; GCSNCPLCTextboxGetDevice(-1, "GNCPLCTextBox00000", strStat);

GCSNCPLCTextboxSetZeroSuppressStatus	Set zero suppress display status

Syntax	GCSNCPLCTextboxSetZeroSuppressStatus (nWindowNo, strName, fStatus);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG fStatus : zero suppress setting 1 : zero suppress enabled 0 : zero suppress disabled
Return value	None
Details	Sets the zero suppress display status.
Example	Sets the GNCPLCTextBox00000 zero suppress in the self screen to ON. GCSNCPLCTextboxSetZeroSuppressStatus(-1, "GNCPLCTextBox00000", 1);

GCSNCPLCTextboxGetZeroSuppressStatus	Get zero suppress display status

Syntax	GCSNCPLCTextboxGetZeroSuppressStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Zero suppress display status 1 : zero suppress enabled 0 : zero suppress disabled
Details	Gets zero suppress display status.
Example	Gets the GNCPLCTextBox00000 zero suppress setting in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetZeroSuppressStatus(-1, "GNCPLCTextBox00000");

GCSNCPLCTextboxSetInterlockDevice	Set interlock device

Syntax	GCSNCPLCTextboxSetInterlockDevice(nWindowNo, strName, pszDevice) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pszDevice : interlock character string
Return value	Setting result 0: Abnormal 1: Normal
Details	Sets the interlock device address to allocate to the control.
Example	Sets "R0" to GNCPLCTextBox00000 interlock device in the self screen. LONG Stat; Stat = GCSNCPLCTextboxSetInterlockDevice (-1, "GNCPLCTextBox00000", “R0”) ;

GCSNCPLCTextboxGetInterlockDevice	Get interlock device

Syntax	GCSNCPLCTextboxGetInterlockDevice(nWindowNo, strName, pszDevice) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) STRING pszDevice : interlock character string
Return value	Getting result 0: Abnormal 1: Normal
Details	Gets the interlock device address to allocate to the control.
Example	Gets the GNCPLCTextBox00000 interlock device in the self screen in strStat. STRING strStat; GCSNCPLCTextboxGetInterlockDevice (-1, "GNCPLCTextBox00000", strStat) ;

GCSNCPLCTextboxSetInterlockDeviceAction	Set interlock device operation

Syntax	GCSNCPLCTextboxSetInterlockDeviceAction(nWindowNo, strName, nAction) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nAction : interlock device operation
Return value	Setting result 0: Abnormal 1: Normal
Details	Sets the interlock device operation. One of the following values is set for nAction. 0: None 1: ON 2: OFF When a setting outside the range is made, the setting value is not changed and "0 (Abnormal)" is returned in the return value.
Example	Sets "ON" to GNCPLCTextBox00000 operation in the self screen. GCSNCPLCTextboxSetInterlockDeviceAction(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetInterlockDeviceAction	Get interlock device operation

Syntax	GCSNCPLCTextboxGetInterlockDeviceAction(nWindowNo, strName) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Getting result 0: None 1: ON 2: OFF
Details	Gets the PLC textbox operation. When a value outside the range is got, the value is returned.
Example	Gets the GNCPLCTextBox00000 operation in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetInterlockDeviceAction (-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetInterlockDeviceBitPos	Set bit position of interlock device

Syntax	GCSNCPLCTextboxSetInterlockDeviceBitPos(nWindowNo, strName, nBitPos) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG nBitPos : bit position of interlock device
Return value	Setting result 0: Abnormal 1: Normal
Details	Sets the bit position of the interlock device. One of the following values is set for nBitPos. 0 to 15 When a setting outside the range is made, the setting value is not changed and "0 (Abnormal)" is returned in the return value.
Example	Sets "1" to the bit position of GNCPLCTextBox00000 interlock device in the self screen. GCSNCPLCTextboxSetInterlockDeviceBitPos(-1, "GNCPLCTextBox00000", 1) ;

GCSNCPLCTextboxGetInterlockDeviceBitPos	Get bit position of interlock device

Syntax	GCSNCPLCTextboxGetInterlockDeviceBitPos(nWindowNo, strName) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Getting result 0 to 15
Details	Gets the bit position of the PLC textbox interlock device. When a value outside the range is got, the setting value is returned.
Example	Gets the bit position of GNCPLCTextBox00000 interlock device in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetInterlockDeviceBitPos (-1, "GNCPLCTextBox00000") ;

GCSNCPLCTextboxSetInterlockDeviceProjectNo	Set project number of interlock device

Syntax

GCSNCPLCTextboxSetInterlockDeviceProjectNo
(nWindowNo, strName, nProjectNo) ;

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name
(i) LONG nProjectNo : project number of interlock device

Return value

Setting result
0: Abnormal
1: Normal

Details

Sets the project number of the interlock device.
One of the following values is set for nProjectNo.
0 to 6

Example

Sets "1" to the project number of GNCPLCTextBox00000 interlock device in the self screen.

```
GCSNCPLCTextboxSetInterlockDeviceProjectNo
                                    (-1, "GNCPLCTextBox00000", 1) ;
```

GCSNCPLCTextboxGetInterlockDeviceProjectNo	Get project number of interlock device

Syntax

GCSNCPLCTextboxGetInterlockDeviceProjectNo(nWindowNo, strName) ;

Argument

(i) LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i) STRING strName : control name

Return value

Getting result
0 to 6

Details

Gets the project number of the PLC textbox interlock device.
When a value outside the range is got, the setting value is returned.

Example

Gets the project number of GNCPLCTextBox00000 interlock device in the self screen in Stat.

```
LONG Stat;
Stat = GCSNCPLCTextboxGetInterlockDeviceProjectNo
                                    (-1, "GNCPLCTextBox00000") ;
```

GCSNCPLCTextboxSetInterlockColor	Set background color during interlock

Syntax	GCSNCPLCTextboxSetInterlockColor(nWindowNo, strName, gcColor) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) LONG gcColor : color code for background color during
Return value	None
Details	Sets the background color during interlock. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GNCPLCTextBox00000 background color during interlock in the self screen to white (0xffffff). GCSNCPLCTextboxSetInterlockColor(-1, "GNCPLCTextBox00000", HFFFFFF) ;

GCSNCPLCTextboxGetInterlockColor	Get background color during interlock

Syntax	GCSNCPLCTextboxGetInterlockColor (nWindowNo, strName) ;
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	Color code for background color during interlock
Details	Gets the background color during interlock.
Example	Gets the GNCPLCTextBox00000 background color during interlock in the self screen in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetInterlockColor(-1, "GNCPLCTextBox00000") ;

5.16 NCDataTextBox

GCSNCDataTextboxSetBorderID	Set 3D border resource ID

Syntax	GCSNCDataTextboxSetBorderID(nWindowNo, strName, usID);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 3D border resource ID in the self screen to 1. GCSNCDataTextboxSetBorderID(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetBorderID	Set 3D border resource ID

Syntax	GCSNCDataTextboxGetBorderID(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	3D border resource ID
Details	Sets the 3D border resource ID.
Example	Gets the GNCDataTextBox00000 3D border resource ID in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetBorderID(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetFontID	Set font resource ID

Syntax	GCSNCDataTextboxSetFontID(nWindowNo, strName, usID);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 font resource ID in the self screen to 1. GCSNCDataTextboxSetFontID(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetFontID	Get font resource ID

Syntax	GCSNCDataTextboxGetFontID(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GNCDataTextBox00000 font resource ID in the self screen in Stat. Stat = GCSNCDataTextboxGetFontID(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetCaption	Set caption information

Syntax

GCSNCDataTextboxSetCaption(nWindowNo, strName, gmCaption);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)GMEM gmCaption	: global memory for caption information

Return value

None

Details

Sets caption information.
After the caption information is set, the control area is registered as a redraw area.

Example

Sets the GNCDataTextBox00000 caption information in the self screen as follows:
Color : White (0xffffff)
Horizontal position : 0
Vertical position : 1
Left margin : 10
Right margin : 0
Top margin : 0
Bottom margin : 0

```
mem = GMEMCreate("TESTMEM", 14) ;
GMEMSetLong(mem, 0, HFFFFFF) ;
GMEMSetChar(mem, 4, 0) ;
GMEMSetChar(mem, 5, 1) ;
GMEMSetShort(mem, 6, 10) ;
GMEMSetShort(mem, 8, 0) ;
GMEMSetShort(mem, 10, 0) ;
GMEMSetShort(mem, 12, 0) ;
GCSNCDataTextboxSetCaption(-1, "GNCDataTextBox00000", mem) ;
GMEMDelete(mem);
```

GCSNCDataTextboxGetCaption	Get caption information

Syntax

GCSNCDataTextboxGetCaption(nWindowNo, strName, gmCaption);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(o)GMEM gmCaption	: global memory for caption information

Return value

Store the acquired caption information in gmCaption.

Details

Gets the caption information.

Example

Gets the GNCDataTextBox00000 caption information in the self screen as follows:

```
nCol : Color
nHPos : Horizontal position
nVPos : Vertical position
nLMgn : Left margin
nRMgn : Right margin
nTMgn : Top margin
nBMgn : Bottom margin
mem = GMEMCreate("TESTMEM", 14) ;
GCSNCDataTextboxGetCaption(-1, "GNCDataTextBox00000",
mem) ;
nCol = GMEMGetLong(mem, 0) ;
nHPos = GMEMGetChar(mem, 4) ;
nHPos = GMEMGetChar(mem, 5) ;
nLMgn = GMEMGetShort(mem, 6) ;
nRMgn = GMEMGetShort(mem, 8) ;
nTMgn = GMEMGetShort(mem, 10) ;
nBMgn = GMEMGetShort(mem, 12) ;
GMEMDelete(mem);
```

GCSNCDataTextboxSetFocusEffect	Set effect during focus

Syntax	GCSNCDataTextboxSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: 1: no effect 2: cursor display 3: select all After the effect is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 effect during focus in the self screen to 1. GCSNCDataTextboxSetFocusEffect(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetFocusEffect	Get effect during focus

Syntax	GCSNCDataTextboxGetFocusEffect(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	1: no effect 2: cursor display 3: select all Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GNCDataTextBox00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetFocusEffect(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetFocusColor	Set background color during focus

Syntax	GCSNCDataTextboxSetFocusColor(nWindowNo, strName, gcColor);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG gcColor : color code for background color during focus
Return value	None
Details	Sets the background color during focus. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 background color during focus in the self screen to white (0xffffff). GCSNCDataTextboxSetFocusColor(-1, "GNCDataTextBox00000", HFFFFFF);

GCSNCDataTextboxGetFocusColor	Get background color during focus

Syntax	GCSNCDataTextboxGetFocusColor(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Color code for background color during focus
Details	Gets the background color during focus.
Example	Gets the GNCDataTextBox00000 background color during focus in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetFocusColor(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetNormalColor	Set normal background color

Syntax	GCSNCDataTextboxSetNormalColor(nWindowNo, strName, gcColor);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the normal background color is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 normal background color in the self screen to white (0xffffff). GCSNCDataTextboxSetNormalColor(-1, "GNCDataTextBox00000", HFFFFFF) ;

GCSNCDataTextboxGetNormalColor	Get normal background color

Syntax	GCSNCDataTextboxGetNormalColor(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Color code for normal background color
Details	Gets the normal background color.
Example	Gets the GNCDataTextBox00000 normal background color in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetNormalColor(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetDisableColor	Set background color when disabled

Syntax	GCSNCDataTextboxSetDisableColor(nWindowNo, strName, gcColor);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 background color when disabled in the self screen to white (0xffffff). GCSNCDataTextboxSetDisableColor(-1, "GNCDataTextBox00000", HFFFFFF) ;

GCSNCDataTextboxGetDisableColor	Get background color when disabled

Syntax	GCSNCDataTextboxGetDisableColor(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
Example	Gets the GNCDataTextBox00000 background color when disabled in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDisableColor(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetTextType	Set character string type

Syntax	GCSNCDataTextboxSetTextType(nWindowNo, strName, ucType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucType : character string type
Return value	None
Details	Sets the character string type. One of the following values is set for ucType: 0: character string 1: Binary integer value 2: Signed decimal integer value 3: Unsigned decimal integer value 4: Hexadecimal integer value 5: Real number After the character string type is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 display character string type in the self screen to 1. GCSNCDataTextboxSetTextType(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetTextType	Get character string type

Syntax	GCSNCDataTextboxGetTextType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: character string 1: Binary integer value 2: Signed decimal integer value 3: Unsigned decimal integer value 4: Hexadecimal integer value 5: Real number Other than above: Illegal setting
Details	Gets the display character string type.
Example	Gets the GNCDataTextBox00000 display character string type in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetTextType(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetCommaStatus	Set comma display status

Syntax	GCSNCDataTextboxSetCommaStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : comma display status
Return value	None
Details	Sets whether commas will be displayed for text boxes or not. One of the following values is set for fStatus: 0: normal display status 1: comma display status After the comma display status is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000comma display status in the self screen to 1. GCSNCDataTextboxSetCommaStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetCommaStatus	Get comma display status

Syntax	GCSNCDataTextboxGetCommaStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: normal display status 1: comma display status
Details	Gets the setting as to whether the test box display status is normal display status or the comma display status.
Example	Gets the GNCDataTextBox00000comma display status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetCommaStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetRefuseInputNumberStatus	Set numerical input disabled status

Syntax	GCSNCDataTextboxSetRefuseInputNumberStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : number input status
Return value	None
Details	Sets whether numerical input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GNCDataTextBox00000 numerical input disabled status in the self screen to 1. GCSNCDataTextboxSetRefuseInputNumberStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetRefuseInputNumberStatus	Get numerical input disabled status

Syntax	GCSNCDataTextboxGetRefuseInputNumberStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets the setting as to whether numerical input will be rejected for text boxes or not.
Example	Gets the GNCDataTextBox00000 numerical input disabled status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputNumberStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetRefuseInputSmallLetterStatus	Set one-byte lower case character input disabled status

Syntax	GCSNCDataTextboxSetRefuseInputSmallLetterStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : one-byte lower case character input
Return value	None
Details	Sets whether one-byte lower case character input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled Example Sets the GTextBox00000
Example	Sets the GNCDataTextBox00000 one-byte small letter input disabled status in the self screen to 1. GCSNCDataTextboxSetRefuseInputSmallLetterStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetRefuseInputSmallLetterStatus	Get one-byte lower case character input disabled status

Syntax	GCSNCDataTextboxGetRefuseInputSmallLetterStatus (nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets the setting as to whether one-byte lower case character input will be rejected for text boxes or not.
Example	Gets the GNCDataTextBox00000 one-byte small letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputSmallLetterStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetRefuseInputCapitalLetterStatus	Set one-byte upper case character input disabled status

Syntax	GCSNCDataTextboxSetRefuseInputCapitalLetterStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : one-byte upper case character input
Return value	None
Details	Sets whether one-byte upper case character input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GNCDataTextBox00000 one-byte capital letter input disabled status in the self screen to 1. GCSNCDataTextboxSetRefuseInputCapitalLetterStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetRefuseInputCapitalLetterStatus	Get one-byte upper case character input disabled status

Syntax	GCSNCDataTextboxGetRefuseInputCapitalLetterStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets whether one-byte upper case character input will be rejected for text boxes or not.
Example	Gets the GNCDataTextBox00000 one-byte capital letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputCapitalLetterStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetRefuseInputSymbolLetterStatus	Set one-byte symbol input disabled status

Syntax	GCSNCDataTextboxSetRefuseInputSymbolLetterStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : one-byte symbol input status
Return value	None
Details	Sets whether one-byte symbol input will be rejected for text boxes or not. One of the following values is set for fStatus: 0: input enabled 1: input disabled
Example	Sets the GNCDataTextBox00000 one-byte symbol letter input disabled status in the self screen to 1. GCSNCDataTextboxSetRefuseInputSymbolLetterStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetRefuseInputSymbolLetterStatus	Get one-byte symbol input disabled status

Syntax	GCSNCDataTextboxGetRefuseInputSymbolLetterStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: input enabled 1: input disabled
Details	Gets whether one-byte symbol input will be rejected for text boxes or not.
Example	Gets the GNCDataTextBox00000 one-byte symbol letter input disabled status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputSymbolLetterStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetZeroSuppressStatus	Set zero suppress display status

Syntax	GCSNCDataTextboxSetZeroSuppressStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : zero suppress display status
Return value	None
Details	Sets whether to enable zero suppress display for text boxes or not. One of the following values is set for fStatus: 0: Normal display status 1: Zero suppress display status
Example	Sets the GNCDataTextBox00000 zero suppress display status in the self screen to 1. GCSNCDataTextboxSetZeroSuppressStatus(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetZeroSuppressStatus	Get zero suppress display status

Syntax	GCSNCDataTextboxGetZeroSuppressStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Normal display status 1: Zero suppress display status
Details	Gets whether the text box display is in normal display status or in zero suppress display status.
Example	Gets the GNCDataTextBox00000 zero suppress display status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetZeroSuppressStatus(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetString	Set display character string

Syntax	GCSNCDataTextboxSetString(nWindowNo, strName, pString);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING pString : display character string
Return value	0: setting failed 1: setting succeeded
Details	Sets the display character string. After the display character string is set, the control area is registered as a redraw area. (Note 1) By executing the display character string setting, data will be written in the NC. (Note 2) Character strings can be set even when the focus is not placed on the NC data text box.
Example	Sets the character string "ABCD" in GNCDataTextBox00000 in the self screen. GCSNCDataTextboxSetString(-1, "GNCDataTextBox00000", "ABCD") ;

GCSNCDataTextboxGetString	Get display character string

Syntax	GCSNCDataTextboxGetString(nWindowNo, strName, pString);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (o)STRING pString : display character string
Return value	0: acquisition failed 1: acquisition succeeded
Details	Stores the display character string in pString. (Note 1) By executing "Get display character string", data will be acquired from the NC. (Note 2) When the focus is placed, the data displayed in the NC text box and the NC data may differ. In that case, NC data will be acquired.
Example	Gets the GNCDataTextBox00000 display character string in the self screen in strStat. STRING strStat; GCSNCDataTextboxGetString(-1, "GNCDataTextBox00000", strStat) ;

GCSNCDataTextboxSetStringBuffer	Set input character string

Syntax

GCSNCDataTextboxSetStringBuffer(nWindowNo, strName, ulBufferSize, pStringBuffer);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)LONG ulBufferSize	: length of input character string
(i)STRING pStringBuffer	: input character string

Return value

0: setting failed
1: setting succeeded

Details

Sets input character string to the NC data text box. Enter [INPUT] to write the input character string to the NC. After the input character string is set, the control area is registered as a redraw area.
(Note 1) If the input character string is set with the focus placed on the text box, the input character string will be displayed. If not, the displayed data will be rewritten by the function of display updating cycle.
(Note 2) The length of input character string is limited by the property setting "Number of the maximum characters". If the input character string exceeds the limit, only the maximum number of characters from the top will be set.

Example

Set the input character string "ABCD" to GNCDataTextBox00000 in the self screen.

```
GCSNCDataTextboxSetStringBuffer(-1,
"GNCDataTextBox00000", 4, "ABCD") ;
```

GCSNCDataTextboxGetStringBuffer	Get input character string

Syntax

GCSNCDataTextboxGetStringBuffer(nWindowNo, strName, ulBufferSize);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(o)GMEM ulBufferSize	: global memory for saving the length of input character string

Return value

Input character string

Details

Gets the value displayed in the NC data text box.
When the focus is located at the text box, the data displayed in the NC data text box may differ from the one in the NC. In that case, the data displayed in the NC data text box will be acquired.

Example

Gets displayed character string in GNCDataTextBox00000 in the self screen to strString.

```
mem = GMEMCreate("TESTMEM", 4) ;
STRING strString ;
strString = GCSNCDataTextboxGetStringBuffer(-1,
"GNCDataTextBox00000", mem) ;
GMEMDelete(mem);
```

GCSNCDataTextboxSetGNCValue	Set display value

Syntax

GCSNCDataTextboxSetGNCValue(nWindowNo, strName, gvValue);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)GMEM gvValue	: global memory for setting value

Return value

0: setting failed
1: setting succeeded

Details

Sets the value displayed in a text box.
Numbers are converted to character strings according to the display format.
After the value displayed in a text box is set, the control area is registered as a redraw area.

Example

Sets the signed decimal integer value 100 as a value to be displayed in the GNCDataTextBox00000 in the self screen.

```
GCSNCDataTextboxSetTextType(-1, "GNCDataTextBox00000", 2) ;
mem = GMEMCreate("TESTMEM", 4) ;
GMEMSetLong(mem, 0, 100) ;
GCSNCDataTextboxSetGNCValue(-1, "GNCDataTextBox00000", mem) ;
GMEMDelete(mem);
```

GCSNCDataTextboxGetGNCValue	Get display value

Syntax

GCSNCDataTextboxGetGNCValue(nWindowNo, strName, gvValue);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(o)GMEM gvValue	: global memory to store numerical value

Return value

0: acquisition failed
1: acquisition succeeded
2: variable <empty>

Details

Gets the value displayed in a text box.
Character strings are converted to numbers according to the display format.

Example

Gets the numerical value being displayed in GNCDataTextBox00000 in the self screen as nVal : LONG integer value.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSNCDataTextboxGetGNCValue(-1, "GTextBox00000", mem) ;
nVal = GMEMGetLong(mem, 0) ;
GMEMDelete(mem);
```

GCSNCDataTextboxSetBackGroundPattern	Set background fill status

Syntax	GCSNCDataTextboxSetBackGroundPattern(nWindowNo, strName, nBackGroundPattern);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. One of the following values is set for nBackGroundPattern: -1: no fill -2: with background fill After the background fill status is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 background fill status to -2. GCSNCDataTextboxSetBackGroundPattern(-1, "GNCDataTextBox00000", -2) ;

GCSNCDataTextboxGetBackGroundPattern	Get background fill status

Syntax	GCSNCDataTextboxGetBackGroundPattern(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Background fill status -1: no fill -2: with background fill
Details	Gets the background color fill status.
Example	Gets the GNCDataTextBox00000 background fill status in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetBackGroundPattern(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetIntegerWidth	Set integer part width

Syntax	GCSNCDataTextboxSetIntegerWidth(nWindowNo, strName, ucWidth);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucWidth : The number of integer part figures
Return value	None
Details	Sets the number of integer part figures.
Example	Sets the GNCDataTextBox00000 the number of integer part figures in the self screen to 2. GCSNCDataTextboxSetIntegerWidth(-1, "GNCDataTextBox00000", 2);

GCSNCDataTextboxGetIntegerWidth	Get integer part width

Syntax	GCSNCDataTextboxGetIntegerWidth(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	The number of integer part figures
Details	Gets the number of integer part figures.
Example	Gets the GNCDataTextBox00000 the number of integer part figures in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetIntegerWidth(-1, "GNCDataTextBox00000");

GCSNCDataTextboxSetDecimalWidth	Set decimal part width

Syntax	GCSNCDataTextboxSetDecimalWidth(nWindowNo, strName, ucWidth);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucWidth : The number of decimal part figures
Return value	None
Details	Sets the number of decimal part figures.
Example	Sets the GNCDataTextBox00000 the number of decimal part figures in the self screen to 2. GCSNCDataTextboxSetDecimalWidth(-1, "GNCDataTextBox00000", 2);

GCSNCDataTextboxGetDecimalWidth	Get decimal part width

Syntax	GCSNCDataTextboxGetDecimalWidth(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	The number of decimal part figures
Details	Gets the number of decimal part figures.
Example	Gets the GNCDataTextBox00000 the number of decimal part figures in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDecimalWidth(-1, "GNCDataTextBox00000");

GCSNCDataTextboxSetScale	Set scale

Syntax	GCSNCDataTextboxSetScale(nWindowNo, strName, gmScale);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)GMEM gmScale : global memory for scale
Return value	None
Details	Sets the scale.
Example	Sets the scale of GNCDataTextBox00000 in the self screen as DOUBLE real number value : 2.54. mem = GMEMCreate("TESTMEM", 8) ; GMEMSetDouble(mem, 0, 2.54) ; GCSNCDataTextboxSetScale(-1, " GNCDataTextBox00000", mem) ; GMEMDelete(mem);

GCSNCDataTextboxGetScale	Get scale

Syntax	GCSNCDataTextboxGetScale(nWindowNo, strName, gmScale);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (o)GMEM gmScale : global memory to store scale
Return value	None
Details	Gets the scale.
Example	Gets the scale of GNCDataTextBox00000 in the self screen as nVal : DOUBLE real number value. GMEM mem; DOUBLE nVal; mem = GMEMCreate("TESTMEM", 8) ; GCSNCDataTextboxGetScale(-1, "GNCDataTextBox00000", mem) ; nVal = GMEMGetDouble(mem, 0) ; GMEMDelete(mem);

GCSNCDataTextboxSetOffset	Set offset

Syntax	GCSNCDataTextboxSetOffset(nWindowNo, strName, lOffset);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lOffset : offset
Return value	None
Details	Sets the offset.
Example	Sets the GNCDataTextBox00000 offset in the self screen to 100. GCSNCDataTextboxSetOffset(-1, "GNCDataTextBox00000", 100);

GCSNCDataTextboxGetOffset	Get offset

Syntax	GCSNCDataTextboxGetOffset(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Offset
Details	Gets the offset.
Example	Gets the GNCDataTextBox00000 offset in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetOffset(-1, "GNCDataTextBox00000");

GCSNCDataTextboxSetSystemNumber	Set part system number

Syntax	GCSNCDataTextboxSetSystemNumber(nWindowNo, strName, lSystem);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lSystem : part system number
Return value	None
Details	Sets the part system number. After the part system number is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 part system number in the self screen to 1. GCSNCDataTextboxSetSystemNumber(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetSystemNumber	Get part system number

Syntax	GCSNCDataTextboxGetSystemNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Part system number
Details	Gets the part system number.
Example	Gets the GNCDataTextBox00000 part system number in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSystemNumber(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetGround	Set ground information

Syntax	GCSNCDataTextboxSetGround(nWindowNo, strName, lGround);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lGround : ground information
Return value	None
Details	Sets the ground information. One of the following values is set for lGround: 0 : Foreground 1 : Background
Example	Sets the GNCDataTextBox00000 ground information in the self screen to 0 (foreground). GCSNCDataTextboxSetGround(-1, "GNCDataTextBox00000", 0) ;

GCSNCDataTextboxGetGround	Get ground information

Syntax	GCSNCDataTextboxGetGround(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Ground information
Details	Gets the ground information. 0 : Foreground 1 : Background
Example	Gets the GNCDataTextBox00000 ground information in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetGround(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetAxisInfo	Set axis information

Syntax	GCSNCDataTextboxSetAxisInfo(nWindowNo, strName, ulAxis);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ulAxis : axis information
Return value	None
Details	Sets the axis information. One of the following values is set for ulAxis: 0 : No axis designation 1 to 32 : Axis number
Example	Sets the GNCDataTextBox00000 axis number in the self screen to 1. GCSNCDataTextboxSetAxisInfo(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetAxisInfo	Get axis information

Syntax	GCSNCDataTextboxGetAxisInfo(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Axis information
Details	Gets the axis information.
Example	Gets the GNCDataTextBox00000 axis information in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetAxisInfo(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetSection	Set section No.
Syntax	GCSNCDataTextboxSetSection(nWindowNo, strName, lSection);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lSection : section No.
Return value	None
Details	Sets the section No.
Example	Sets the GNCDataTextBox00000 section No. in the self screen to 1. GCSNCDataTextboxSetSection(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetSection	Get section No.
Syntax	GCSNCDataTextboxGetSection(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Section No.
Details	Gets the section No.
Example	Gets the GNCDataTextBox00000 section No. in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSection(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetSubSection	Set sub-section No.

Syntax	GCSNCDataTextboxSetSubSection(nWindowNo, strName, lSubSection);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lSubSection : sub-section No.
Return value	None
Details	Sets the sub-section No.
Example	Sets the GNCDataTextBox00000 sub-section No. in the self screen to 1. GCSNCDataTextboxSetSubSection(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetSubSection	Get sub-section No.

Syntax	GCSNCDataTextboxGetSubSection(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Sub-section No.
Details	Gets the sub-section No.
Example	Gets the GNCDataTextBox00000 sub-section No. in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSubSection(-1, "GNCDataTextBox00000") ;

GCSNCDataTextboxSetDataType	Set data type

Syntax	GCSNCDataTextboxSetDataType(nWindowNo, strName, lDataType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lDataType : data type
Return value	None
Details	Sets the data type. One of the following values is set for lDataType: 0x1 : 1-byte integer type 0x2 : 2-byte integer type 0x3 : 4-byte integer type 0x5 : 8-byte real type 0x10: Character string type
Example	Sets the GNCDataTextBox00000 data type in the screen No. to 1. GCSNCDataTextboxSetDataType(-1, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetDataType	Get data type

Syntax	GCSNCDataTextboxGetDataType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0x1 : 1-byte integer type 0x2 : 2-byte integer type 0x3 : 4-byte integer type 0x5 : 8-byte real type 0x10: Character string type Other than above: Illegal setting
Details	Gets the data type.
Example	Gets the GNCDataTextBox00000 data type in the self screen in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDataType(-1, "GNCDataTextBox00000") ;

5.17 Menu

GCSMenuRefresh	Redraw menu
Syntax	GCSMenuRefresh(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	None
Details	Redraws the menu button.
Example	Redraws GNXMenu00000 in the self screen. GCSMenuRefresh(-1, "GNXMenu00000") ;

GCSMenuSetRefreshValidFlag	Set menu display
Syntax	GCSMenuSetRefreshValidFlag(nWindowNo, strName, nRefreshValidFlag);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT nRefreshValidFlag : display valid flag
Return value	None
Details	Sets whether to enable or disable the menu display. One of the following values is set for nRefreshValidFlag: 1: Menu display enabled 0: Menu display disabled When the menu display is changed from disabled to enabled, always execute GCSMenuRefresh().
Example	Sets the GNXMenu00000 menu display in the self screen to enabled and redraws the menu. GCSMenuSetRefreshValidFlag(-1, "GNXMenu00000", 1) ; GCSMenuRefresh(-1, "GNXMenu00000") ;

GCSMenuSetMenuButtonName_all	Set character string on one-row menu (10 menus)

Syntax	GCSMenuSetMenuButtonName_all(nWindowNo, strName, nMenuGroupID, strData);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)STRING strData : character string to be set
Return value	Setting result
Details	Sets the character strings (names) for all the menu buttons (10 buttons) of one-row menu. Up to seven characters can be displayed in 1 row for the menu name. Put the line break code between the rows to make two rows.
Example	Sets the names as shown below for the GNXMenu00000 menu buttons in the self screen. 1st menu : "MENU1" 2nd menu : "MENU2" 3rd menu : "MENU3" 4th menu : "MENU4" 5th menu : "MENU5" 6th menu : "MENU6" 7th menu : "MENU7" 8th menu : "MENU8" 9th menu : "MENU9" 10th menu : (empty) GCSMenuSetMenuButtonName_all(-1, "GNXMenu00000", 1, "MENU1,MENU2,MENU3,MENU4,MENU5,MENU6,MENU7,MENU8,MENU9, ") ;

GCSMenuSetMenuButtonUpperName_all	Set character string on upper row of two-row menu (10 menus)

Syntax	GCSMenuSetMenuButtonUpperName_all(nWindowNo, strName, nMenuGroupID, strData);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)STRING strData : character string to be set
Return value	Setting result
Details	Sets the character strings (names) for all the menu buttons (10 buttons) of the upper row of two-row menu. Up to seven characters can be displayed in 1 row for the menu name.
Example	Sets the names as shown below for the GNXMenu00000 upper row menu buttons in the self screen. 1st menu : "MENU1" 2nd menu : "MENU2" 3rd menu : "MENU3" 4th menu : "MENU4" 5th menu : "MENU5" 6th menu : "MENU6" 7th menu : "MENU7" 8th menu : "MENU8" 9th menu : "MENU9" 10th menu : (empty) GCSMenuSetMenuButtonUpperName_all(-1, "GNXMenu00000", 1, "MENU1,MENU2,MENU3,MENU4,MENU5,MENU6,MENU7,MENU8,MENU9, ") ;

GCSMenuSetMenuButtonLowerName_all	Set character string on lower row of two-row menu (10 menus)

Syntax	GCSMenuSetMenuButtonLowerName_all(nWindowNo, strName, nMenuGroupID, strData);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)STRING strData : character string to be set
Return value	Setting result
Details	Sets the character strings (names) for all the menu buttons (10 buttons) of the lower row of two-row menu. Seven characters maximum can be displayed on a row for the menu name. Put the line break code between the lows when two rows.
Example	Sets the names as shown below for the GNXMenu00000 lower row menu buttons in the self screen. 1st menu : "MENU1" 2nd menu : "MENU2" 3rd menu : "MENU3" 4th menu : "MENU4" 5th menu : "MENU5" 6th menu : "MENU6" 7th menu : "MENU7" 8th menu : "MENU8" 9th menu : "MENU9" 10th menu : (empty) GCSMenuSetMenuButtonLowerName_all(-1, "GNXMenu00000", 1, "MENU1,MENU2,MENU3,MENU4,MENU5,MENU6,MENU7,MENU8,MENU9, ") ;

GCSMenuSetMenuButtonName_one	Set character string on one-row menu (one menu)

Syntax

GCSMenuSetMenuButtonName_one(nWindowNo, strName, nMenuGroupID, nMenuNo, strData);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)LONG nMenuGroupID : menu group ID (fixed to 1)
(i)SHORT nMenuNo : menu No.
(i)STRING strData : character string to be set

Return value

Setting result

Details

Sets the character string (name) for the specified menu button (one button) of one-row menu.
Seven characters maximum can be displayed on a row for the menu name. Put the line break code between the lows when two rows.

Example

Sets the character strings at and subsequent to nOffset in the character string table for the GNXMenu00000 1st to 9th menu buttons in the self screen

```
SHORT nOffset ;       'Offset (0-) from the beginning of the character string table
SHORT nMenuLoop ;
STRING strMenu ;
nOffset = 1 ;
nMenuLoop = 1 ;
FOR (9)
  GCSResourceLoadString((nOffset + nMenuLoop), strMenu) ;
  GCSMenuSetMenuButtonName_one(-1, "GNXMenu00000", 1, nMenuLoop,
     strMenu) ;
  nMenuLoop = nMenuLoop + 1 ;
NEXT
```

GCSMenuSetMenuButtonUpperName_one	Set character string on upper row of two-row menu (one menu)

Syntax	GCSMenuSetMenuButtonUpperName_one(nWindowNo, strName, nMenuGroupID, nMenuNo, strData);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)STRING strData : character string to be set
Return value	Setting result
Details	Sets the character string (name) for the specified menu button (one button) of the upper row of two-row menu. Up to seven characters can be displayed in 1 row for the menu name.
Example	Sets the character strings at and subsequent to nOffset in the character string table for the GNXMenu00000 1st to 9th menu buttons in the self screen

```
SHORT nOffset ;       'Offset (0-) from the beginning of the character string table
SHORT nMenuLoop ;
STRING strMenu ;
nOffset = 1 ;
nMenuLoop = 1 ;
FOR (9)
  GCSResourceLoadString((nOffset + nMenuLoop), strMenu) ;
  GCSMenuSetMenuButtonName_one(-1, "GNXMenu00000", 1, nMenuLoop,
     strMenu) ;
  nMenuLoop = nMenuLoop + 1 ;
NEXT
```

GCSMenuSetMenuButtonLowerName_one	Set character string on lower row of two-row menu (one menu)

Syntax	GCSMenuSetMenuButtonLowerName_one(nWindowNo, strName, nMenuGroupID, nMenuNo, strData);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)STRING strData : character string to be set
Return value	Setting result
Details	Sets the character string (name) for the specified menu button (one button) of the lower row of two-row menu. Seven characters maximum can be displayed on a row for the menu name. Put the line break code between the lows when two rows.
Example	Sets the character strings at and subsequent to nOffset in the character string table for the GNXMenu00000 1st to 9th menu buttons in the self screen

```
SHORT nOffset ;       'Offset (0-) from the beginning of the character string table
SHORT nMenuLoop ;
STRING strMenu ;
nOffset = 1 ;
nMenuLoop = 1 ;
FOR (9)
  GCSResourceLoadString((nOffset + nMenuLoop), strMenu) ;
  GCSMenuSetMenuButtonName_one(-1, "GNXMenu00000", 1, nMenuLoop,
     strMenu) ;
  nMenuLoop = nMenuLoop + 1 ;
NEXT
```

GCSMenuSetMenuButtonState_one	Set state of one-row menu (one menu)

Syntax	GCSMenuSetMenuButtonState_one(nWindowNo, strName, nMenuGroupID, nMenuNo, nMenuState);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)LONG nMenuState : menu selection status
Return value	Setting result
Details	Sets the selection state for the specified one-row menu button (one button). One of the following values is set for nMenuState: 0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Example	Sets the menu selection status to ON for the GNXMenu00000 1st menu button in the self screen. GCSMenuSetMenuButtonState_one(-1, "GNXMenu00000", 1, 1, 1) ;

GCSMenuSetMenuButtonUpperState_one	Set state of upper row of two-row menu (one menu)

Syntax	GCSMenuSetMenuButtonUpperState_one(nWindowNo, strName, nMenuGroupID, nMenuNo, nMenuState);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)LONG nMenuState : menu selection status
Return value	Setting result
Details	Sets the selection state for the upper row of the specified two-row menu button (one button). One of the following values is set for nMenuState: 0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Example	Sets the menu selection status to ON for the upper row of the GNXMenu00000 1st menu button in the self screen. GCSMenuSetMenuButtonUpperState_one(-1, "GNXMenu00000", 1, 1, 1) ;

GCSMenuSetMenuButtonLowerState_one	Set state of lower row of two-row menu (one menu)

Syntax	GCSMenuSetMenuButtonLowerState_one(nWindowNo, strName, nMenuGroupID, nMenuNo, nMenuState);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)LONG nMenuState : menu selection status
Return value	Setting result
Details	Sets the selection state for the lower row of the specified two-row menu button (one button). One of the following values is set for nMenuState: 0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Example	Sets the menu selection status to ON for the lower row of the GNXMenu00000 1st menu button in the self screen. GCSMenuSetMenuButtonLowerState_one(-1, "GNXMenu00000", 1, 1, 1) ;

GCSMenuGetMenuButtonState_one	Get state of one-row menu (one menu)

Syntax	GCSMenuGetMenuButtonState_one(nWindowNo, strName, nMenuGroupID, nMenuNo);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No.
Return value	0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Details	Gets the selection state for the specified one-row menu button (one button).
Example	Gets the menu selection status of the GNXMenu00000 1st menu button in the self screen. LONG Stat; Stat = GCSMenuSetMenuButtonState_one(-1, "GNXMenu00000", 1, 1) ;

GCSMenuGetMenuButtonUpperState_one	Get state of upper row of two-row menu (one menu)

Syntax	GCSMenuGetMenuButtonUpperState_one(nWindowNo, strName, nMenuGroupID, nMenuNo);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No.
Return value	0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Details	Gets the selection state for the upper row of the specified two-row menu button (one button).
Example	Gets the menu selection status of the upper row of the GNXMenu00000 1st menu button in the self screen. LONG Stat; Stat = GCSMenuGetMenuButtonUpperState_one(-1, "GNXMenu00000", 1, 1) ;

GCSMenuGetMenuButtonLowerState_one	Get state of lower row of two-row menu (one menu)

Syntax	GCSMenuGetMenuButtonLowerState_one(nWindowNo, strName, nMenuGroupID, nMenuNo);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No.
Return value	0: Menu button OFF (not selected, normal) 1: Menu button ON (selected) -1: Menu button disabled
Details	Gets the selection state for the lower row of the specified two-row menu button (one button).
Example	Gets the menu selection status of the lower row of the GNXMenu00000 1st menu button in the self screen. LONG Stat; Stat = GCSMenuGetMenuButtonLowerState_one(-1, "GNXMenu00000", 1, 1) ;

GCSMenuSetMenuButtonOffIcon_one	Set icon of one-row menu (menu) at OFF

Syntax	GCSMenuSetMenuButtonOffIcon_one(nWindowNo, strName, nMenuGroupID, nMenuNo, nIconID);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)SHORT nIconID : icon resource ID at the time of OFF
Return value	Setting result
Details	Sets the icon resource ID at the time of OFF for the specified one-row menu button (one button).
Example	Set the first icon resource ID at the time of OFF for the GNXMenu00000 1st menu button in the self screen. LONG Stat; Stat = GCSMenuSetMenuButtonOffIcon_one(-1, "GNXMenu00000", 1, 1, 1) ;

GCSMenuSetMenuButtonOnIcon_one	Set icon of one-row menu (menu) at ON

Syntax	GCSMenuSetMenuButtonOnIcon_one(nWindowNo, strName, nMenuGroupID, nMenuNo, nIconID);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nMenuGroupID : menu group ID (fixed to 1) (i)SHORT nMenuNo : menu No. (i)SHORT nIconID : icon resource ID at the time of ON
Return value	Setting result
Details	Sets the icon resource ID at the time of ON for the specified one-row menu button (one button).
Example	Set the first icon resource ID at the time of ON for the GNXMenu00000 1st menu button in the self screen. LONG Stat; Stat = GCSMenuSetMenuButtonOnIcon_one(-1, "GNXMenu00000", 1, 1, 1) ;

GCSMenuSendProcessID	Issue process ID

Syntax	GCSMenuSendProcessID(nWindowNo, strName, nProcessID);
Argument	(i)LONG nWindowNo : screen No. (i)STRING strName : control name (i)LONG nProcessID : process ID
Return value	Setting result
Details	Issues process ID to the panel One of the following values is set for nWindowNo. 1000: Monitor screen 2000: Setup screen 3000: Edit screen One of the following values is set for nProcessID: 0: Close custom window 1: Display Monitor screen 2: Display Setup screen 3: Display Edit screen 4: Display Diagnosis screen 5: Display Maintenance screen 6: Display SFP screen 7: Display F0 screen 8: Display the screen for window display 9: Display the screen for window selection 10: Display menu list 11: Switch part system 12: Display parameter/operation guidance 13: Cancel direct screen selection
Example	Closes the custom window registered to menu customization. LONG Stat; Stat = GCSMenuSendProcessID(1000, 0, 0) ;

GCSMenuChangeMenuType	Switch display between one-row menu and two-row menu

Syntax	GCSMenuChangeMenuType (nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Setting result
Details	Switches the display between the one-row menu and the two-row menu. Switching can be executed between the following menu types. - One-row menu (VGA) (Classic) <--> Two-row menu (VGA) (Classic) - One-row menu (XGA) (Classic) <--> Two-row menu (XGA) (Classic) - One-row menu (VGA) <--> Two-row menu (VGA) - One-row menu (XGA) <--> Two-row menu (XGA)
Example	Switches the display of the GNXMenu00000 menu on self screen between the one-row menu and the two-row menu. LONG Stat; Stat = GCSMenuChangeMenuType(-1, "GNXMenu00000") ;

GCSMenuSetDstObject()	Specify called control

Syntax

GCSMenuSetDstObject (nWindowNo, strName, strDstName);

Argument

(i) LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i) STRING strName : Control name
(i) STRING strDstName : Called control name

Return value

Setting result

Details

Specify the called control.
(Note) After the specified called control is called, the called control will be focused.

Example

Specify GBASICCONTROL00005 in the self screen as the called control in GNXMENU00000 in the self screen.

```
LONG nStatus = 0;
nStatus = GCSSetDstObject (-1,
"GNXMENU00000","GBASICCONTROL00005");
```

OnKeyPress of BasicControl is called by using the key code of the menu return button as the argument when the menu return button for the menu part is pressed.

```
$GBasicControl00005-OnKeyPress
        :
  if(LLPARAM == 122)                    ' Menu return button
    ' Process when the menu return button is pressed
        :
  endif;
$End
```

GCSMenuSetNextMenuButtonState	Set menu switching button selection state

Syntax
GCSMenuSetNextMenuButtonState(nWindowNo, strName, nButtonState, lType)

Argument
(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)LONG nButtonState : A button selection state
(i)LONG lType : Type
bit12: Specifies whether to update drawing.
ON: Does not update drawing.
OFF: Updates drawing.

Return value
Setting result

Details
Sets the selection state for the menu switching button.
Set any of the following values for nButtonState.
0: Button OFF (Not selected, default)
-1: Button enabled

Example
Sets the button selection state to normal for the menu switching button of GNXMenu00000 on self screen.
GCSMenuSetNextMenuButtonState(-1, "GNXMenu00000", 0, 0) ;

GCSMenuSetOpeMessage	Set operation message

Syntax
GCSMenuSetOpeMessage(nWindowNo, strName, strOpeMessageString)

Argument
(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)STRING strOpeMessageString : Operation message

Return value
Setting result

Details
Sets operation messages using character strings.
Up to 40 characters can be used for an operation message.

Example
Sets the operation message part of GNXMenu00000 to the "Message1" character string on self screen.
GCSMenuSetOpeMessage(-1, "GNXMenu00000", "Message1") ;

GCSMenuBeginAnimState	Start menu animated motion setting

Syntax

GCSMenuBeginAnimState(nWindowNo, strName, nMenuAnimType)

Argument

(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)SHORT nMenuAnimType : Type of menu animation
0: No animation
1: Rightward sliding
2: Upward sliding
3: Downward sliding

Return value

0: Normal end

Details

Starts the setting for animated motion.
After the start setting for animated motion has been completed, set the updating function for the menu to be displayed by animation (setting of name, normal/highlighted/hidden display, icons, etc.).
After the GCSMenuEndAnimState() setting has been completed, animated motion starts.
Set any of the following values for nMenuAnimType.
0: No animation
1: Rightward sliding
2: Upward sliding
3: Downward sliding
(Note) If the operation parameter (#8976 Menu animation disabled) is enabled, menu animation is not executed.

Example

Executes menu animation after the GNXMenu00000 menu name on self screen is changed.

```
GCSMenuBeginAnimState(-1, "GNXMenu00000", 1);
        // Starting the menu state change
GCSMenuSetMenuButtonUpperName_all(-1, "GNXMenu00000",
1,"MENU1,MENU2,MENU3,MENU4,MENU5,MENU6,MENU7,ME
NU8,MENU9,");
        // Setting menu names
GCSMenuEndAnimState(-1, "GNXMenu00000");
        // Completing the menu state change and starting
rightward animation 4.3.23
```

GCSMenuEndAnimState	Complete menu animated motion setting

Syntax

GCSMenuEndAnimState(nWindowNo, strName)

Argument

(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name

Return value

0: Normal end

Details

Sets the completion for animated motion setting.
Set the updating function for the menu to be displayed by animation (setting of name, normal/highlighted/hidden display, icons, etc.). Calling the this function starts animated motion.

(Note) If the control parameter (#8976 Menu Anim Invalid) is enabled, menu animation is not executed.

Example

Executes menu animation after the GNXMenu00000 menu name on self screen is changed.

```
GCSMenuBeginAnimState(-1, "GNXMenu00000", 1);
        // Starting the menu state change
GCSMenuSetMenuButtonUpperName_all(-1, "GNXMenu00000",
1,"MENU1,MENU2,MENU3,MENU4,MENU5,MENU6,MENU7,ME
NU8,MENU9, ");
        // Setting menu names
GCSMenuEndAnimState(-1, "GNXMenu00000");
        // Completing the menu state change and starting
rightward animation
```

GCSMenuStopAnim	Stop menu animation

Syntax

GCSMenuStopAnim(nWindowNo, strName)

Argument

(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name

Return value

0: Normal end

Details

Stops menu animation and displays the menu after the state has been changed.
Use this function to stop menu animation that is running using GCSMenuEndAnimState().
If you call this function when menu animation is not executed, the stop process is not executed and the program ends normally.

Example

Stops the animation of GNXMenu00000 on self screen, and sets the operation message part to the "Message1" character string.

```
GCSMenuStopAnim(-1, "GNXMenu00000");
        // Stopping menu animation
GCSMenuSetOpeMessage(-1, "GNXMenu00000", "Message1") ;
        // Setting the "Message1" character string
```

5.18 FileInOut

GCSIOCheckUpDateFile	Check existence of file

Syntax	GCSIOCheckUpDateFile(nWindowNo, strName, strFileInfo);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING strFileInfo : file path to be checked (up to 128 characters in the path)
Return value	TRUE :The file exists FALSE :The file does not exist
Details	Checks if the specified file exists.
Example	Checks if the common variable file " COMMON.VAR " exists in the "COMMON" directory of "Drive D" using GNXFileInOut00000 in the self screen. LONG Stat; Stat = GCSIOCheckUpDateFile(-1, "GNXFileInOut00000", "D:/COMMON/COMMON.VAR");

GCSIOFileTransfer	Start file transfer

Syntax

GCSIOFileTransfer(nWindowNo, strName, strInFileInfo, strOutFileInfo);

Argument

(i)LONG nWindowNo : Screen No. (Specify -1 for self screen.)
(i)STRING strName : Control name
(i)STRING strInFileInfo : Source file path (up to 128 characters in the path)
(i)STRING strOutFileInfo : Destination file path (up to 128 characters in the path)

Return value

Error code
0 : Normal completion
1 : Unable to open the transfer source file
2 : Unable to open the transfer destination file
5 : Unable to read the transfer source file
8 : Unable to write to the transfer destination file
22: Specified file does not exist
24: Memory capacity over
31: Specified file is in automatic operation
32: Specified file is in program check
33: Specified file is in program restart
34: An illegal character is included in Argument
35: Unable to transfer without password input
37: The same file is specified for the transfer source and destination
39: Excessive number of programs is registered
60: File name length is longer than the maximum
61: Directory is specified for transfer source file name (Note)
63: FLD drive is not connected
66: The number of files registered is excessive
67: Edit lock B is active
68: Edit lock C is active
70: Data serve is not connected
(M800V/M80V/M800/M80 (Windows-based display unit) and M700VW)
71: In PLC RUN
72: Data protect is active
74: Memory card is not connected
84: Write protect is active on the specified device
86: Specified path is incorrect
87: In transferring
88: The file to transfer is not specified

Details

Inputs and outputs NC data between NC memory/NC memory 2 and an external device.

Example

Transfers the common variable file "COMMON.VAR" in the "COMMON" directory of "Drive D" using GNXFileInOut00000 in the self screen.

```
LONG Stat;
Stat = GCSIOFileTransfer(-1, "GNXFileInOut00000",
"D:/COMMON/COMMON.VAR", "M01:/DAT/COMMON.VAR");
```

(Note) A full path has to be used to specify the file name in the file information to be given to the argument. However, an error does not occur even when a file name is omitted from the transfer destination file information (the device and directory have to be specified). If the file name is omitted, transfer is performed with the transfer source file name.

GCSIODeleteFile	Delete file/directory

Syntax

GCSIODeleteFile(nWindowNo, strName, strFileInfo);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)STRING strFileInfo : file path to be deleted (up to 128 characters in the path)

Return value

Error code
0: Normal completion
15: Unable to delete the specified file (directory)
22:Specified file does not exist
31: Specified file is in automatic operation
32: Specified file is in program check
33: Specified file is in program restart
34: An illegal character is included in Argument
57: A file exists in the directory
63:FLD drive is not connected
67: Edit lock B is active
68:Edit lock C is active
70: Data server is not connected
(M800V/M80V/M800/M80 (Windows-based display unit) and M700VW)
72: Data protect is active
74: Memory card is not connected
84: Write protect is active on the specified device
86: Specified path is incorrect

Details

Deletes the specified file (directory).

Example

Deletes the common variable file "COMMON.VAR" from the "COMMON" directory of "Drive D" using GNXFileInOut00000 in the self screen.

```
LONG Stat;
Stat = GCSIODeleteFile(-1, "GNXFileInOut00000",
"D:/COMMON/COMMON.VAR");
```

GCSIOCreateDirectory	Create directory

Syntax	GCSIOCreateDirectory(nWindowNo, strName, strDirInfo);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING strDirInfo : path of directory to be created (up to 128 characters in the path)
Return value	Error code 0: Normal completion 18: Unable to create directory 21: Unable to create directory on this device 34: An illegal character is included in Argument 39: Excessive number of programs is registered 63: FLD drive is not connected 66: The number of files registered is excessive 70: Data server is not connected (M800V/M80V/M800/M80 (Windows-based display unit) and M700VW) 74: Memory card is not connected 84: Write protect is active on the specified device 85: Specified directory already exists 86: Specified path is incorrect
Details	Creates the specified directory.
Example	Creates the "TEST" directory on "Drive D" using GNXFileInOut00000 in the self screen. LONG Stat; Stat = GCSIOCreateDirectory(-1, "GNXFileInOut00000", "D:/TEST");

GCSIORenameFile	Rename file/directory

Syntax

GCSIORenameFile(nWindowNo, strName, strOldFileName, strNewFileName);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)STRING strOldFileName : file path before change (up to 128 characters in the path)
(i)STRING strNewFileName : file path after change (up to 128 characters in the path)

Return value

Error code
0 : Normal completion
16: Unable to rename the selected file
29: Different device has been specified
31: Specified file is in automatic operation
32: Specified file is in program check
33: Specified file is in program restart
34: An illegal character is included in Argument
37: The same file name is specified for the names before and after rename
60: File name length is longer than the maximum
63:FLD drive is not connected
67:Edit lock B is active
68: Edit lock C is active
70: Data server is not connected
(M800V/M80V/M800/M80 (Windows-based display unit) and M700VW)
72: Data protect is active
74: Memory card is not connected
84: Write protect is active on the specified device
86: Specified path is incorrect.

Details

Changes the file (directory) name to the specified one.

Example

Changes the "COMMON" directory of "Drive D" to "TEST" using GNXFileInOut00000 in the self screen.

```
LONG Stat;
Stat = GCSIOCreateDirectory(-1, "GNXFileInOut00000",
"D:/COMMON", "D:/TEST");
```

GCSInterrupt	Interrupt file transfer
Syntax	GCSInterrupt(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Error code 0 : Normal completion
Details	Interrupts the file transfer that is being carried out.
Example	Interrupts the file transfer of GNXFileInOut00000 in the self screen. LONG Stat; Stat = GCSInterrupt(-1, "GNXFileInOut00000");

GCSIOInitializeMessage	Initialize file input/output part
Syntax	GCSIOInitializeMessage(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	None
Details	Initializes the drawing of message and bar graph.
Example	Initializes the message and bar graph displayed using GNXFileInOut00000 in the self screen. GCSIOInitializeMessage(-1, "GNXFileInOut00000");

GCSGetLastError	Get error information

Syntax

GCSGetLastError(nWindowNo, strName);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name

Return value

Error code
0: Normal completion
1: Unable to open the transfer source file
2: Unable to open the transfer destination file
5: Unable to read the transfer source file
8: Unable to write to the transfer destination file
15: Unable to delete the specified file (directory)
16: Unable to rename the selected file
18: Unable to create directory
21: Unable to create directory on this device
22: Specified file does not exist
24: Memory capacity over
29: Different device has been specified
31: Specified file is in automatic operation
32: Specified file is in program check
33: Specified file is in program restart
34: An illegal character is included in Argument
35: Unable to transfer without password input
37: The same file is specified for the transfer source and destination
39: Excessive number of programs is registered
57: A file exists in the directory
60: The file name length is longer than the maximum
61: Directory is specified for transfer source file name
63: FLD drive is not connected
66: The number of files registered is excessive
67: Edit lock B is active
68: Edit lock C is active
70: Data server is not connected
(M800V/M80V/M800/M80 (Windows-based display unit) and M700VW)
71: In PLC RUN
72: Data protect is active
74: Memory card is not connected
79: Serial number is incorrect
84: Write protect is active on the specified device
85: The directory already exists
86: Specified path is incorrect
87: In transferring
88: The file to transfer is not specified

Details

Gets the information of currently occurring errors, including the error that occurs during file transfer.

Example

Gets the information of the errors that occur during file transfer of GNXFileInOut00000 in the self screen.

```
LONG Stat;
Stat = GCSGetLastError(-1, "GNXFileInOut00000");
```

GCSIOGetTransferStatus	Get transfer status
Syntax	GCSIOGetTransferStatus(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Transfer status 0:No status 1:Being transferred 2:Transfer completed 3:In Error
Details	Gets the file transfer status.
Example	Gets the file transfer status of GNXFileInOut00000 in the self screen. LONG Stat; Stat = GCSIOGetTransferStatus(-1, "GNXFileInOut00000");

5.19 AlarmMessage

GCSAlarmMessageSetDispMessageType	Set character string display form

Syntax	GCSAlarmMessageSetDispMessageType(nWindowNo, strName, ucDispType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucDispType : character string display form
Return value	None
Details	Sets the property "Character string display form" to change the type to display character string. One of the following values is set for ucSystemNumber: 0: When the message is split, the residual message is not displayed at the next drawing update. 1: When the message is split, the residual message is displayed at the next drawing update. 2: A character string is displayed according to the setting of the parameter (#11021 PLC mesg disp type).
Example	Sets GNXAlarmMessage00000 in the self screen so that if a message is split, the residual message is not displayed at the next drawing update. GCSAlarmMessageSetDispMessageType(-1," GNXAlarmMessage00000", 0);

5.20 MonitorStatus

GCSMonStatusSetDispSysNumber	Set part system number to display

Syntax	GCSMonStatusSetDispSysNumber(nWindowNo, strName, ucSystemNumber);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucSystemNumber : part system number to display (0 to 8)
Return value	TRUE :setting succeeded FALSE :setting failed
Details	Sets the property "Part system designation" to change the part system to display. However, if "0" is given to the argument, the number of part systems to be displayed is the same as that of the valid part systems (the number of part systems to be displayed varies according to the part size).
Example	Changes the part system displayed with GNXMonitorStatus00000 in the self screen to the part system number 2. LONG Stat; Stat = GCSMonStatusSetDispSysNumber(-1, "GNXMonitorStatus00000", 2);

GCSMonStatusGetDispSysNumber	Get part system number to display

Syntax	GCSMonStatusGetDispSysNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	The part system number being displayed (0 to 8)
Details	Gets the value of the property "NumberOfSystems".
Example	Gets the part system displayed with GNXMonitorStatus00000 in the self screen in MonStat. LONG MonStat; MonStat = GCSMonStatusGetDispSysNumber(-1, "GNXMonitorStatus00000");

5.21 Counter

GCSCounterAxisChange	Change axis to be displayed
Syntax	GCSCounterAxisChange(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Error 1: Normal completion
Details	Changes the axis to be displayed when the number of control axes has exceeded the maximum number of lines for display. After the axis is set, the control area is registered as a redraw area.
Example	Changes the axis to be displayed on GNXCounter00000 in the self screen. LONG lStatus; lStatus = GCSCounterAxisChange(-1, "GNXCounter00000") ;

GCSCounterSetCounterType	Set counter type

Syntax

GCSCounterSetCounterType(nWindowNo, strName, lCounterType);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)LONG lCounterType : counter type

Return value

0: Error
1: Normal completion

Details

Sets the counter type.
Sets lCounterType to one of the following values.
0: Current position
1: Workpiece coordinate position
2: Machine's position
3: Program position
7: Remaining command
8: Manual interruption amount
9: Next command
10: Restart position
11: Restart remain distance
15: Tip workpiece coordinate position
17: Tool axis movement
18: Tip machining position
19: Relative position
20: Table coordinate position
21: Work installation position
22: Inclined surface coordinate position
26: PLC axis position
After the counter type is set, the control area is registered as a redraw area.

Example

Sets the GNXCounter00000 counter type in the self screen to 1.

```
LONG lStatus;
lStatus = GCSCounterSetCounterType(-1, "GNXCounter00000",
1) ;
```

5.22 CycleTime

GCSCycleTimeSetTimeType	Set type of time to be displayed

Syntax

GCSCycleTimeSetTimeType(nWindowNo, strName, lArea, lTimeType);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)LONG lArea : display position (0: upper / 1: lower)
(i)LONG lTimeType : type of time

Return value

0: Error
1: Normal completion

Details

Sets the type of time to be displayed.
Sets lTimeType to one of the following values.
0: Default (upper: automatic start / lower: cycle time)
1: Date
2: Time
3: Power-ON time
4: Automatic operation time
5: Automatic start time
6: External accumulated time 1
7: External accumulated time 2
8: Cycle time
9: Cut time
After the time type is set, the control area is registered as a redraw area.

Example

Sets the GNXCycleTime00000 time type (upper) in the self screen to 1.

```
LONG lStatus;
lStatus = GCSCycleTimeSetTimeType(-1, "GNXCycleTime00000", 0, 1) ;
```

5.23 LoadMeter

GCSLoadMeterSetDispType	Set display type of load meter
Syntax	GCSLoadMeterSetDispType(nWindowNo, strName, lDispType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG lDispType : display type
Return value	None
Details	Sets the display type. Sets lDispType to one of the following values. 0 : Display spindle load (Load meter 1) only (upper) 1 : Display Z axis load (Load meter 2) only (lower) 2 : Display both spindle and Z axis loads 3 : Display Z axis load (Load meter 2) only (upper) After the display type is set, the control area is registered as a redraw area.
Example	Sets the GNXLoadMeter00000 display type in the self screen to 1. GCSLoadMeterSetDispType(-1, "GNXLoadMeter00000", 1) ;

5.24 SPCommand

GCSSPCommandSetDispSpAxis	Set display order of spindles

Syntax

GCSSPCommandSetDispSpAxis(nWindowNo, strName, gmSPs, lSpNum);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)GMEM gmSPs : global memory for storing spindle Nos.
(i)LONG lSpNum : the number of spindles to be displayed

Return value

None

Details

Sets the display order of spindles.
If you wish to display the 1st spindle, set the spindle No. to 0.
Either the setting value of the property item S_Number or that of the argument lSpNum, whichever is smaller, is enabled.
After the display order is set, the control area is registered as a redraw area.
(Note) Prepare global memories for storing spindle Nos. by the number of spindles to be displayed (Four bytes/spindle).

Example

Sets the GNXSPCommand00000 spindle No. in the self screen to the 1st and 3rd spindles.

```
GMEM mem;
mem = GMEMCreate("DISPSP", 8); 'Create global memories for two spindles
GMEMSetLong(mem, 0, 0); 'Set the 1st spindle (0) in the first memory
GMEMSetLong(mem, 4, 2); 'Set the 3rd spindle (2) in the second memory
GCSSPCommandSetDispSpAxis(-1, "GNXSPCommand00000", mem, 2) ;
GMEMDelete(mem);
```

5.25 NCTable

GCSNCTableSetCellBackColor	Set background color of each cell

Syntax	GCSNCTableSetCellBackColor(nWindowNo, strName, gcColor,usLine, usRow);
Argument	(i)LONG nWindowNo : screen No.(Specify -1 for self screen) (i)STRING strName : control name (i)LONG gcColor : color code for background color (RGB) (i)SHORT usLine : line No. (from 0) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Setting result 0x0 : Normal l0x2001 : cell No. is out of range
Details	Sets the background color to the designated cell. After the background color is set, the control area is registered as a redraw area.
Example	Sets green (0x008000) to the background color of the cell in GNCTable00000 line No.3, row No.2 in the self screen. LONG Stat; Stat = GCSNCTableSetCellBackColor(-1, "GNCTable00000", H008000, 3, 2);

GCSNCTableSetCellString	Set character string of each cell

Syntax	GCSNCTableSetCellString(nWindowNo, strName, pString, usLine, usRow);
Argument	(i)LONG nWindowNo : screen No.(Specify -1 for self screen) (i)STRING strName : control name (i)STRING pString : character string to be set (0 to 128 characters) (i)SHORT usLine : line No. (from 0) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2095 : failed to reserve memory
Details	Sets the character string to the designated cell. After the character string is set, the control area is registered as a redraw area.
Example	Sets character string "Test" to the cell in GNCTable00000 line No.3, row No.2 in the self screen. LONG Stat; Stat = GCSNCTableSetCellString(-1, "GNCTable00000", "Test", 3, 2);

GCSNCTableGetCellString	Get string for each cell

Syntax	GCSNCTableGetCellString(nWindowNo, strName, pString, usLine, usRow);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (o)STRING pString : character string to be acquired (i)SHORT usLine : line No. (from 0) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Getting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2094 : length of the acquired character string is illegal
Details	Gets the charater string in the designated cell.
Example	Gets up to 50 characters of the displayed character string in GNCTable00000 line No.3, row No.2 in the self screen to "pString". STRING pString; LONG Stat; Stat = GCSNCTableGetCellString(-1, "GNCTable00000", pString,3, 2);

GCSNCTableSetSubCursorNum	Set position of sub cursor

Syntax	GCSNCTableSetSubCursorNum(nWindowNo, strName, usLine, usRow);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (i)SHORT usLine : line No. (from 0) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2091 : sub cursor is invalid
Details	Sets the display position of the sub cursor. After the position is set, the control area is registered as a redraw area.
Example	Sets the display position of sub cursor to GNCTable00000 line No.3, row No.2 in the self screen. LONG Stat; Stat = GCSNCTableSetSubCursorNum(-1, "GNCTable00000", 3, 2);

GCSNCTableGetSubCursorNum	Get position of sub cursor

Syntax	GCSNCTableGetSubCursorNum(nWindowNo, strName, gmPoint);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (o)GMEM gmPoint : global memory for line and row No.
Return value	(Long) Getting result 0x0 : Normal 0x2091 : sub cursor is invalid
Details	Gets the display position of sub cursor

Example	Gets the display position of sub cursor in GNCTable00000 in the self screen to:
	nLine: line No. of sub cursor
	nRow: row No. of sub cursor
	SHORT nLine;
	SHORT nRow;
	LONG Stat;
	GMEM mem;
	mem = GMEMCreate("MATRIX", 4);
	Stat = GCSNCTableGetSubCursorNum(-1, "GNCTable00000", mem);
	nLine = GMEMGetShort(mem, 0);
	nRow = GMEMGetShort(mem, 2);
	GMEMDelete(mem);

GCSNCTableSetCellWidth	Set width of cell

Syntax	GCSNCTableSetCellWidth(nWindowNo, strName, usRowNum, nWidth, nRatioMode);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (i)SHORT usRowNum : row No. (from 0) (i)SHORT nWidth : row width (from 1) (i)SHORT nRatioMode : row ratio mode (0:ratio 1:Pixel)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range
Details	Sets width of the designated cell. After the width is set, the control area is registered as a redraw area.
Example	Sets width of cell in GNCTable00000 line No.3, row in the self screen to "30" with Pixel mode. LONG Stat; Stat = GCSNCTableSetCellWidth(-1, "GNCTable00000", 3, 30, 1);

GCSNCTableGetCellNumFromPoint	Get cell No. including designated coordinate

Syntax

GCSNCTableGetCellNumFromPoint(nWindowNo, strName, ptPoint gmPoint);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)GMEM gptLPoint : global memory for coordinate
(o)GMEM gmPoint : global memory for line and row No.

Return value

(Long) Getting result
0x0 : Normal
0x2001 : cell No. is out of range

Details

Gets cell No. including the designated coordinate (absolute coordinate).

Example

Get the cell position corresponding to
X coordinate : 100 (Pixel), Y coordinate : 50 (Pixel) in
GNCTable00000" in the self screen as:
nLine: cell's line No. including corresponding coordinate
nRow: cell's row No. including corresonding coordinate

```
GMEM mem1;
mem1 = GMEMCreate("TESTMEM", 4);
GMEMSetShort(mem1, 0, 100);
GMEMSetShort(mem1, 2, 50);
SHORT nLine;
SHORT nRow;
LONG Stat;
GMEM mem2;
mem2 = GMEMCreate("MATRIX", 4);
Stat = GCSNCTableGetCellNumFromPoint(-1, "GNCTable00000",
mem1, mem2 );
nLine = GMEMGetShort(mem2, 0);
nRow = GMEMGetShort(mem2, 2);
GMEMDelete(mem1);
GMEMDelete(mem2);
```

GCSNCTableSetLineTitleString	Set line title

Syntax

GCSNCTableSetLineTitleString(nWindowNo, strName, pString, usLine);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)STRING pString : character string to be set (0 to 128)
(i)SHORT usLine : line No. (from 0)

Return value

(Long) Setting result
0x0 : Normal
0x2001 : cell No. is out of range
0x2092 : title hidden mode
0x2095 : failed to reserve memory

Details

Sets title to the designated line.
After the title is set, the control area is registered as a redraw area.

Example

Set character string "No.1" to GNCTable00000 line No.3 in the self screen as the title.

```
LONG Stat;
Stat = GCSNCTableSetLineTitleString(-1, "GNCTable00000",
″No.1″, 3);
```

GCSNCTableGetLineTitleString	Get line title

Syntax	GCSNCTableGetLineTitleString(nWindowNo, strName, pString, usLine);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (o)STRING pString : character string to be acquired (i)SHORT usLine : line No. (from 0)
Return value	(Long) Getting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2092 : title hidden mode 0x2094 : length of the acquired character string is illegal
Details	Gets the line title in the designated line.
Example	Gets the displayed character string for a line title in GNCTable00000 line No.3 in the self screen. STRING pString; LONG Stat; Stat =GCSNCTableGetLineTitleString(-1, "GNCTable00000", pString, 3);

GCSNCTableSetRowTitleString	Set row title

Syntax

GCSNCTableSetRowTitleString(nWindowNo, strName, pString, usRow);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)STRING pString : character string to be set (0 to 128)
(i)SHORT usRow : row No. (from 0)

Return value

(Long) Setting result
0x0 : Normal
0x2001 : cell No. is out of range
0x2092 : title hidden mode
0x2095 : failed to reserve memory

Details

Sets title to the designated row.
After the title is set, the control area is registered as a redraw area.

Example

Set character string "SYSTEM1" to GNCTable00000 row No.2 in the self screen as a title.

```
LONG Stat;
Stat =
GCSNCTableSetRowTitleString(-1,"GNCTable00000","SYSTEM1
", 2);
```

GCSNCTableGetRowTitleString	Get row title

Syntax	GCSNCTableGetRowTitleString(nWindowNo, strName, pString, usRow);
Argument	(i)LONG nWindowNo : screen No.(Specify -1 for self screen) (i)STRING strName : control name (o)STRING pString : character string to be acquired (i)SHORT usRow : row No. (from 0)
Return value	(Long) Getting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2092 : title hidden mode 0x2094 : length of the acquired character string is illegal
Details	Gets the title in the designated row.
Example	Gets the displayed character string for a title in GNCTable00000 row No.2 in the self screen. STRING pString; LONG Stat; Stat =GCSNCTableGetRowTitleString(-1, "GNCTable00000", pString, 2);

GCSNCTableSetLineTitleStringPosition	Set character string position of line title

Syntax

GCSNCTableSetLineTitleStringPosition(nWindowNo, strName, ucHPosition, ucVPosition, usLine);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)CHAR ucHPosition : character position - horizontal
0 : Align left
1 : Center
2 : Align right
(i)CHAR ucVPosition : character position - vertical
0 : Align top
1 : Center
2 : Align bottom
(i)SHORT usLine : line No. (from 0)

Return value

(Long) Setting result
0x0 : Normal
0x2001 : cell No. is out of range
0x2092 : title hidden mode
0x2093 : value is out of setting range

Details

Sets a character string position of designated line title.
After the position is set, the control area is registered as a redraw area.

Example

Set "Align left" to "horizontal" and set "Align bottom" to "vertical" of the character position of line title in GNCTable00000 line No.3 in the self screen.

```
LONG Stat;
Stat = GCSNCTableSetLineTitleStringPosition(-1,
"GNCTable00000", 0, 2, 3);
```

GCSNCTableSetRowTitleStringPosition	Set character string position of row title

Syntax

GCSNCTableSetRowTitleStringPosition(nWindowNo, strName, ucHPosition, ucVPosition, usRow);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)CHAR ucHPosition : character position - horizontal
0 : Align left
1 : Center
2 : Align right
(i)CHAR ucVPosition : character position - vertical
0 : Align top
1 : Center
2 : Align bottom
(i)SHORT usRow : row No. (from 0)

Return value

(Long) Setting result
0x0 : Normal
0x2001 : cell No. is out of range
0x2092 : title hidden mode
0x2093 : value is out of setting range

Details

Sets a character string position of designated row title.
After the position is set, the control area is registered as a redraw. area.

Example

Set "Center" to "horizontal" and set "Align top" to "vertical" of the character position of row title in GNCTable00000 line No.2 in the self screen.

```
LONG Stat;
Stat = GCSNCTableSetRowTitleStringPosition(-1,
"GNCTable00000", 1, 0, 2);
```

GCSNCTableSetCellStringPosition	Set character string position of data area

Syntax

GCSNCTableSetCellStringPosition(nWindowNo, strName, ucHPosition, ucVPosition, usLine, usRow);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(i)CHAR ucHPosition : character position - horizontal
0 : Align left
1 : Center
2 : Align right
(i)CHAR ucVPosition : character position - vertical
0 : Align top
1 : Center
2 : Align bottom
(i)SHORT usLine : line No. (from 0)
(i)SHORT usRow : row No. (from 0)

Return value

(Long) Setting result
0x0 : Normal
0x2001 : cell No. is out of range
0x2093 : value is out of setting range

Details

Sets a character string position of designated data area.
After the position is set, the control area is registered as a redraw area.

Example

Set "Align right" to "horizontal" and set "Center" to "vertical" of the character string position of data area in GNCTable00000 line No.3, row No.2 in the self screen.

```
LONG Stat;
Stat = GCSNCTableSetCellStringPosition(-1, "GNCTable00000", 2, 1, 3, 2);
```

GCSNCTableSetBeforeSubCursorNum	Set the last sub cursor No.

Syntax	GCSNCTableSetBeforeSubCursorNum(nWindowNo, strName, usLine, usRow);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (i)SHORT usLine : line No. (from 0) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2091 : sub cursor is invalid
Details	Sets a designated cell position as the last position before the sub cursor moves. After the position is set, the control area is registered as a redraw area.
Example	Set GNCTable00000 line No.0, row No.0 in the self screen as the last position before sub cursor moves. LONG Stat; Stat = GCSNCTableSetBeforeSubCursorNum(-1, "GNCTable00000", 0, 0);

GCSNCTableGetBeforeSubCursorNum	Get the last sub cursor No.

Syntax

GCSNCTableGetBeforeSubCursorNum(nWindowNo, strName, gmPoint);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen)
(i)STRING strName : control name
(o)GMEM gmPoint : Global memory for line and row No.

Return value

(Long) Setting result
0x0 : Normal
0x2091 : sub cursor is invalid

Details

Gets the last position before the sub cursor moves.
If it is combined with the function that gets the sub cursor position, the movement amount of the sub cursor is obtained.

Example

The last position for GNCTable00000 in the self screen before the sub cursor moves is obtained as:
nLine: line No. of the cell before sub cursor moves
nRow: row No. of the cell before sub cursor moves

```
SHORT nLine;
SHORT nRow;
LONG Stat;
GMEM mem;
mem = GMEMCreate("MATRIX", 4);
Stat = GCSNCTableGetBeforeSubCursorNum(-1,
"GNCTable00000", mem);
nLine = GMEMGetShort(mem, 0);
nRow = GMEMGetShort(mem, 2);
GMEMDelete(mem);
```

GCSNCTableSetLineTitleBackColor	Set background color of line title

Syntax	GCSNCTableSetLineTitleBackColor(nWindowNo, strName, gcColor, usLine);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (i)LONG gcColor : color code for background color(RGB) (i)SHORT usLine : line No. (from 0)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2092 : title hidden mode
Details	Sets background color of the designated line title. After the color is set, the control area is registered as a redraw area.
Example	Set green (0x008000) to the background color of the line title in GNCTable00000 line No.3 in the self screen. LONG Stat; Stat = GCSNCTableSetLineTitleBackColor(-1, "GNCTable00000", H008000, 3);

GCSNCTableSetRowTitleBackColor	Set background color of row title

Syntax	GCSNCTableSetRowTitleBackColor(nWindowNo, strName, gcColor, usRow);
Argument	(i)LONG nWindowNo : screen No.(Specify -1 for self screen) (i)STRING strName : control name (i)LONG gcColor : color code for backgroung color(RGB) (i)SHORT usRow : row No. (from 0)
Return value	(Long) Setting result 0x0 : Normal 0x2001 : cell No. is out of range 0x2092 : title hidden mode
Details	Sets background color of the designated row title. After the color is set, the control area is registered as a redraw area.
Example	Set green (0x008000) to the background color of the row title for GNCTable00000 line No.2 in the self screen. LONG Stat; Stat = GCSNCTableSetRowTitleBackColor(-1, "GNCTable00000", H008000, 2);

GCSNCTableSetSubCursorBackColor	Set background color of sub cursor

Syntax	GCSNCTableSetSubCursorBackColor(nWindowNo, strName, gcColor);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen) (i)STRING strName : control name (i)LONG gcColor : color code for background color(RGB)
Return value	(Long) Setting result 0x0 : Normal
Details	Sets background color of the sub cursor. After the color is set, the control area is registered as a redraw area.
Example	Set green (0x008000) to the background color of the sub cursor in GNCTable00000 in the self screen. LONG Stat; Stat = GCSNCTableSetSubCursorBackColor (-1, "GNCTable00000", H008000);

5.26 InputBox

GCSInputBoxSetPasswordStatus	Set password display status
Syntax	GCSInputBoxSetPasswordStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : password display status
Return value	None
Details	Sets whether the password will be displayed for input boxes or not. One of the following values is set for fStatus. 0: normal display status 1: password display status After the password display status is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 password display status in the self screen to 1. GCSInputBoxSetPasswordStatus(-1, "GInputBox00000", 1) ;

GCSInputBoxGetPasswordStatus	Get password display status
Syntax	GCSInputBoxGetPasswordStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: password display status
Details	Gets whether the input box display status is normal display status or the password display status.
Example	Gets the GInputBox00000 password display status in the self screen. LONG Stat; Stat = GCSInputBoxGetPasswordStatus(-1, "GInputBox00000") ;

GCSInputBoxSetString	Set display character string

Syntax	GCSInputBoxSetString(nWindowNo, strName, pString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (i) STRING pString : display character string
Return value	0: setting failed 1: setting succeeded
Details	Sets the display character string. After the display character string is set, the control area is registered as a redraw area.
Example	Sets the character string "ABCD" to GInputBox00000 in the self screen. GCSInputBoxSetString(-1, "GInputBox00000", "ABCD") ;

GCSInputBoxGetString	Get display character string

Syntax	GCSInputBoxGetString(nWindowNo, strName, pString);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name (o) STRING pString : display character string
Return value	0: acquisition failed 1: acquisition succeeded
Details	Stores the display character string in pString.
Example	Gets the GInputBox00000 display character string in the self screen in strStat. STRING strStat; GCSInputBoxGetString(-1, "GInputBox00000", strStat) ;

GCSInputBoxSetGValue	Set display value

Syntax	GCSInputBoxSetGValue(nWindowNo, strName, gvValue);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)GMEM gvValue : global memory for setting value
Return value	0: setting failed 1: setting succeeded
Details	Sets the value displayed in an input box. Numbers are converted to character strings according to the format defined by the control to reflect the INPUT. If the conversion of numbers to character strings fails, or the converted character strings exceeds the maximum number of characters, or if any control is not specified as the destination to reflect the INPUT, an error is returned in the return value. After the display value is set, the control area is registered as a redraw area.
Example	Sets the signed decimal integer value 100 as a value to be displayed in the GInputBox00000 in the self screen. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSInputBoxSetGValue(-1, "GInputBox00000", mem) ; GMEMDelete(mem);

GCSInputBoxGetGValue	Get display value

Syntax	GCSInputBoxGetGValue(nWindowNo, strName, gvValue);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (o)GMEM gvValue : global memory to numerical value
Return value	0: acquisition failed 1: acquisition succeeded
Details	Gets the value displayed in an input box. Display character strings are converted to numbers according to the character string type defined by the control to reflect the INPUT and stored in the argument. If the value is not displayed or the conversion into numbers fails, or if any control is not specified as the destination to reflect the INPUT, an error is returned in the return value.
Example	Gets the value displayed in GInputBox00000 in the self screen as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSInputBoxGetGValue(-1, "GInputBox00000", mem) ; nVal = GMEMGetLong(mem, 0) ; GMEMDelete(mem);

GCSInputBoxSetFocusEffect	Set effect during focus

Syntax	GCSInputBoxSetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect. 1: No effect 2: With cursor 3: Select After the effect during focus is set, the control area is registered as a redraw area.
Example	Sets the GInputBox00000 effect during focus in the self screen to 1. GCSInputBoxSetFocusEffect(-1, "GInputBox00000", 1) ;

GCSInputBoxGetFocusEffect	Get effect during focus

Syntax	GCSInputBoxGetFocusEffect(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	1: No effect 2: With cursor 3: Select Besides the above: illegal display setting
Details	Gets the effect used during focus.
Example	Gets the GInputBox00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetFocusEffect(-1, "GInputBox00000") ;

GCSInputBoxSetBackGroundPattern	Set background fill status

Syntax	GCSInputBoxSetBackGroundPattern(nWindowNo, strName, nBackGroundPattern);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. One of the following values is set for nBackGroundPattern: -1: no fill -2: with background fill After the background fill status is set, the control area is registered as a redraw area.
Example	Sets the GInputBox00000 background fill status in the self screen to -2. GCSInputBoxSetBackGroundPattern(-1, "GInputBox00000", -2) ;

GCSInputBoxGetBackGroundPattern	Get background fill status

Syntax	GCSInputBoxGetBackGroundPattern(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Background fill status -1: no fill -2: with background fill
Details	Gets the background color fill status.
Example	Gets the GInputBox00000 background fill status in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetBackGroundPattern(-1, "GInputBox00000") ;

GCSInputBoxSetCalcFlag	Set operation function

Syntax	GCSInputBoxSetCalcFlag(nWindowNo, strName, ucCalcFlag);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucCalcFlag : availability of operation function
Return value	None
Details	Specify the availability of operation function. One of the following values is set for ucCalcFlag. 0: Without operation function 1: With operation function
Example	Sets the GInputBox00000 operation function in the self screen to "1". GCSInputBoxSetCalcFlag(-1, "GInputBox00000", 1) ;

GCSInputBoxGetCalcFlag	Get operation function

Syntax	GCSInputBoxGetCalcFlag(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Without operation function 1: With operation function Besides the above: illegal setting
Details	Gets the availability of operation function
Example	Gets the GInputBox00000 operation function availability in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetCalcFlag(-1, "GInputBox00000") ;

GCSInputBoxSetInputValue	Set the control to reflect the INPUT

Syntax	GCSInputBoxSetInputValue(nWindowNo, strName, ucSetType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucSetType : setting method (absolute/incremental)
Return value	0: setting failed 1: setting succeeded
Details	Sets the input contents to the control to reflect the INPUT. The setting method can be selected from absolute or incremental. When the absolute is selected, the value will directly be set. And when the incremental is selected, the value in the control to which the INPUT will be reflected, will be added to the input value. One of the following values is set for ucSetType. 0: Absolute setting 1: Incremental setting After the setting is made, the control area is registered as a redraw area. If any control is not specified as the destination to reflect the INPUT, an error is returned in the return value.
Example	Sets the added value to the GInputBox00000 control specified to reflect the INPUT in the self screen (1). GCSInputBoxSetInputValue(-1, "GInputBox00000", 1) ;

GCSInputBoxSetEchoback	Set echo back

Syntax	GCSInputBoxSetEchoback(nWindowNo, strName, ucEchoback);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucEchoback : availability of echo back
Return value	None
Details	Set whether to echo back or not. One of the following values is set for ucEchoback. 0: Do not echo back 1: Echo back When 1(Echo back) is set and, also, a control is specified as the destination to reflect the INPUT, its content will be reflected and displayed in the input box.
Example	Sets the GInputBox00000 echo back in the self screen to 1. GCSInputBoxSetEchoback(-1, "GInputBox00000", 1) ;

GCSInputBoxGetEchoback	Get echo back

Syntax	GCSInputBoxGetEchoback(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Do not echo back 1: Echo back Besides the above: illegal display setting
Details	Gets whether to echo back or not to echo back.
Example	Gets the availability of echo back in the GInputBox0000 in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetEchoback(-1, "GInputBox00000") ;

GCSInputBoxSetAbsIncType	Set input method (absolute/incremental)

Syntax	GCSInputBoxSetAbsIncType(nWindowNo, strName, nInputType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nInputType : input method (absolute/incremental)
Return value	None
Details	Select the input method (absolute/incremental) One of the following values is set for nInputType. 0: Absolute input 1: Incremental input
Example	Sets GInputBox00000 input method (absolute/incremental) in the self screen to "1". GCSInputBoxSetAbsIncType(-1, "GInputBox00000", 1) ;

GCSInputBoxGetAbsIncType	Get input method (absolute/incremental)

Syntax	GCSInputBoxGetAbsIncType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Absolute input 1: Incremental input Besides the above: illegal display setting
Details	Gets the input method (absolute/incremental)
Example	Gets GInputBox00000 input method (absolute/incremental) in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetAbsIncType(-1, "GInputBox00000") ;

GCSInputBoxSetRefrectControl	Set the control to reflect the INPUT

Syntax

GCSInputBoxSetRefrectControl(nWindowNo, strName, strRefrectControlName);

Argument

(i)LONG nWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)STRING strRefrectControlName	: name of the control to reflect the INPUT

Return value

0: setting failed
1: setting succeeded

Details

Set the control to reflect the INPUT.
One of the following control names is set for strRefrectControlName.
- Text box control
- PLC text box control
- NC data text box control

If a control name not mentioned above is set, an error is returned in the return value.

Example

Sets the text box(GTextBox00001) in the control to reflect the INPUT of GInputBox00000 in the self screen in Stat.

```
GCSInputBoxSetRefrectControl(-1,
"GInputBox00000","GTextBox00001") ;
```

GCSInputBoxGetRefrectControlTextType	Set character string type of control to reflect the INPUT

Syntax	GCSInputBoxGetRefrectControlTextType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Character string type When the control to reflect the INPUT is a "text box". GTEXT_TYPE_STRING(0) : character string GTEXT_TYPE_SHORT(1) : short value GTEXT_TYPE_USHORT(2) : unsigned short value GTEXT_TYPE_LONG(3) : long value GTEXT_TYPE_ULONG(4) : unsigned long value GTEXT_TYPE_FLOAT(5) : float value Besides the above : illegal setting When the control to reflect the INPUT is a "PLC text box". GPLCTEXT_TYPE_DEC(1) : signed decimal integer GPLCTEXT_TYPE_UDEC(2) : unsigned decimal integer GPLCTEXT_TYPE_HEX(3) : hexadecimal integer GPLCTEXT_TYPE_FLOAT(4) : real number (float) GPLCTEXT_TYPE_DOUBLE(5) : real number (double) GPLCTEXT_TYPE_STRING(6) : character sequence Besides the above : illegal setting When the control to reflect the INPUT is a "NC data text box". GNCTEXT_TYPE_STRING(0) : character string GNCTEXT_TYPE_BIN(1) : binary integer GNCTEXT_TYPE_DEC(2) : signed decimal integer GNCTEXT_TYPE_UDEC(3) : unsigned decimal integer GNCTEXT_TYPE_HEX(4) : hexadecimal integer GNCTEXT_TYPE_DOUBLE(5) : real number (double) Besides the above : illegal setting
Details	Gets the character string type (character string/numerical value/real number etc.) of the control to reflect the INPUT.
Example	Gets GInputBox00000 character string type of control to reflect the INPUT in the self screen to Stat. LONG Stat; Stat= GCSInputBoxGetRefrectControlTextType(-1,"GInputBox00000") ;

GCSInputBoxGetInputMode	Get input mode (insert/overwrite)
Syntax	GCSInputBoxGetInputMode(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Insert mode 1: Overwrite mode
Details	Gets the current input mode (insert/overwrite).
Example	Gets the GInputBox00000 current input mode in the self screen in Stat. LONG Stat; Stat = GCSInputBoxGetInputMode(-1, "GInputBox00000") ;
Complement	When the cursor type is "Vertical line", it always gets 0(insert mode).

GCSInputBoxGetSubCursorControl	Get the ID of the sub cursor display control

Syntax

GCSInputBoxGetSubCursorControl (nWindowNo, strName);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name

Return value

-1 : acquisition failed
ID of the control where the : acquisition succeeded sub cursor is pointed

Details

Gets the ID of the control where the sub cursor is pointed at.
When the sub cursor settings are not made, an error (-1) is returned to the return value.

Example

Gets GInputBox00000 ID of the control where the sub cursor is pointed in the self screen.

```
LONG _lCtrlid;
_lCtrlid = GCSInputBoxGetSubCursorControl
(-1,"GInputBox00000") ;
```

Complement

There are two ways to get the control name from the control ID.

(1) Acquisition from the macro editing dialog box
The definition of a control ID will be output to the file created when converted for interpreter execution.
The control ID definition file will be "PNLPG###.GID".
: Three digits indicating the page No. in hexadecimal number.
Recreate the converted "PNLPG###.GID" after adding or deleting a control.

Output example)

```
GINPUTBOX00000,0,GInputBox
GTEXTBOX00001,1,GTextBox
GNCPLCTEXTBOX00004,2,GNCPLCTextBox
GNCDATATEXTBOX00007,3,GNCDataTextBox
GBASICCONTROL00000,4,GBasicControl
```

(2) Acquisition by generating the source
The control ID will be output as the enum definition to the file created when the source is generated for compilation execution.
The file to which the control ID will be output will be a panel/window hpp file. The hpp file name at default will be "GCPanel*****.hpp" for a panel and "GCWindow*****.hpp" for a window.
***** : Five digits indicating the page No. in hexadecimal number.
Regenerate the source code and recreate "GCPanel*****.hpp" or GCWindow*****.hpp" after adding or deleting a control.

Output example)

```
class GCPanel00000 : public GCPanel
{
public:
  //{{CONTROL_ID
  enum {
        GSOFTKEY00000 = 0,
        GTEXTBOX00001 = 1,
        GNCPLCTEXTBOX00004 = 2,
        GNCDATATEXTBOX00007 = 3,
        GBASICCONTROL00000 = 4,
};
//}}CONTROL_ID
```

5.27 SoftKey

GCSSoftKeySetPasswordStatus	Set password display status

Syntax	GCSSoftKeySetPasswordStatus(nWindowNo, strName, fStatus);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG fStatus : password display status
Return value	None
Details	Sets whether the password will be displayed for the ten-key control. One of the following values is set for fStatus. 0: normal display status 1: password display status After the password display status is set, the control area is registered as a redraw area.
Example	Sets the GTextBox00000 password display status in the self screen to "1". GCSSoftKeySetPasswordStatus(-1, "GSoftKey00000", 1) ;

GCSSoftKeyGetPasswordStatus	Get password display status

Syntax	GCSSoftKeyGetPasswordStatus(nWindowNo, strName);
Argument	(i) LONG nWindowNo : screen No. (Specify -1 for self screen.) (i) STRING strName : control name
Return value	0: normal display status 1: password display status
Details	Gets whether the ten-key control display status is normal display status or the password display status.
Example	Gets the GSoftKey00000 password display status in the self screen in Stat. LONG Stat; Stat = GCSSoftKeyGetPasswordStatus(-1, "GSoftKey00000") ;

GCSSoftKeySetString	Set display character string

Syntax	GCSSoftKeySetString(nWindowNo, strName, pString);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING pString : display character string
Return value	0: setting failed 1: setting succeeded
Details	Sets the character string to display in the input data display part. After the display character string is set, the control area is registered as a redraw area.
Example	Sets the character string "ABCD" to GSoftKey00000 input data display part in the self screen. GCSSoftKeySetString(-1, "GSoftKey00000", "ABCD") ;

GCSSoftKeyGetString	Get display character string

Syntax	GCSSoftKeyGetString(nWindowNo, strName, pString);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (o)STRING pString : display character string
Return value	0: acquisition failed 1: acquisition succeeded
Details	Stores the display character string in pString.
Example	Gets the character string displayed in GSoftKey00000 input data display part in the self screen in strStat. STRING strStat; GCSSoftKeyGetString(-1, "GSoftKey00000", strStat) ;

GCSSoftKeySetGValue	Set display value

Syntax	GCSSoftKeySetGValue(nWindowNo, strName, gvValue);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)GMEM gvValue : global memory for setting value
Return value	0: setting failed 1: setting succeeded
Details	Sets the value to be displayed in the input data display part. Numbers are converted to character strings according to the format defined by the control to reflect the INPUT. If the conversion of numbers to character strings fails, or the converted character strings exceeds the maximum number of characters, or if any control is not specified as the destination to reflect the INPUT, an error is returned in the return value. After the display value is set, the control area is registered as a redraw area.
Example	Sets the signed decimal integer value 100 as a value to be displayed in the GSoftKey00000 in the self screen. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSSoftKeySetGValue(-1, "GSoftKey00000", mem) ; GMEMDelete(mem);

GCSSoftKeyGetGValue	Get display value

Syntax	GCSSoftKeyGetGValue(nWindowNo, strName, gvValue);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (o)GMEM gvValue : global memory for numerical value
Return value	0: acquisition failed 1: acquisition succeeded
Details	Gets the value displayed in the input data display part. Displayed character strings are converted to numbers according to the character string type defined by the control to reflect the INPUT and stored in the argument. If the value is not displayed or the conversion into numbers fails, an error is returned in the return value.
Example	Gets the value displayed in GSoftKey00000 in the self screen as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSSoftKeyGetGValue(-1, "GSoftKey00000", mem) ; nVal = GMEMGetLong(mem, 0) ; GMEMDelete(mem);

GCSSoftKeySetFocusEffect	Set effect during focus

Syntax	GCSSoftKeySetFocusEffect(nWindowNo, strName, ucFocusEffect);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect. 1: No effect 2: With cursor Only when the property "Input type" is set to "All key type", the cursor will be displayed at the input data display part. After the effect during focus is set, the control area is registered as a redraw area.
Example	Sets the GSoftKey00000 effect during focus in the self screen to "1". GCSSoftKeySetFocusEffect(-1, "GSoftKey00000", 1) ;

GCSSoftKeyGetFocusEffect	Get effect during focus

Syntax	GCSSoftKeyGetFocusEffect(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	1: No effect 2: With cursor Besides the above: illegal display setting
Details	Gets the setting during focus.
Example	Gets the GSoftKey00000 effect during focus in the self screen in Stat. LONG Stat; Stat = GCSSoftKeyGetFocusEffect(-1, "GSoftKey00000") ;

GCSSoftKeySetCalcFlag	Set operation function

Syntax	GCSSoftKeySetCalcFlag(nWindowNo, strName, ucCalcFlag);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucCalcFlag : availability of operation function
Return value	None
Details	Specify the availability of operation function. One of the following values is set for ucCalcFlag. 0: Without operation function 1: With operation function "Without operation function" and "With operation function" can be switched over only when the property "Input type" is set to "ALL key type". When the operation function is not provided, an operator will be counted as a character. When the "Input type" is set to "Decimal number" or "Hexadecimal", the operation function will always be valid.
Example	Sets the GSoftKey00000 operation function in the self screen to "1". GCSSoftKeySetCalcFlag(-1, "GSoftKey00000", 1) ;

GCSSoftKeyGetCalcFlag	Get operation function

Syntax	GCSSoftKeyGetCalcFlag(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Without operation function 1: With operation function Besides the above: illegal setting
Details	Gets the availability of operation function.
Example	Gets the GSoftKey00000 operation function availability in the self screen in Stat. LONG Stat; Stat = GCSSoftKeyGetCalcFlag(-1, "GSoftKey00000") ;

GCSSoftKeySetInputValue	Set the control to reflect the INPUT

Syntax	GCSSoftKeySetInputValue(nWindowNo, strName, ucSetType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG ucSetType : Setting method (absolute/incremental)
Return value	0: setting failed 1: setting succeeded
Details	Sets the input contents to the control to reflect the INPUT. The setting method can be selected from absolute or incremental. When the absolute is selected, the value will directly be set. And when the incremental is selected, the value in the control to which the INPUT will be reflected, will be added to the input value. One of the following values is set for ucSetType. 0: Absolute setting 1: Incremental setting After the setting is made, the control area is registered as a redraw area. If any control is not specified as the destination to reflect the INPUT, an error is returned in the return value.
Example	Sets the added value to the GInputBox00000 control specified to reflect the INPUT in the self screen (1). GCSSoftKeySetInputValue(-1, "GSoftKey00000", 1) ;

GCSSoftKeySetAbsIncType	Set input method (absolute/incremental)

Syntax	GCSSoftKeySetAbsIncType(nWindowNo, strName, nInputType);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)LONG nInputType : input method (absolute/incremental)
Return value	None
Details	Select the input method (absolute/incremental) When "Inc" is set, the Inc key button on the ten-key control will be highlighted. One of the following values is set for nInputType. 0: Absolute input 1: Incremental input
Example	Sets the input method (absolute/incremental) of GSoftKey00000 in the self screen to "1". GCSSoftKeySetAbsIncType(-1, "GSoftKey00000", 1) ;

GCSSoftKeyGetAbsIncType	Get input method (absolute/incremental)

Syntax	GCSSoftKeyGetAbsIncType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0: Absolute input 1: Incremental input Besides the above: illegal display setting
Details	Gets the input method (absolute/incremental)
Example	Gets GSoftKey00000 input method (absolute/incremental)in the self screen in Stat. LONG Stat; Stat = GCSSoftKeyGetAbsIncType(-1, "GSoftKey00000") ;

GCSSoftKeySetRefrectControl	Set the control to reflect the INPUT

Syntax	GCSSoftKeySetRefrectControl(nWindowNo, strName, strRefrectControlName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING strRefrectControlName : Name of the control to reflect the INPUT
Return value	0: setting failed 1: setting succeeded
Details	Set the control to which the INPUT is to be reflected. One of the following control names is set for strRefrectControlName. - Text box control - PLC text box control - NC data text box control If a control name not mentioned above is set, an error is returned in the return value.
Example	Sets the text box (GTextBox00001) in the control to reflect the INPUT of GInputBox00000 in the self screen in Stat. GCSSoftKeySetRefrectControl(-1, "GSoftKey00000", "GTextBox00001") ;

GCSSoftKeyGetRefrectControlTextType	Set character string type of control to reflect the INPUT

Syntax	GCSSoftKeyGetRefrectControlTextType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Character string type When the control to reflect the INPUT is a "text box". GTEXT_TYPE_STRING(0) : character string GTEXT_TYPE_SHORT(1) : short value GTEXT_TYPE_USHORT(2) : unsigned short value GTEXT_TYPE_LONG(3) : long value GTEXT_TYPE_ULONG(4) : unsigned long value GTEXT_TYPE_FLOAT(5) : float value Besides the above : illegal setting When the control to reflect the INPUT is a "PLC text box". GPLCTEXT_TYPE_DEC(1) : signed decimal integer GPLCTEXT_TYPE_UDEC(2) : unsigned decimal integer GPLCTEXT_TYPE_HEX(3) : hexadecimal integer GPLCTEXT_TYPE_FLOAT(4) : real number Besides the above : illegal setting When the control to reflect the INPUT is a "NC data text box". GNCTEXT_TYPE_STRING(0) : character string GNCTEXT_TYPE_BIN(1) : binary integer GNCTEXT_TYPE_DEC(2) : signed decimal integer GNCTEXT_TYPE_UDEC(3) : unsigned decimal integer GNCTEXT_TYPE_HEX(4) : hexadecimal integer GNCTEXT_TYPE_DOUBLE(5) : real number (double) Besides the above : illegal setting
Details	Gets the character string type (character string/numerical value/real number etc.) of the control to reflect the INPUT.
Example	Gets GSoftKey00000 character string type of control to reflect the INPUT in the self screen to Stat. LONG Stat; Stat= GCSSoftKeyGetRefrectControlTextType(-1," GSoftKey00000") ;

GCSSoftKeyGetInputMode	Get input mode (insert/overwrite)
Syntax	GCSSoftKeyGetInputMode(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	0:Insert mode 1:Overwrite modeGets the current input mode (insert/overwrite).
Details	Gets the current input mode (insert/overwrite).
Example	Gets the GSoftKey00000 current input mode in the self screen in Stat. LONG Stat; Stat = GCSSoftKeyGetInputMode(-1, "GSoftKey00000") ;
Complement	When the cursor type is "Vertical line", it always gets 0(insert mode).

GCSSoftKeySetTitleString	Set title bar display character string
Syntax	GCSSoftKeySetTitleString(nWindowNo, strName, pTitleString);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING pTitleString : title bar display character string
Return value	None
Details	Gets the character string to be displayed on the title bar. When the title name exceeds the maximum number of characters, the title name will overlap with the movement mark △. Make sure that the title name does not exceed the maximum number of characters. After the setting is made, the control area is registered as a redraw area.
Example	Sets "ABCD" to the GSoftKey00000 title bar in the self screen. GCSSoftKeySetTitleString(-1, "GSoftKey00000", "ABCD") ;

GCSSoftKeyGetSubCursorControl	Get the ID of the sub cursor display control

Syntax

GCSSoftKeyGetSubCursorControl (nWindowNo, strName);

Argument

(i)LONG nWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name

Return value

-1 : acquisition failed
ID of the control where the sub cursor is pointed : acquisition succeeded

Details

Gets the ID of the control where the sub cursor is pointed at.
When the sub cursor settings are not made, an error (-1) is returned to the return value.

Example

Gets GSoftKey00000 of the control where the sub cursor is pointed in the self screen.

```
LONG _lCtrlid;
_lCtrlid = GCSSoftKeyGetSubCursorControl (-1,"
GSoftKey00000") ;
```

Complement

There are two ways to get the control name from the control ID.

(1) Acquisition from the macro editing dialog box
The definition of a control ID will be output to the file created when converted for interpreter execution.
The control ID definition file will be "PNLPG###.GID".
: Three digits indicating the page No. in hexadecimal number.
Recreate the converted "PNLPG###.GID" after adding or deleting a control.

Output example)

```
GSOFTKEY00000,0,GSoftKey
GTEXTBOX00001,1,GTextBox
GNCPLCTEXTBOX00004,2,GNCPLCTextBox
GNCDATATEXTBOX00007,3,GNCDataTextBox
GBASICCONTROL00000,4,GBasicControl
```

(2) Acquisition by generating the source
The control ID will be output as the enum definition to the file created when the source is generated for compilation execution.
The file to which the control ID will be output will be a panel/window hpp file. The hpp file name at default will be "GCPanel*****.hpp" for a panel and "GCWindow*****.hpp" for a window.
***** : Five digits indicating the page No. in hexadecimal number.
Regenerate the source code and recreate "GCPanel*****.hpp" or GCWindow*****.hpp" after adding or deleting a control.

Output example)

```
class GCPanel00000 : public GCPanel
{
public:
  //{{CONTROL_ID
  enum {
        GSOFTKEY00000 = 0,
        GTEXTBOX00001 = 1,
        GNCPLCTEXTBOX00004 = 2,
        GNCDATATEXTBOX00007 = 3,
        GBASICCONTROL00000 = 4,
};
//}}CONTROL_ID
```

GCSNCPageChgBtnSetDestinationPageID	Set destination screen number

Syntax	GCSNCPageChgBtnSetDestinationPageID(nWindowNo, strName, wDstPageID);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)UNSIGNED SHORT wDstPageID : Destination screen No.
Return value	FALSE : Abnormal TRUE : Normal
Details	Sets the destination screen number.
Example	Sets the destination screen number of GNCPageChangeButton00000 on self screen to 7000. GCSNCPageChgBtnSetDestinationPageID (-1,"GNCPageChangeButton00000",7000);

GCSNCPageChgBtnGetDestinationPageID	Get destination screen number

Syntax	GCSNCPageChgBtnGetDestinationPageID(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	0-9999: Destination screen No.
Details	Gets the destination screen number.
Example	Gets the destination screen number of GNCPageChangeButton00000 on self screen to wStat. UNSIGNED SHORT wStat; wStat = GCSNCPageChgBtnGetDestinationPageID (-1, "GNCPageChangeButton00000");

GCSNCPageChgBtnSetFocusMoveToWindow	Set focus move to window

Syntax	GCSNCPageChgBtnSetFocusMoveToWindow(nWindowNo, strName, wFocusMove);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)UNSIGNED SHORT wFocusMove : Focus move to window
Return value	FALSE : Abnormal TRUE : Normal
Details	Sets focus move to the target window. For wFocusMove, specify any of the following values. 0 : Perform focus move 1 : Do not perform focus move
Example	Sets focus move for the GNCPageChangeButton00000 window in the self screen to [1: Do not perform focus move]. GCSNCPageChgBtnSetFocusMoveToWindow (-1, "GNCPageChangeButton00000", 1);

GCSNCPageChgBtnGetFocusMoveToWindow	Get focus move to window

Syntax	GCSNCPageChgBtnGetFocusMoveToWindow (nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	0 : Perform focus move 1 : Do not perform focus move
Details	Gets focus move to the target window.
Example	Gets focus move to the GNCPageChangeButton00000 window on the self screen to wStat. UNSIGNED SHORT wStat; wStat = GCSNCPageChgBtnGetFocusMoveToWindow (-1, "GNCPageChangeButton00000");

GCSNCPageChgBtnSetCloseWindow	Set close of source window

Syntax	GCSNCPageChgBtnSetCloseWindow(nWindowNo, strName, wCloseWindow);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)UNSIGNED SHORT wCloseWindow : Close of the source window
Return value	FALSE(0) : Abnormal TRUE(1) : Normal
Details	Sets close of the source window. For wCloseWindow, specify any of the following values. 0 : Do not close the source window 1 : Closes the source window
Example	Sets close of the GNCPageChangeButton00000 window on the self screen to "1" (Close). GCSNCPageChgBtnSetCloseWindow(-1,"GNCPageChangeButton00000", 1);

GCSNCPageChgBtnGetCloseWindow	Get close of source window

Syntax	GCSNCPageChgBtnGetCloseWindow(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	0 : Do not close the source window 1 : Closes the source window
Details	Gets close of the source window.
Example	Gets close of the GNCPageChangeButton00000 window on the self screen to wStat. UNSIGNED SHORT wStat; nStat = GCSNCPageChgBtnGetCloseWindow (-1, "GNCPageChangeButton00000");

GCSNCPageChgBtnSetInputKeyID	Set input key ID

Syntax

GCSNCPageChgBtnSetInputKeyID(nWindowNo, strName, wInputKeyID);

Argument

(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)UNSIGNED SHORT wInputKeyID : Input key ID

Return value

FALSE(0) : Abnormal
TRUE(1) : Normal

Details

Sets the input key ID.
For wCloseWindow, specify any of the following values.
0: No setting
1: Switches the part system.
2 :SFP
3 :F0
4: Menu list
5: Help
6: Displays the window.
7: Switches the window.
8: Menu 1
9: Menu 2
11: Menu 3
12: Menu 4
13: Menu 5
14: Menu 6
15: Menu 7
16: Menu 8
17: Menu 9
18: Menu 10
19: Switches the menu to the left.
20: Switches the menu to the right.
21：MONITOR
22：SETUP
23：EDIT
24：DIAGN
25：MAINTE

Example

Sets the input key ID of GNCPageChangeButton00000 on the self screen to "no setting".

```
GCSNCPageChgBtnSetInputKeyID(-1,
"GNCPageChangeButton00000", 0);
```

GCSNCPageChgBtnGetInputKeyID	Get input key ID

Syntax

GCSNCPageChgBtnGetInputKeyID (nWindowNo, strName);

Argument

(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name

Return value

Input key ID
0: No setting
1: Switches the part system.
2 :SFP
3 :F0
4: Menu list
5: Help
6: Displays the window.
7: Switches the window.
8: Menu 1
9: Menu 2
11: Menu 3
12: Menu 4
13: Menu 5
14: Menu 6
15: Menu 7
16: Menu 8
17: Menu 9
18: Menu 10
19: Switches the menu to the left.
20: Switches the menu to the right.
21：MONITOR
22：SETUP
23：EDIT
24：DIAGN
25：MAINTE

Details

Gets the input key ID setting.

Example

Gets the input key ID of GNCPageChangeButton00000 on the self screen to wStat.

```
UNSIGNED SHORT wStat;
wStat = GCSNCPageChgBtnGetInputKeyID
(-1, "GNCPageChangeButton00000");
```

5.28 Graph

GCSNCGraphSetChartType	Set chart type
Syntax	GCSNCGraphSetChartType(nWindowNo, strName, nChartType);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nChartType : Chart type (0: Pie chart, 1: Band chart)
Return value	None
Details	Sets the chart type.
Example	Sets the chart type of GNCGraph00000 on self screen to 0: Pie chart. GCSNCGraphSetChartType(-1, "GNCGraph00000", 0);

GCSNCGraphGetChartType	Get chart type
Syntax	GCSNCGraphGetChartType(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Chart type (0: Pie chart, 1: Band chart)
Details	Gets the chart type.
Example	Gets the chart type of GNCGraph00000 on self screen. SHORT nChartType; nChartType = GCSNCGraphGetChartType(-1, "GNCGraph00000");

GCSNCGraphSetDispDirection	Set display direction

Syntax	GCSNCGraphSetDispDirection(nWindowNo, strName, nDirection);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nDirection : Display direction (0:Vertical, 1:Horizontal)
Return value	None
Details	Sets the display direction (vertical or horizontal) of the band chart.
Example	Sets the display direction of GNCGraph00000 on self screen to 1: Landscape. GCSNCGraphSetDispDirection(-1, "GNCGraph00000", 1);

GCSNCGraphGetDispDirection	Get display direction

Syntax	GCSNCGraphGetDispDirection(nWindowNo, strName, ItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Display direction (0:Vertical, 1:Horizontal)
Details	Sets the display direction (vertical or horizontal) of the band chart.
Example	Gets the display direction of GNCGraph00000 on self screen. SHORT nDirection; nDirection =GCSNCGraphGetDispDirection(-1, "GNCGraph00000")

GCSNCGraphSetItemNum	Set number of items

Syntax	GCSNCGraphSetItemNum(nWindowNo, strName, ItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG ItemNum : Number of items (1 to 32)
Return value	None
Details	Sets the number of items to be displayed in the chart.
Example	Sets the number of items for GNCGraph00000 on self screen to 6. GCSNCGraphSetItemNum(-1, "GNCGraph00000", 6);

GCSNCGraphGetItemNum	Get number of items

Syntax	GCSNCGraphGetItemNum(nWindowNo, strName, ItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Number of chart items
Details	Gets the number of items to be displayed in the chart.
Example	Gets the number of items for GNCGraph00000 on self screen. LONG ItemNum; ItemNum = GCSNCGraphGetItemNum(-1, "GNCGraph00000");

GCSNCGraphSetSection	Set section number

Syntax	GCSNCGraphSetSection(nWindowNo, strName, lSection);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lSection : Section number
Return value	None
Details	Sets the section number.
Example	Sets the section number of GNCGraph00000 on self screen to 55. GCSNCGraphSetSection(-1, "GNCGraph00000", 55);

GCSNCGraphGetSection	Get section number

Syntax	GCSNCGraphGetSection(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Section number
Details	Gets the section number.
Example	Gets the section number of GNCGraph00000 on self screen. LONG lSubSection; lSubSection = GCSNCGraphGetSection(-1, "GNCGraph00000");

GCSNCGraphSetSubSection	Set sub-section number

Syntax	GCSNCGraphSetSubSection(nWindowNo, strName, lSubSection);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lSubSection : Sub-section number
Return value	None
Details	Sets the sub-section number.
Example	Sets the sub-section number of GNCGraph00000 on self screen to 90000. GCSNCGraphSetSubSection (-1, "GNCGraph00000", 90000);

GCSNCGraphGetSubSection	Get sub-section number

Syntax	GCSNCGraphGetSubSection(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Sub-section number
Details	Gets the sub-section number.
Example	Gets the sub-section number of GNCGraph00000 on self screen. LONG lSubSection; lSubSection = GCSNCGraphGetSubSection (-1, "GNCGraph00000");

GCSNCGraphSetGround	Set ground

Syntax	GCSNCGraphSetGround(nWindowNo, strName, lGround);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lGround : Ground
Return value	None
Details	Sets the ground.
Example	Sets the ground of GNCGraph00000 on self screen to 0. GCSNCGraphSetGround(-1, "GNCGraph00000", 0);

GCSNCGraphGetGround	Get ground

Syntax	GCSNCGraphGetGround(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Ground
Details	Gets the ground.
Example	Gets the ground of GNCGraph00000 on self screen. LONG lGround; lGround = GCSNCGraphGetGround(-1, "GNCGraph00000");

GCSNCGraphSetAxisInfo	Set axis number

Syntax GCSNCGraphSetAxisInfo(nWindowNo, strName, lAxis);

Argument (i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)LONG lAxis : Axis number

Return value None

Details Sets the axis number.

Example Sets the axis number of GNCGraph00000 on self screen to 2.

```
GCSNCGraphSetAxisInfo(-1, "GNCGraph00000", 2);
```

GCSNCGraphGetAxisInfo	Get axis number

Syntax GCSNCGraphGetAxisInfo(nWindowNo, strName);

Argument (i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name

Return value Axis number

Details Gets the axis number.

Example Gets the axis number of GNCGraph00000 on self screen.

```
LONG lAxis;
lAxis = GCSNCGraphGetAxisInfo(-1, "GNCGraph00000");
```

GCSNCGraphSetDataType	Set data type

Syntax GCSNCGraphSetDataType(nWindowNo, strName, lDataType);

Argument (i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name
(i)LONG lDataType : Data type

Return value None

Details Sets the data type of NC data.
For lDataType, specify any of the following values.
GNCGRAPH_DTYPE_CHAR(1) : char type
GNCGRAPH_DTYPE_SHORT(2) : short value
GNCGRAPH_DTYPE_LONG(3) : long value
GNCGRAPH_DTYPE_DOUBLE(5) : double value
GNCGRAPH_DTYPE_UCHAR(33) : unsigned char value
GNCGRAPH_DTYPE_USHORT(34) : unsigned short value
GNCGRAPH_DTYPE_ULONG(35) : unsigned long value

Example Sets the data type of GNCGraph00000 on self screen.

```
GCSNCGraphSetDataType(-1, "GNCGraph00000", 2);
```

GCSNCGraphGetDataType	Get data type

Syntax GCSNCGraphGetDataType(nWindowNo, strName);

Argument (i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.)
(i)STRING strName : Control name

Return value GNCGRAPH_DTYPE_CHAR(1) : char type
GNCGRAPH_DTYPE_SHORT(2) : short value
GNCGRAPH_DTYPE_LONG(3) : long value
GNCGRAPH_DTYPE_DOUBLE(5) : double value
GNCGRAPH_DTYPE_UCHAR(33) : unsigned char value
GNCGRAPH_DTYPE_USHORT(34) : unsigned short value
GNCGRAPH_DTYPE_ULONG(35) : unsigned long value

Details Gets the data type of NC data.

Example Gets the data type of GNCGraph00000 on self screen.

```
LONG DataType;
DataType = GCSNCGraphGetDataType(-1, "GNCGraph00000");
```

GCSNCGraphSetReadMethod	Set reading method

Syntax	GCSNCGraphSetReadMethod(nWindowNo, strName, nReadMethod);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nReadMethod : Reading method
Return value	None
Details	Sets the reading method of NC data.
Example	Sets 1 to the reading method of GNCGraph00000 on self screen. GCSNCGraphSetReadMethod(-1, "GNCGraph00000", 1);

GCSNCGraphGetReadMethod	Get reading method

Syntax	GCSNCGraphGetReadMethod(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Reading method
Details	Sets the reading method of NC data.
Example	Gets the reading method of GNCGraph00000 on self screen. SHORT nReadMethod; nReadMethod = GCSNCGraphGetReadMethod(-1, "GNCGraph00000");

GCSNCGraphSetReadOffset	Set read offset

Syntax	GCSNCGraphSetReadOffset(nWindowNo, strName, lReadOffset);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lReadOffset : Read offset
Return value	None
Details	Sets the read offset of NC data.
Example	Sets the read offset of GNCGraph00000 on self screen to 100. GCSNCGraphSetReadOffset(-1, "GNCGraph00000", 100);

GCSNCGraphGetReadOffset	Get read offset

Syntax	GCSNCGraphGetReadOffset(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Read offset
Details	Gets the read offset of NC data.
Example	Gets the read offset of GNCGraph00000 on self screen. LONG ReadOffset; ReadOffset = GCSNCGraphGetReadOffset(-1, "GNCGraph00000");

GCSNCGraphSetGraphFillPattern	Set pattern

Syntax	GCSNCGraphSetGraphFillPattern(nWindowNo, strName, nItemNum, nFillPattern);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32) (i)SHORT nFillPattern : Pattern
Return value	None
Details	Sets the background pattern (item specification).
Example	Sets the pattern of item 2 of GNCGraph00000 on self screen to 7. GCSNCGraphSetGraphFillPattern(-1, "GNCGraph00000", 2, 7);

GCSNCGraphGetGraphFillPattern	Get pattern

Syntax	GCSNCGraphGetGraphFillPattern(nWindowNo, strName, nItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32)
Return value	Pattern
Details	Gets the background pattern (item specification).
Example	Gets the pattern of item 3 of GNCGraph00000 on self screen. SHORT nFillPattern; GCSNCGraphGetGraphFillPattern(-1, "GNCGraph00000", 3, nFillPattern);

GCSNCGraphSetGraphForeColor	Set foreground color

Syntax	GCSNCGraphSetGraphForeColor(nWindowNo, strName, nItemNum, lForeColor);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32) (i)LONG lForeColor : Foreground color
Return value	None
Details	Sets the foreground color (item specification).
Example	Sets the foreground color of item 4 of GNCGraph00000 on self screen to white. GCSNCGraphSetGraphForeColor(-1, "GNCGraph00000", 4, HFFFFFF);

GCSNCGraphGetGraphForeColor	Get foreground color

Syntax	GCSNCGraphGetGraphForeColor(nWindowNo, strName, nItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32)
Return value	Foreground color
Details	Gets the foreground color (item specification).
Example	Gets the foreground color of item 3 of GNCGraph00000 on self screen. LOING ForeColor; GCSNCGraphGetGraphForeColor(-1, "GNCGraph00000", 3, ForeColor);

GCSNCGraphSetGraphBackColor	Set background color

Syntax	GCSNCGraphSetGraphBackColor(nWindowNo, strName, nItemNum, lBackColor);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32) (i)LONG lBackColor : Background color
Return value	None
Details	Sets the background color (item specification).
Example	Sets the background color of item 4 of GNCGraph00000 on self screen to white. GCSNCGraphSetGraphBackColor(-1, "GNCGraph00000", 4, HFFFFFF);

GCSNCGraphGetGraphBackColor	Get background color

Syntax	GCSNCGraphGetGraphBackColor(nWindowNo, strName, nItemNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32)
Return value	Background color
Details	Gets the background color (item specification).
Example	Gets the background color of item 3 of GNCGraph00000 on self screen. LOING BackColor; GCSNCGraphGetGraphBackColor(-1, "GNCGraph00000", 3, BackColor);

GCSNCGraphSetScaleColor	Set scale color

Syntax	GCSNCGraphSetScaleColor(nWindowNo, strName, lScaleColor);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lScaleColor : Scale color
Return value	None
Details	Sets the scale color.
Example	Sets the scale of GNCGraph00000 on self screen to white. GCSNCGraphSetScaleColor(-1, "GNCGraph00000", HFFFFFF);

GCSNCGraphGetScaleColor	Get scale color

Syntax	GCSNCGraphGetScaleColor(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Scale color
Details	Gets the scale color.
Example	Gets the scale color of GNCGraph00000 on self screen. LOING lScaleColor; lScaleColor = GCSNCGraphGetScaleColor(-1, "GNCGraph00000");

GCSNCGraphSetScaleNum	Set scale number

Syntax	GCSNCGraphSetScaleNum(nWindowNo, strName, lScaleNum);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lScaleNum : Scale number
Return value	None
Details	Sets the scale number.
Example	Sets the scale number of GNCGraph00000 on self screen to 101. GCSNCGraphSetScaleNum(-1, "GNCGraph00000", 101);

GCSNCGraphGetScaleNum	Get scale number

Syntax	GCSNCGraphGetScaleNum(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Scale number
Details	Gets the scale number.
Example	Gets the scale number of GNCGraph00000 on self screen. LOING lScaleNum; lScaleNum = GCSNCGraphGetScaleNum(-1, "GNCGraph00000");

GCSNCGraphSetScaleWidth	Set scale width

Syntax	GCSNCGraphSetScaleWidth(nWindowNo, strName, nScaleWidth);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nScaleWidth : Scale width
Return value	None
Details	Sets the width of the scale.
Example	Sets the scale width of GNCGraph00000 on self screen to 101. GCSNCGraphSetScaleWidth(-1, "GNCGraph00000", 101);

GCSNCGraphGetScaleWidth	Get scale width

Syntax	GCSNCGraphGetScaleWidth(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Scale width
Details	Gets the scale width.
Example	Gets the scale width of GNCGraph00000 on self screen. SHORT nScaleWidth; nScaleWidth = GCSNCGraphGetScaleWidth(-1, "GNCGraph00000");

GCSNCGraphSetScaleMin	Set scale (minimum)

Syntax	GCSNCGraphSetScaleMin(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)DOUBLE dValue : Scale (minimum)
Return value	None
Details	Sets the scale (minimum) of the chart.
Example	Sets the Scale Min of GNCGraph00000 on self screen to -200.0. GCSNCGraphSetScaleMin(-1, "GNCGraph00000", -200.0);

GCSNCGraphGetScaleMin	Get scale (minimum)

Syntax	GCSNCGraphGetScaleMin(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (o)DOUBLE dValue : Scale (minimum)
Return value	None
Details	Gets the scale (minimum) of the chart.
Example	Gets the scale (minimum) of GNCGraph00000 on self screen. DOUBLE dValue; GCSNCGraphGetScaleMin(-1, "GNCGraph00000", dValue);

GCSNCGraphSetScaleMax	Set scale (maximum)

Syntax	GCSNCGraphSetScaleMax(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)DOUBLE dValue : Scale (maximum)
Return value	None
Details	Sets the scale (maximum) of the chart.
Example	Sets the Scale Max of GNCGraph00000 on self screen to 201.001. GCSNCGraphSetScaleMax(-1, "GNCGraph00000", 201.001);

GCSNCGraphGetScaleMax	Get scale (maximum)

Syntax	GCSNCGraphGetScaleMax(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (o)DOUBLE dValue : Scale (maximum)
Return value	None
Details	Gets the scale (maximum) of the chart.
Example	Gets the scale (maximum) of GNCGraph00000 on self screen. DOUBLE dValue; GCSNCGraphGetScaleMax(-1, "GNCGraph00000", dValue);

GCSNCGraphSetLowerLimit	Set boundary value (lower limit)

Syntax	GCSNCGraphSetLowerLimit(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)DOUBLE dValue : Boundary value (lower limit)
Return value	None
Details	Sets the boundary value (lower limit).
Example	Sets the boundary value (lower limit) of GNCGraph00000 on self screen to -1000.123. GCSNCGraphSetLowerLimit(-1, "GNCGraph00000", -1000.123);

GCSNCGraphGetLowerLimit	Get boundary value (lower limit)

Syntax	GCSNCGraphGetLowerLimit(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (o)DOUBLE dValue : Boundary value (lower limit)
Return value	None
Details	Gets the boundary value (lower limit).
Example	Gets the boundary value (lower limit) of GNCGraph00000 on self screen. DOUBLE dValue; GCSNCGraphGetLowerLimit(-1, "GNCGraph00000", dValue);

GCSNCGraphSetUpperLimit	Set boundary value (upper limit)

Syntax	GCSNCGraphSetUpperLimit(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)DOUBLE dValue : Boundary value (upper limit)
Return value	None
Details	Sets the boundary value (upper limit).
Example	Sets the boundary value (upper limit) of GNCGraph00000 on self screen to 1000.001. GCSNCGraphSetUpperLimit(-1, "GNCGraph00000", 1000.001);

GCSNCGraphGetUpperLimit	Get boundary value (upper limit)

Syntax	GCSNCGraphGetUpperLimit(nWindowNo, strName, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (o)DOUBLE dValue : Boundary value (upper limit)
Return value	None
Details	Gets the boundary value (upper limit).
Example	Gets the boundary value (upper limit) of GNCGraph00000 on self screen. DOUBLE dValue; GCSNCGraphGetUpperLimit(-1, "GNCGraph00000", dValue);

GCSNCGraphSetSort	Set sort type

Syntax	GCSNCGraphSetSort(nWindowNo, strName, nSort);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nSort : Sort type (0: Specified order, 1: Ascending order, 2: Descending order)
Return value	None
Details	Sets the chart (item) display order.
Example	Sets the short type of GNCGraph00000 on original screen to "1:Ascending order". GCSNCGraphSetSort(-1, "GNCGraph00000", 1);

GCSNCGraphGetSort	Get sort type

Syntax	GCSNCGraphGetSort(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Sort type
Details	Gets the sort type.
Example	Gets the sort type of GNCGraph00000 on self screen. SHORT nSort; nSort = GCSNCGraphGetSort(-1, "GNCGraph00000");

GCSNCGraphGetItemData	Get chart item data (reading order)

Syntax	GCSNCGraphGetItemData(nWindowNo, strName, nItemNum, dValue);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)SHORT nItemNum : Item No. (1 to 32) (o)DOUBLE dValue : Read data of the specified chart item
Return value	Result 0 : Abnormal 1 : Normal
Details	Specifies the chart item, and gets data read from NC. (Item numbers are specified in the reading order.)
Example	Gets data of graph item 32 of GNCGraph00000 on self screen. DOUBLE dValue; GCSNCGraphGetItemData(-1, "GNCGraph00000", 32, dValue);

GCSNCGraphSetSystemNumber	Set part system number

Syntax	GCSNCGraphSetSystemNumber(nWindowNo, strName, lSystemNumber);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name (i)LONG lSystemNumber : Part system number
Return value	None
Details	Sets the part system number.
Example	Sets the part system number of GNCGraph00000 on self screen to 1. GCSNCGraphSetSystemNumber(-1, "GNCGraph00000", 1);

GCSNCGraphGetSystemNumber	Get part system number

Syntax	GCSNCGraphGetSystemNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : Screen No. (Specify "-1" for self screen.) (i)STRING strName : Control name
Return value	Part system number
Details	Gets the part system number.
Example	Gets the part system number of GNCGraph00000 on self screen. LONG lSystemNumber; lSystemNumber = GCSNCGraphGetSystemNumber(-1, "GNCGraph00000");

5.29 AlarmList

GCSNCAlarmListUpdateHistory	Update alarm history
Syntax	GCSNCAlarmListUpdateHistory(nWindowNo, strName);
Argument	(i)LONG nWindowNo: screen No. (Specify -1 for self screen.) (i)STRING strName: control name
Return value	None
Details	Updates the alarm history.
Example	Updates the GNCAlarmList00000 alarm history in the self screen. GCSNCAlarmListUpdateHistory(-1, "GNCAlarmList00000");

GCSNCAlarmListSetDispType	Set display type

Syntax	GCSNCAlarmListSetDispType(nWindowNo, strName, usType);
Argument	(i)LONG nWindowNo: screen No. (Specify -1 for self screen.) (i)STRING strName: control name (i)SHORT usType: display type
Return value	None
Details	Sets the alarm type to be displayed in the alarm list. 0: NC message 1: Stop code 2: Alarm message 3: Operator message After the alarm type is set, the control area is registered as a redraw area if the display target is alarm list.
Example	Sets NC message to the GNCAlarmList00000 display type in the self screen. GCSNCAlarmListSetDispType(-1, "GNCAlarmList00000", 0);

GCSNCAlarmListGetDispType	Get display type

Syntax	GCSNCAlarmListGetDispType(nWindowNo, strName);
Argument	(i)LONG nWindowNo: screen No. (Specify -1 for self screen.) (i)STRING strName: control name
Return value	Display type
Details	Gets the alarm type to be displayed in the alarm list. 0: NC message 1: Stop code 2: Alarm message 3: Operator message
Example	Gets the GNCAlarmList00000 display type in the self screen. SHORT usType; usType = GCSNCAlarmListGetDispType(-1, "GNCAlarmList00000");

GCSNCAlarmListGetSelectLine	Get select line number

Syntax	GCSNCAlarmListGetSelectLine(nWindowNo, strName);
Argument	(i)LONG nWindowNo: Screen No. (Specify "-1" for self screen.) (i)STRING strName: Control name
Return value	Select line number
Details	Gets the select line number in message area. If it is unspecified or out of range, 0 is returned. If page switch is executed with PageUp key or PageDown key after selecting a line, 1 (top line) is returned.
Example	Gets the GNCAlarmList00000 select line number in the self screen. SHORT nLine; nLine = GCSNCAlarmListGetSelectLine(-1, "GNCAlarmList00000");

GCSNCAlarmListGetAlarmInfo	Get alarm information

Syntax

GCSNCAlarmListGetAlarmInfo(nWindowNo, strName, lSelectLine, lAlarmInfo, Data);

Argument

(i)LONG nWindowNo: Screen No. (Specify "-1" for self screen.)
(i)STRING strName: Control name
(i)LONG lSelectLine: Specify the line number of alarm information to be acquired
(i)LONG lAlarmInfo: Specify the item of alarm information to be acquired
(o)Variable-type Data: Acquired data

Return value

0: Acquisition failed
1: Acquisition succeeded

Details

Gets the alarm information currently displayed in alarm list or alarm history.
For the item of alarm information to be acquired, set the value of the specified bit ON in the table below, and specify the STRING type as the Variable-type Data.
To acquire multiple items at the same time, combine the specified bit with a logical sum and set it. In this case, specify the variable name of the character string array with the elements corresponding to the number of items to be acquired in the Variable-type Data.

<Alarm list>

Specified bit	NC message/ Stop code	Alarm message/ Operator message
0	Alarm class	Alarm class
1	Alarm message	Alarm message
2	Parameter 1	Parameter
3	Parameter 2	-
4	Part system	-

<Alarm history>

Specified bit	NC message	Alarm message
0	Date	Date
1	Time	Time
2	Alarm class	Alarm class
3	Alarm message	Alarm message
4	Alarm data 1	Message number
5	Alarm data 2	Classification number
6	Part system	-

Example

Gets the alarm information currently displayed in GNCAlarmList00000 alarm list (first row of NC message) in the self screen.

<Acquisition example of single item>
Gets the alarm information (Alarm class).

```
STRING strAlarm;
LONG lResult;
lResult = GCSNCAlarmListGetAlarmInfo(-1,
"GNCAlarmList00000", 1,    H1, strAlarm);
```

(Example) For "P277 macro alarm message 100", "P277" is returned to strAlarm.

<Acquisition example of several items>
Gets the alarm information (Alarm class and Parameter 1).

```
STRING strInfo(2);
LONG lResult;
lResult = GCSNCAlarmListGetAlarmInfo(-1,
"GNCAlarmList00000", 1,    H5, strInfo);
```

(Example) For "P277 macro alarm message 100", "P277" is returned to strInfo(0) and "100" is returned to strInfo(1).

5.30 Extension Menu

GCSNCExMenuSetFontID	Set font resource ID
Syntax	GCSNCExMenuSetFontID (lWindowNo, strName, usID);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usID : font resource ID
Return value	None
Details	Sets the font resource ID.
Example	Sets the GNCExMenu00000 font resource ID in the self screen to 1. GCSNCExMenuSetFontID(-1, "GNCExMenu00000", 1);

GCSNCExMenuGetFontID	Get font resource ID
Syntax	GCSNCExMenuGetFontID (lWindowNo, strName);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Font resource ID
Details	Gets the font resource ID.
Example	Gets the GNCExMenu00000 font resource ID in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetFontID(-1, "GNCExMenu00000");

GCSNCExMenuGetMenuNumFromPoint	Get menu No. from mouse press position

Syntax

GCSNCExMenuGetMenuNumFromPoint (lWindowNo, strName, usPointX, usPointY);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usPointX : X coordinate of pointing device
(i)SHORT usPointY : Y coordinate of pointing device

Return value

Button No.
Menu button No. : 1 to 10
Menu return button : 11
Menu switching button : 12
No press button, or "display status" property is set to other than "full display" : 0

Details

Gets the menu No. from the mouse press position.
The upper left of the control is the origin (0, 0) of the pointing device coordinates.

Example

Gets the menu No. from the mouse press position in GNCExMenu00000 in the self screen in Stat.

```
SHORT usX;                'Coordinate X
SHORT usY;                'Coordinate Y
SHORT Stat;               'Button No.

'Gets coordinate
usX = LUPARAM & H0000FFFF;              'Get coordinate X
usY = (LUPARAM & HFFFF0000) >> 16;      'Get coordinate Y

Stat = GCSNCExMenuGetMenuNumFromPoint(-1,
"GNCExMenu00000", usX, usY);
```

GCSNCExMenuGetDispMenuGroup	Get menu group currently displayed

Syntax	GCSNCExMenuGetDispMenuGroup (lWindowNo, strName);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Menu group No. (1 to 10)
Details	Gets the menu group No. currently displayed.
Example	Gets the menu group No. currently displayed in GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetDispMenuGroup(-1, "GNCExMenu00000");

GCSNCExMenuSetDisplay	Set display type of menu

Syntax	GCSNCExMenuSetDisplay (lWindowNo, strName, usMGNum, usMBNum, ucDType);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)CHAR ucDType : display type
Return value	None
Details	Sets the display type of menu. Sets ucDType to one of the following values. 0: Square 1: Image When a setting outside the range is made, the the setting value is not changed..
Example	Sets the display type of the 1st menu button of GNCExMenu00000 in the self screen to "Square". GCSNCExMenuSetDisplay(-1, "GNCExMenu00000", 1, 1, 0);

GCSNCExMenuGetDisplay	Get display type of menu

Syntax	GCSNCExMenuGetDisplay (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Display type 0: Square 1: Image
Details	Gets the display type of menu. When a setting outside the range is made, "0: Square" is returned.
Example	Gets the display type of the 1st menu button of GNCExMenu00000 in the self screen in Stat. CHAR Stat; Stat = GCSNCExMenuGetDisplay(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetUpperMenuOnPattern	Set pattern at the time of upper row menu ON

Syntax	GCSNCExMenuSetUpperMenuOnPattern (lWindowNo, strName, usMGNum, nPattern);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT nPattern : pattern
Return value	None
Details	Sets the pattern at the time of the upper row menu ON. One of the following values is set for nPattern 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out When a setting outside the range is made, the setting value is not changed..
Example	Sets the pattern at the time of the upper row menu ON to "pattern 0" for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOnPattern(-1, "GNCExMenu00000", 1, 2);

GCSNCExMenuGetUpperMenuOnPattern	Get pattern at the time of upper row menu ON

Syntax	GCSNCExMenuGetUpperMenuOnPattern (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Pattern 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out
Details	Gets the pattern at the time of the upper row menu ON. When a setting outside the range is made, "0: Background painting out" is returned.
Example	Gets the pattern at the time of the upper row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetUpperMenuOnPattern (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOnPattern	Set pattern at the time of lower row menu ON
Syntax	GCSNCExMenuSetLowerMenuOnPattern (lWindowNo, strName, usMGNum, nPattern);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT nPattern : pattern
Return value	None
Details	Sets the pattern at the time of the lower row menu ON. One of the following values is set for nPattern. 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out When a setting outside the range is made, the setting value is not changed..
Example	Sets the pattern at the time of the lower row menu ON to "pattern 0" for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOnPattern(-1, "GNCExMenu00000", 1, 2);

GCSNCExMenuGetLowerMenuOnPattern	Get pattern at the time of lower row menu ON

Syntax

GCSNCExMenuGetLowerMenuOnPattern (lWindowNo, strName, usMGNum);

Argument

(i)LONG	lWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING	strName	: control name
(i)SHORT	usMGNum	: menu group No. (1 to 10)

Return value

Pattern

0	: Background painting out
1	: Foreground painting out
2 to 39	: Pattern 0-37
40	: With no painting out

Details

Gets the pattern at the time of the lower row menu ON.
When a setting outside the range is made, "0: Background painting out " is returned.

Example

Gets the pattern at the time of the lower row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat.

```
SHORT Stat;
Stat = GCSNCExMenuGetLowerMenuOnPattern (-1,
"GNCExMenu00000", 1);
```

GCSNCExMenuSetUpperMenuOnForeColor	Set foreground color at the time of upper row menu ON

Syntax	GCSNCExMenuSetUpperMenuOnForeColor (lWindowNo, strName, usMGNum, lForeColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lForeColor : foreground color
Return value	None
Details	Sets the foreground color at the time of the upper row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the foreground color at the time of the upper row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOnForeColor(-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetUpperMenuOnForeColor	Get foreground color at the time of upper row menu ON

Syntax	GCSNCExMenuGetUpperMenuOnForeColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Foreground color
Details	Gets the foreground color at the time of the upper row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the foreground color at the time of the upper row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetUpperMenuOnForeColor(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOnForeColor	Set foreground color at the time of lower row menu ON

Syntax	GCSNCExMenuSetLowerMenuOnForeColor (lWindowNo, strName, usMGNum, lForeColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lForeColor : foreground color
Return value	None
Details	Sets the foreground color at the time of the lower row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the foreground color at the time of the lower row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOnForeColor(-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOnForeColor	Get foreground color at the time of lower row menu ON

Syntax	GCSNCExMenuGetLowerMenuOnForeColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Foreground color
Details	Gets the foreground color at the time of the lower row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the foreground color at the time of the lower row menu ON for menu group 1 in GNCExMenu00000 of the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOnForeColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetUpperMenuOnBackColor	Set background color at the time of upper row menu ON

Syntax	GCSNCExMenuSetUpperMenuOnBackColor (lWindowNo, strName, usMGNum, lBackColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lBackColor : background color
Return value	None
Details	Sets the background color at the time of the upper row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the background color at the time of the upper row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOnBackColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetUpperMenuOnBackColor	Get background color at the time of upper row menu ON

Syntax	GCSNCExMenuGetUpperMenuOnBackColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Background color
Details	Gets the background color at the time of the upper row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the background color at the time of the upper row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat LONG Stat; Stat = GCSNCExMenuGetUpperMenuOnBackColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOnBackColor	Set background color at the time of lower row menu ON

Syntax	GCSNCExMenuSetLowerMenuOnBackColor (lWindowNo, strName, usMGNum, lBackColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lBackColor : background color
Return value	None
Details	Sets the background color at the time of the lower row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the background color at the time of the lower row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOnBackColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOnBackColor	Get background color at the time of lower row menu ON

Syntax	GCSNCExMenuGetLowerMenuOnBackColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Background color
Details	Gets the background color at the time of the lower row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the background color at the time of the lower row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOnBackColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetMenuOnImage	Set image ID at the time of menu ON

Syntax	GCSNCExMenuSetMenuOnImage (lWindowNo, strName, usMGNum, usMBNum, usImageID);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usImageID : image ID
Return value	None
Details	Sets the image ID at the time of the menu ON. One of the following values is set for usImageID. 0 : No image 1 or later : User definition image ID When a setting outside the range is made, the setting value is not changed..
Example	Sets the image ID at the time of ON for the GNCExMenu00000 1st menu button in the self screen to "1". GCSNCExMenuSetMenuOnImage(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetMenuOnImage	Get image ID at the time of menu ON

Syntax	GCSNCExMenuGetMenuOnImage (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Image ID 0 : No image 1 or later : User definition image ID
Details	Gets the image ID at the time of the menu ON. When a setting outside the range is made, "0: No image" is returned.
Example	Gets the image ID at the time of ON for the GNCExMenu00000 1st menu button in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuOnImage(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetUpperMenuOnStringColor	Set character color at the time of upper row menu ON

Syntax	GCSNCExMenuSetUpperMenuOnStringColor (lWindowNo, strName, usMGNum, lStrColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lStrColor : character color
Return value	None
Details	Sets the character color at the time of the upper row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the character color at the time of upper row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOnStringColor(-1, "GNCExMenu00000", 1, H008000)

GCSNCExMenuGetUpperMenuOnStringColor	Get character color at the time of upper row menu ON

Syntax	GCSNCExMenuGetUpperMenuOnStringColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Character color
Details	Gets the character color at the time of the upper row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the character color at the time of upper row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetUpperMenuOnStringColor(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOnStringColor	Set character color at the time of lower row menu ON

Syntax	GCSNCExMenuSetLowerMenuOnStringColor (lWindowNo, strName, usMGNum, lStrColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lStrColor : character color
Return value	None
Details	Sets the character color at the time of the lower row menu ON. When a setting outside the range is made, the setting value is not changed..
Example	Sets the character color at the time of lower row menu ON to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOnStringColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOnStringColor	Get character color at the time of lower row menu ON

Syntax	GCSNCExMenuGetLowerMenuOnStringColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Character color
Details	Gets the character color at the time of the lower row menu ON. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the character color at the time of lower row menu ON for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOnStringColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetUpperMenuOffPattern	Set pattern at the time of upper row menu OFF

Syntax	GCSNCExMenuSetUpperMenuOffPattern (lWindowNo, strName, usMGNum, nPattern);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT nPattern : pattern setting value
Return value	None
Details	Sets the pattern at the time of the upper row menu OFF. One of the following values is set for nPattern. 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out When a setting outside the range is made, the setting value is not changed..
Example	Sets the pattern at the time of the upper row menu OFF to pattern "0" for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOffPattern(-1, "GNCExMenu00000", 1, 2);

GCSNCExMenuGetUpperMenuOffPattern	Get pattern at the time of upper row menu OFF

Syntax	GCSNCExMenuGetUpperMenuOffPattern (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Pattern 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out
Details	Gets the pattern at the time of the upper row menu OFF. When a setting outside the range is made, "0: Background painting out" is returned.
Example	Gets the pattern at the time of the upper row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetUpperMenuOffPattern(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOffPattern	Set pattern at the time of lower row menu OFF

Syntax	GCSNCExMenuSetLowerMenuOffPattern (lWindowNo, strName, usMGNum, nPattern);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT nPattern : pattern setting value
Return value	None
Details	Sets the pattern at the time of the lower row menu OFF. One of the following values is set for nPattern. 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out When a setting outside the range is made, the setting value is not changed..
Example	Sets the pattern at the time of the lower row menu OFF to pattern "0" for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOffPattern (-1, "GNCExMenu00000", 1, 2);

GCSNCExMenuGetLowerMenuOffPattern	Get pattern at the time of lower row menu OFF

Syntax	GCSNCExMenuGetLowerMenuOffPattern (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Pattern 0 : Background painting out 1 : Foreground painting out 2 to 39 : Pattern 0-37 40 : With no painting out
Details	Gets the pattern at the time of the lower row menu OFF. When a setting outside the range is made, "0: Background painting out" is returned.
Example	Gets the pattern at the time of the lower row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetLowerMenuOffPattern (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetUpperMenuOffForeColor	Set foreground color at the time of upper row menu OFF

Syntax	GCSNCExMenuSetUpperMenuOffForeColor (lWindowNo, strName, usMGNum, lForeColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lForeColor : foreground color
Return value	None
Details	Sets the foreground color at the time of the upper row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the foreground color at the time of the upper row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOffForeColor(-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetUpperMenuOffForeColor	Get foreground color at the time of upper row menu OFF

Syntax	GCSNCExMenuGetUpperMenuOffForeColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Foreground color
Details	Gets the foreground color at the time of the upper row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the foreground color at the time of the upper row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetUpperMenuOffForeColor(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOffForeColor	Set foreground color at the time of lower row menu OFF

Syntax	GCSNCExMenuSetLowerMenuOffForeColor (lWindowNo, strName, usMGNum, lForeColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lForeColor : foreground color
Return value	None
Details	Sets the foreground color at the time of the lower row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the foreground color at the time of the lower row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOffForeColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOffForeColor	Get foreground color at the time of lower row menu OFF

Syntax	GCSNCExMenuGetLowerMenuOffForeColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Foreground color
Details	Gets the foreground color at the time of the lower row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the foreground color at the time of the lower row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOffForeColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetUpperMenuOffBackColor	Set background color at the time of upper row menu OFF

Syntax	GCSNCExMenuSetUpperMenuOffBackColor (lWindowNo, strName, usMGNum, lBackColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lBackColor : background color
Return value	None
Details	Sets the background color at the time of the upper row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the background color at the time of the upper row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOffBackColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetUpperMenuOffBackColor	Get background color at the time of upper row menu OFF

Syntax	GCSNCExMenuGetUpperMenuOffBackColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Background color
Details	Gets the background color at the time of the upper row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the background color at the time of the upper row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetUpperMenuOffBackColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOffBackColor	Set background color at the time of lower row menu OFF

Syntax	GCSNCExMenuSetLowerMenuOffBackColor (lWindowNo, strName, usMGNum, lBackColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lBackColor : background color
Return value	None
Details	Sets the background color at the time of the lower row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the background color at the time of the lower row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOffBackColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOffBackColor	Get background color at the time of lower row menu OFF

Syntax	GCSNCExMenuGetLowerMenuOffBackColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Background color
Details	Gets the background color at the time of the lower row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the background color at the time of the lower row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOffBackColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetMenuOffImage	Set image ID at the time of menu OFF

Syntax	GCSNCExMenuSetMenuOffImage (lWindowNo, strName, usMGNum, usMBNum, usImageID);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usImageID : Image ID
Return value	None
Details	Sets the image ID at the time of the menu OFF. One of the following values is set for usImageID. 0 : No image 1 or later : User definition image ID When a setting outside the range is made, the setting value is not changed..
Example	Sets the image ID at the time of OFF for the GNCExMenu00000 1st menu button in the self screen to "1". GCSNCExMenuSetMenuOffImage(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetMenuOffImage	Get image ID at the time of menu OFF

Syntax	GCSNCExMenuGetMenuOffImage (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Image ID 0 : No image 1 or later : User definition image ID
Details	Gets the image ID at the time of the menu OFF. When a setting outside the range is made, "0: No image" is returned.
Example	Gets the image ID at the time of OFF for the GNCExMenu00000 1st menu button in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuOffImage(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetUpperMenuOffStringColor	Set character color at the time of upper row menu OFF

Syntax	GCSNCExMenuSetUpperMenuOffStringColor (lWindowNo, strName, usMGNum, lStrColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lStrColor : character color
Return value	None
Details	Sets the character color at the time of the upper row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the character color at the time of the upper row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetUpperMenuOffStringColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetUpperMenuOffStringColor	Get character color at the time of upper row menu OFF

Syntax	GCSNCExMenuGetUpperMenuOffStringColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Character color
Details	Gets the character color at the time of the upper row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the character color at the time of the upper row menu OFF for menu group 1 of GNCExMenu00000 in the self screen. LONG Stat; Stat = GCSNCExMenuGetUpperMenuOffStringColor(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetLowerMenuOffStringColor	Set character color at the time of lower row menu OFF

Syntax	GCSNCExMenuSetLowerMenuOffStringColor (lWindowNo, strName, usMGNum, lStrColor);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)LONG lStrColor : character color
Return value	None
Details	Sets the character color at the time of the lower row menu OFF. When a setting outside the range is made, the setting value is not changed..
Example	Sets the character color at the time of the lower row menu OFF to (green: H008000) for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetLowerMenuOffStringColor (-1, "GNCExMenu00000", 1, H008000);

GCSNCExMenuGetLowerMenuOffStringColor	Get character color at the time of lower row menu OFF

Syntax	GCSNCExMenuGetLowerMenuOffStringColor (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Character color
Details	Gets the character color at the time of the lower row menu OFF. When a setting outside the range is made, "0: Black" is returned.
Example	Gets the character color at the time of the lower row menu OFF for menu group 1 of GNCExMenu00000 in the self screen in Stat. LONG Stat; Stat = GCSNCExMenuGetLowerMenuOffStringColor (-1, "GNCExMenu00000", 1);

GCSNCExMenuSetMenuAction	Set button operations of menu

Syntax	GCSNCExMenuSetMenuAction (lWindowNo, strName, usMGNum, usMBNum, usAction);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usAction : button operation
Return value	None
Details	Sets the button operation of the menu. One of the following values is set for usAction. 0: No operation 1: Momentary 2: Alternate 3: Button group 1 4: Button group 2 5: Button group 3 6: Button group 4 7: Button group 5 When a setting outside the range is made, the setting value is not changed..
Example	Sets the button operation of the 1st menu button of GNCExMenu00000 in the self screen to "Momentary". GCSNCExMenuSetMenuAction(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetMenuAction	Get button operations of menu

Syntax	GCSNCExMenuGetMenuAction (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Button operation 0: No operation 1: Momentary 2: Alternate 3: Button group 1 4: Button group 2 5: Button group 3 6: Button group 4 7: Button group 5
Details	Gets the button operation of the menu. When a setting outside the range is made, "0: No operation" is returned.
Example	Gets the button operation of the 1st menu button of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuAction(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuString	Set character string of menu

Syntax	GCSNCExMenuSetMenuString (lWindowNo, strName, usMGNum, usMBNum, strMenu);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)STRING strMenu : character string
Return value	None
Details	Sets the character string of the menu. When a setting outside the range is made, the setting value is not changed..
Example	Sets the character string of the 1st menu button of GNCExMenu00000 in the self screen to "MENU1". GCSNCExMenuSetMenuString(-1, "GNCExMenu00000", 1, 1, "MENU1");

GCSNCExMenuGetMenuString	Get character string of menu

Syntax	GCSNCExMenuGetMenuString (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Character string
Details	Gets the character string of the menu. When a setting outside the range is made, null character is returned.
Example	Gets the character string of the 1st menu button of GNCExMenu00000 in the self screen in Stat. STRING Stat; Stat = GCSNCExMenuGetMenuString(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuStringID	Set character string resource ID of menu

Syntax	GCSNCExMenuSetMenuStringID (lWindowNo, strName, usMGNum, usMenuStrID);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMenuStrID : character string resource ID
Return value	None
Details	Sets the character string resource ID of menu. (Sets ten designated menu group buttons) When a setting outside the range is made, the setting value is not changed..
Example	Sets the character string resource ID to "1" for menu group 1 of GNCExMenu00000 in the self screen. GCSNCExMenuSetMenuStringID(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuGetMenuStringID	Get character string resource ID of menu

Syntax	GCSNCExMenuGetMenuStringID (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Character resource ID
Details	Gets the character string resource ID of menu. When a setting outside the range is made, "0" is returned.
Example	Gets the character string resource ID for menu group 1 of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuStringID(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetMenuPageAction	Set screen switch operation

Syntax	GCSNCExMenuSetMenuPageAction (lWindowNo, strName, usMGNum, usMBNum, usPageAction);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usPageAction : Screen switch operation
Return value	None
Details	Sets the screen switch operation. One of the following values is set for usPageAction. 0: None 1: Open Panel 2: Open Window 3: Close Window 4: ON: Open win., OFF: Close win. 5: ON: Close win., OFF: Open win. When a setting outside the range is made, the setting value is not changed..
Example	Sets the screen switch operation setting value of the 1st menu button of GNCExMenu00000 in the self screen to "4". GCSNCExMenuSetMenuPageAction(-1, "GNCExMenu00000", 1, 1, 4);

GCSNCExMenuGetMenuPageAction	Get screen switch operation

Syntax	GCSNCExMenuGetMenuPageAction (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Screen switch operations 0: None 1: Open Panel 2: Open Window 3: Close Window 4: ON: Open win., OFF: Close win. 5: ON: Close win., OFF: Open win.
Details	Gets the screen switch operation. When a setting outside the range is made, "0: None" is returned.
Example	Gets the screen switch operation setting value of the 1st menu button of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuPageAction(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuPageNum	Set Panel/Window number

Syntax	GCSNCExMenuSetMenuPageNum (lWindowNo, strName, usMGNum, usMBNum, nPageNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT nPageNum : setting value of Panel/Window No.
Return value	None
Details	Sets the Panel/Window number. When a setting outside the range is made, the setting value is not changed..
Example	Sets the Panel/Window number setting value of the 1st menu button of GNCExMenu00000 in the self screen to "1". GCSNCExMenuSetMenuPageNum(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetMenuPageNum	Get Panel/Window number

Syntax	GCSNCExMenuGetMenuPageNum (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Panel/Window number
Details	Gets the Panel/Window number. When a setting outside the range is made, "0" is returned.
Example	Gets the Panel/Window number setting value of the 1st menu button of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuPageNum(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuStatus	Set menu button status

Syntax	GCSNCExMenuSetMenuStatus (lWindowNo, strName, usMGNum, usMBNum, usStatus);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usStatus : menu button status
Return value	None
Details	Sets the menu button status. One of the following values is set for usStatus. 0: OFF status 1: ON status When a setting outside the range is made, the setting value is not changed..
Example	Sets the menu button status setting value of the 1st menu button of GNCExMenu00000 in the self screen to "ON status". GCSNCExMenuSetMenuStatus(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetMenuStatus	Get menu button status

Syntax	GCSNCExMenuGetMenuStatus (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Menu button status 0: OFF status 1: ON status
Details	Gets the menu button status. When a setting outside the range is made, "0: OFF status" is returned.
Example	Gets the menu button status setting value of the 1st menu button of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuStatus(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuViewStatus	Set display status of menu

Syntax	GCSNCExMenuSetMenuViewStatus (lWindowNo, strName, usMGNum, usMBNum, usViewSts);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT usViewSts : setting value of display status
Return value	None
Details	Sets the display status of the menu. One of the following values is set for usViewSts. 0: Hide 1: Partial display 2: Whole display 3: Invalid display When a setting outside the range is made, the setting value is not changed..
Example	Sets the display status setting value of the 1st menu button of GNCExMenu00000 in the self screen to "Invalid display". GCSNCExMenuSetMenuViewStatus(-1, "GNCExMenu00000", 1, 1, 3);

GCSNCExMenuGetMenuViewStatus	Get display status of menu

Syntax	GCSNCExMenuGetMenuViewStatus (lWindowNo, strName, usMGNum, usMBNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10)
Return value	Display status 0: Hide 1: Partial display 2: Whole display 3: Invalid display
Details	Gets the display status of the menu. When a setting outside the range is made, "0: Hide" is returned.
Example	Gets the display status setting value of the 1st menu button of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetMenuViewStatus(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuSetMenuStringPosition	Set character string display position of menu

Syntax	GCSNCExMenuSetMenuStringPosition (lWindowNo, strName, usMGNum, ucHPosition, ucVPosition);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)CHAR ucHPosition : horizontal position (i)CHAR ucVPosition : vertical position
Return value	None
Details	Sets the character string display position of the menu. One of the following values is set for ucHPosition. 0: Align left 1: Center 2: Align right One of the following values is set for ucVPosition. 0: Align top 1: Center 2: Align bottom When a setting outside the range is made, the setting value is not changed..
Example	Sets the character string display position of the menu for menu group 1 of GNCExMenu00000 in the self screen to "Align left" for horizontal position, and to "Center" for vertical position. GCSNCExMenuSetMenuStringPosition(-1, "GNCExMenu00000", 1, 0, 1);

GCSNCExMenuGetMenuStringPosition	Get character string display position of menu

Syntax

GCSNCExMenuGetMenuStringPosition (lWindowNo, strName, usMGNum, memHPosition, memVPosition);

Argument

(i)LONG lWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)SHORT usMGNu	: menu group No. (1 to 10)
(o)GMEM memHPosition	: reservation of global memory area to store horizontal position : CHAR(1)
(o)GMEM memVPosition	: reservation of global memory area to store vertical position : CHAR(1)

Return value

None

Details

Gets the character string display position of the menu.
Gets one of the following values for memHPosition.
Horizontal position
0: Align left
1: Center
2: Align right
Gets one of the following values for memVPosition.
Vertical position
0: Align top
1: Center
2: Align bottom
When a setting outside the range is made, "1: Center" is returned for horizontal position and vertical position.

Example

Gets the character string display position of the menu for menu group 1 of GNCExMenu00000 in the self screen to "memHPos" for horizontal position, and to "memVPos" for vertical position.

```
CHAR cHPos;       'Horizontal position
CHAR cVPos;       'Vertical position
GMEM memHPos;
GMEM memVPos;
memHPos = GMEMCreate("MEMHPOS", 1);
memVPos = GMEMCreate("MEMVPOS", 1);
GCSNCExMenuGetMenuStringPosition(-1, "GNCExMenu00000",
1, memHPos, memVPos);
cHPos = GMEMGetChar(memHPos, 0);
cVPos = GMEMGetChar(memVPos, 0);
GMEMDelete(memHPos);
GMEMDelete(memVPos);
```

GCSNCExMenuSetUpperMenuGroupNum	Set upper row menu group number

Syntax

GCSNCExMenuSetUpperMenuGroupNum (lWindowNo, strName, usMGNum, usSetMGNum);

Argument

(i)LONG lWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)SHORT usMGNum	: menu group No. (1 to 10)
(i)SHORT usSetMGNum	: upper row menu group number (0 to 10)

Return value

None

Details

Sets the menu group number of the upper row in the two-row display.
One of the following values is set for usSetMGNum.

0	: Hide upper row menu
1 to 10	: Displays the specified group number menu to the upper row

When a setting outside the range is made, the setting value is not changed..

Example

Sets the upper row menu group number to "2" for menu group 1 of GNCExMenu00000 in the self screen.

```
GCSNCExMenuSetUpperMenuGroupNum(-1,
"GNCExMenu00000", 1, 2);
```

GCSNCExMenuGetUpperMenuGroupNum	Get upper row menu group number

Syntax	GCSNCExMenuGetUpperMenuGroupNum (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	Upper row menu group number 0 : Hide upper row menu 1 to 10 : Menu group number specified to the upper row menu
Details	Gets the menu group number of the upper row in the two-row display. When a setting outside the range is made, "0" is returned.
Example	Gets the upper row menu group number for menu group 1 of GNCExMenu00000 in the self screen in Stat. SHORT Stat; Stat = GCSNCExMenuGetUpperMenuGroupNum(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetNextMenuGroupNum	Set destination menu group number

Syntax

GCSNCExMenuSetNextMenuGroupNum (lWindowNo, strName, usMGNum, usMBNum, usSetMGNum);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
return button, switch button, right-flick, left-flick : 0
(i)SHORT usMBNum : menu button No. (1 to 10)
return button : 11
switch button, right-flick : 12
left-flick : 13
(i)SHORT usSetMGNum : destination menu group number (0 to 10)

Return value

None

Details

Sets the destination menu group number.
One of the following values is set for usSetMGNum.
0 : No movement
1 to 10 : Display the designated group number menu.
When a setting outside the range is made, the setting value is not changed..

Example

Sets the destination menu group number of the 1st menu button of GNCExMenu00000 in the self screen to "2".
GCSNCExMenuSetNextMenuGroupNum(-1, "GNCExMenu00000", 1, 1, 2);

GCSNCExMenuGetNextMenuGroupNum	Get destination menu group number

Syntax

GCSNCExMenuGetNextMenuGroupNum (lWindowNo, strName, usMGNum, usMBNum);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
return button, switch button, right-flick, left-flick : 0
(i)SHORT usMBNum : menu button No. (1 to 10)
return button : 11
switch button, right-flick : 12
left-flick : 13

Return value

Destination menu group number.
0 : No movement
1 to 10 : Display the designated group number menu.

Details

Gets the destination menu group number.
When a setting outside the range is made, "0: No movement" is returned.

Example

Gets the destination menu group number of the 1st menu button of GNCExMenu00000 in the self screen in Stat.

```
SHORT Stat;
Stat = GCSNCExMenuGetNextMenuGroupNum(-1,
"GNCExMenu00000", 1, 1);
```

GCSNCExMenuChangeMenuGroupNum	Change menu group number to be displayed
Syntax	GCSNCExMenuChangeMenuGroupNum (lWindowNo, strName, usMGNum);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10)
Return value	None
Details	Changes the menu group number to be displayed. When a setting outside the range is made, the setting value is not changed.. (Note) The upper row menu displays the group set in the "upper row menu group number" property of the designated menu group number.
Example	Changes the menu group currently displayed in GNCExMenu00000 in the self screen to "menu group 1". GCSNCExMenuChangeMenuGroupNum(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetOpeMessage	Set operation message (designate character string)

Syntax	GCSNCExMenuSetOpeMessage (lWindowNo, strName, strOpeMessage);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)STRING strOpeMessage : operation message character string
Return value	None
Details	Sets operation messages using character string. Up to 40 characters can be used for strOpeMessage.
Example	Sets the operation message of GNCExMenu00000 in the self screen to "executing". GCSNCExMenuSetOpeMessage(-1, "GNCExMenu00000", "executing");

GCSNCExMenuSetOpeMessageID	Set operation message (designate resource ID)

Syntax	GCSNCExMenuSetOpeMessageID (lWindowNo, strName, usOpeMessageString)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usOpeMessageString : character string resource ID of operation message
Return value	None
Details	Sets operation messages using character string resource ID.
Example	Sets the character string resource ID of the operation message of GNCExMenu00000 in the self screen to "1". GCSNCExMenuSetOpeMessageID(-1, "GNCExMenu00000", 1);

GCSNCExMenuSetMenuDevice	Set PLC device

Syntax

GCSNCExMenuSetMenuDevice (lWindowNo, strName, usMGNum, usMBNum, nIndex, strDevice);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
(i)SHORT usMBNum : menu button No. (1 to 10)
(i)SHORT nIndex : PLC device index No.
0: Write PLC device
1: Read PLC device
(i)STRING strDevice : PLC device character string

Return value

Setting result
0xA001: Menu group No. is not correct.
0xA002: Menu button No. is not correct.
0xA003: PLC device index is not correct.
0xA004: PLC device is not correct.
0x0 : Normal

Details

Sets the PLC device address to allocate to the control.

Example

Sets the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen to PLC device "X0".

```
GCSNCExMenuSetMenuDevice(-1, "GNCExMenu00000", 1, 1, 0, "X0");
```

GCSNCExMenuGetMenuDevice	Get PLC device

Syntax	GCSNCExMenuGetMenuDevice (lWindowNo, strName, usMGNum, usMBNum, nIndex)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT nIndex : PLC device index No. 0: Write PLC device 1: Read PLC device
Return value	PLC device character string
Details	Gets the PLC device address allocated to the control. (Note) Gets "X0008" when set to "X0008", and "X8" when set to "X8".
Example	Gets the PLC device character string of write PLC device of the 1st menu button of GNCExMenu00000 in the self screen in Stat. STRING Stat; Stat = GCSNCExMenuGetMenuDevice(-1, "GNCExMenu00000", 1, 1, 0);

GCSNCExMenuSetMenuDeviceEnable	Set PLC device enabled

Syntax

GCSNCExMenuSetMenuDeviceEnable (lWindowNo, strName, usMGNum, usMBNum, nIndex, usType);

Argument

(i)LONG lWindowNo	: screen No. (Specify -1 for self screen.)
(i)STRING strName	: control name
(i)SHORT usMGNum	: menu group No. (1 to 10)
(i)SHORT usMBNum	: menu button No. (1 to 10)
(i)SHORT nIndex	: PLC device index No. 0: Write PLC device 1: Read PLC device
(i)SHORT usEnable	: PLC device enabled

Return value

Setting result

0xA001	: Menu group No. is not correct.
0xA002	: Menu button No. is not correct.
0xA003	: PLC device index is not correct.
0xA005	: PLC device enabled is not correct.
0x0	: Normal

Details

Sets the PLC device enabled status to allocate to the control.
One of the following values is set for usEnable.
0: Disabled
1: Enabled
2: Same setting as Write PLC device

Example

Sets the PLC device enabled of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen to "Enabled".

```
GCSNCExMenuSetMenuDeviceEnable(-1, "GNCExMenu00000", 1, 1, 0, 1);
```

GCSNCExMenuGetMenuDeviceEnable	Get PLC device enabled

Syntax

GCSNCExMenuGetMenuDeviceEnable (lWindowNo, strName, usMGNum, usMBNum, nIndex)

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
(i)SHORT usMBNum : menu button No. (1 to 10)
(i)SHORT nIndex : PLC device index No.
0: Write PLC device
1: Read PLC device

Return value

PLC device enabled
0: Disabled
1: Enabled
2: Same setting as Write PLC device

Details

Gets the PLC device enabled status allocated to the control.

Example

Gets the PLC device enabled of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen in Stat.

```
SHORT Stat;
Stat = GCSNCExMenuGetMenuDeviceEnable(-1,
"GNCExMenu00000", 1, 1, 0);
```

GCSNCExMenuSetMenuDeviceBitPosition	Set bit position of PLC device

Syntax

GCSNCExMenuSetMenuDeviceBitPosition (lWindowNo, strName, usMGNum, usMBNum, nIndex, ucDeviceBitPos);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
(i)SHORT usMBNum : menu button No. (1 to 10)
(i)SHORT nIndex : PLC device index No.
0: Write PLC device
1: Read PLC device
(i)CHAR ucDeviceBitPos : Bit position of PLC device (0 to 15)

Return value

Setting result
0xA001: Menu group No. is not correct.
0xA002: Menu button No. is not correct.
0xA003: PLC device index is not correct.
0xA006: Bit position of PLC device is not correct.
0x0 : Normal

Details

Sets the bit position of the PLC device to allocate to the control.

Example

Sets the bit position of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen to "5".

```
GCSNCExMenuSetMenuDeviceBitPosition(-1,
"GNCExMenu00000", 1, 1, 0, 5);
```

GCSNCExMenuGetMenuDeviceBitPosition	Get bit position of PLC device

Syntax	GCSNCExMenuGetMenuDeviceBitPosition (lWindowNo, strName, usMGNum, usMBNum, nIndex)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT nIndex : PLC device index No. 0: Write PLC device 1: Read PLC device
Return value	Bit position of PLC device (0 to 15)
Details	Gets the bit position of the PLC device allocated to the control.
Example	Gets the bit position of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen in Stat. CHAR Stat; Stat = GCSNCExMenuGetMenuDeviceBitPosition(-1, "GNCExMenu00000", 1, 1, 0);

GCSNCExMenuSetMenuDeviceVal	Set PLC device value

Syntax

GCSNCExMenuSetMenuDeviceVal (lWindowNo, strName, usMGNum, usMBNum, nIndex, ucDeviceVal);

Argument

(i)LONG lWindowNo : screen No. (Specify -1 for self screen.)
(i)STRING strName : control name
(i)SHORT usMGNum : menu group No. (1 to 10)
(i)SHORT usMBNum : menu button No. (1 to 10)
(i)SHORT nIndex : PLC device index No.
0: Write PLC device
1: Read PLC device
(i)CHAR ucVal : PLC device value

Return value

Setting result
0xA001: Menu group No. is not correct.
0xA002: Menu button No. is not correct.
0xA003: PLC device index is not correct.
0xA007: Writing failure
0x0 : Normal

Details

Sets the PLC device value to allocate to the control.
(Note) Monitors the write PLC device and the read PLC device using Timer event. Changing the device by ladder side changes the menu button status according to the device status. When writing a PLC device by both custom screen side and ladder side, the last writing is valid.

Example

Sets the PLC device value of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen to "1".
GCSNCExMenuSetMenuDeviceVal(-1, "GNCExMenu00000", 1, 1, 0, 1);

GCSNCExMenuGetMenuDeviceVal	Get PLC device value

Syntax	GCSNCExMenuGetMenuDeviceVal (lWindowNo, strName, usMGNum, usMBNum, nIndex)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBNum : menu button No. (1 to 10) (i)SHORT nIndex : PLC device index No. 0: Write PLC device 1: Read PLC device
Return value	PLC device value
Details	Gets the PLC device value allocated to the control.
Example	Gets the PLC device value of the write PLC device of the 1st menu button of GNCExMenu00000 in the self screen in Stat. CHAR Stat; Stat = GCSNCExMenuGetMenuDeviceVal (-1, "GNCExMenu00000", 1, 1, 0);

GCSNCExMenuSetButtonGroupAllOffState	Set all OFF state of button group

Syntax	GCSNCExMenuSetButtonGroupAllOffState (lWindowNo, strName, usMGNum, usMBGNum, ucStatus);
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBGNum : button group No. (1 to 5) (i)CHAR ucStatus : all OFF state
Return value	None
Details	Sets all OFF state of the button group. One of the following values is set for ucStatus. 0: Permission 1: Prohibition
Example	Sets all OFF state of the 1st button group of GNCExMenu00000 in the self screen to "Prohibition". GCSNCExMenuSetButtonGroupAllOff(-1, "GNCExMenu00000", 1, 1, 1);

GCSNCExMenuGetButtonGroupAllOffState	Get all OFF state of button group

Syntax	GCSNCExMenuGetButtonGroupAllOffState (lWindowNo, strName, usMGNum, usMBGNum)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT usMGNum : menu group No. (1 to 10) (i)SHORT usMBGNum : button group No. (1 to 5)
Return value	All OFF state 0: Permission 1: Prohibition
Details	Gets the all OFF state of the button group.
Example	Gets the setting value of all OFF state of the 1st button group of GNCExMenu00000 in the self screen in Stat. CHAR Stat; Stat = GCSNCExMenuGetButtonGroupAllOff(-1, "GNCExMenu00000", 1, 1);

GCSNCExMenuBeginAnimState	Start menu animated motion setting

Syntax	GCSNCExMenuBeginAnimState (lWindowNo, strName, nMenuAnmType)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name (i)SHORT nMenuAnmType : menu animation direction
Return value	None
Details	Starts the menu state change. One of the following values is set for nMenuAnmType. 0: No animation 1: Upward 2: Downward 3: Leftward 4: Rightward
Example	Switches the display to menu group 2 while animating upward for GNCExMenu00000 in the self screen. GCSNCExMenuBeginAnimState (-1, "GNCExMenu00000", 1); GCSNCExMenuChangeMenuGroupNum (-1, "GNCExMenu00000", 2); GCSNCExMenuEndAnimState (-1, "GNCExMenu00000");

GCSNCExMenuEndAnimState	Complete menu animated motion setting (start animation)

Syntax	GCSNCExMenuEndAnimState (lWindowNo, strName)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	None
Details	Completes the menu state change and starts menu animation.
Example	Switches the display to menu group 2 while animating upward for GNCExMenu00000 in the self screen. GCSNCExMenuBeginAnimState (-1, "GNCExMenu00000", 1); GCSNCExMenuChangeMenuGroupNum (-1, "GNCExMenu00000", 2); GCSNCExMenuEndAnimState (-1, "GNCExMenu00000");

GCSNCExMenuStopAnim	Stop menu animation
Syntax	GCSNCExMenuStopAnim (lWindowNo, strName)
Argument	(i)LONG lWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	None
Details	Stops menu animation and displays the menu after the state has been changed.
Example	Stops the current menu animation and displays the menu after the state has been changed for GNCExMenu00000 in the self screen. GCSNCExMenuStopAnim (-1, "GNCExMenu00000");

6. NC Data Access Function

GCSNCDataSetGNCValue	Set numerical value in NC

Syntax

GCSNCDataSetGNCValue(gmNCControl, lSection, lSubSection, lDataType, gmValue);

Argument

(i)GMEM gmNCControl : global memory for NC information (GNCControl)
(i)LONG lSection : section No.
(i)LONG lSubSection : sub-section No.
(i)LONG lDataType : data type
(i)GMEM gmValue : global memory to store numerical value

Return value

0: setting failed
1: setting succeeded

Details

Set the value stored in gmValue to NC.
One of the following values is set for lDataType:
0x1 :1-byte integer type
0x2 :2-byte integer type
0x3 :4-byte integer type
0x5 :8-byte real type

Example

Sets <empty> to the common variable #100 of the 2nd basic part system in the NC No. 1.

```
GMEM gControl ;
gControl = GMEMCreate("NCCONTROL", 16) ;
GMEMSetLong(gControl, 0, 1) ; ' Set the NC No. to 1.
GMEMSetLong(gControl, 4, 2) ; ' Select the 2nd part system.
GMEMSetLong(gControl, 8, 0) ; ' Select basic part system.
GMEMSetLong(gControl, 12, H0) ; ' Not specify the axis number.

GMEM NaN ;
NaN = GMEMCreate("KUU", 8) ;
GMEMSetLong(NaN, 0, HFFFFFFFF) ;
GMEMSetLong(NaN, 4, H7FFFFFFF) ;

LONG Stat ;
Stat = GCSNCDataSetGNCValue(gControl, 4, 800, 5, NaN) ;
GMEMDelete(gControl);
GMEMDelete(NaN);
```

GCSNCDataGetGNCValue	Get numerical value from NC

Syntax

GCSNCDataGetGNCValue(gmNCControl, lSection, lSubSection, lDataType, gmValue);

Argument

(i)GMEM gmNCControl : global memory for NC information (GNCControl)
(i)LONG lSection : section No.
(i)LONG lSubSection : sub-section No.
(i)LONG lDataType : data type
(o)GMEM gmValue : global memory to store numerical value

Return value

0: acquisition failed
1: acquisition succeeded
2: variable <empty>

Details

Stores the numerical value data got from NC in gmValue.
One of the following values is set for lDataType:
0x1 :1-byte integer type
0x2 :2-byte integer type
0x3 :4-byte integer type
0x5 :8-byte real type

Example

Gets the common variable #100 of the 2nd basic part system in the NC No. 1.

```
GMEM gControl ;
gControl = GMEMCreate("NCCONTROL", 16) ;
GMEMSetLong(gControl, 0, 1) ; ' Set the NC No. to 1.
GMEMSetLong(gControl, 4, 2) ; ' Select the 2nd part system.
GMEMSetLong(gControl, 8, 0) ; ' Select basic part system.
GMEMSetLong(gControl, 12, H0) ; ' Not specify the axis number.

LONG Stat ;
GMEM mem ;
mem = GMEMCreate("TESTMEM", 8) ;
Stat = GCSNCDataGetGNCValue(gControl, 4, 800, 5, mem) ;

DOUBLE dVal;
dVal = GMEMGetDouble(mem, 0) ;
GMEMDelete(gControl);
GMEMDelete(mem);
```

GCSNCDataSetString	Set character string in NC
Syntax	GCSNCDataSetString(gmNCControl, lSection, lSubSection, pString);
Argument	(i)GMEM gmNCControl : Global memory for NC information (GNCControl) (i)LONG lSection : section No. (i)LONG lSubSection : sub-section No. (i)STRING pString : character string to be set
Return value	0: setting failed 1: setting succeeded
Details	Sets the character string stored in pString to NC.
Example	Sets the 3rd axis' name of the 2nd basic system in the NC No. 1 to "Z2". LONG Stat ; GMEM gControl ; gControl = GMEMCreate("NCCONTROL", 16) ; GMEMSetLong(gControl, 0, 1) ; ' Set the NC No. to 1. GMEMSetLong(gControl, 4, 2) ; ' Select the 2nd part system. GMEMSetLong(gControl, 8, 0) ; ' Select basic part system. GMEMSetLong(gControl, 12, 3) ; ' Set the axis number to 3. Stat = GCSNCDataSetString(gControl, 127, 2, "Z2") ; GMEMDelete(gControl);

GCSNCDataGetString	Get character string from NC
Syntax	GCSNCDataGetString(gmNCControl, lSection, lSubSection, pString);
Argument	(i)GMEM gmNCControl : Global memory for NC information (GNCControl) (i)LONG lSection : section No. (i)LONG lSubSection : sub-section No. (o)STRING pString : character string
Return value	0: acquisition failed 1: acquisition succeeded
Details	Stores the character string got from NC in pString.
Example	Gets the 3rd axis' name of the 2nd basic system in the NC No. 1. STRING strData ; LONG Stat ; GMEM gControl ; gControl = GMEMCreate("NCCONTROL", 16) ; GMEMSetLong(gControl, 0, 1) ; ' Set the NC No. to 1. GMEMSetLong(gControl, 4, 2) ; ' Select the 2nd part system. GMEMSetLong(gControl, 8, 0) ; ' Select basic part system. GMEMSetLong(gControl, 12, 3) ; ' Set the axis number. to 3. Stat = GCSNCDataGetString(gControl, 127, 2, strData) ; GMEMDelete(gControl);

melSetData	Set NC data

Syntax

melSetData(lAddress, lSectionNum, lSubSectionNum, lAxisFlag, Data);

Argument

(i)LONG lAddress : Address
Specify the following items.
- Specify NC.
- Specify the basic part system or current part system during cross control.
- Specify the part system.
- Specify the ground (Fore/Back).
- Specify the forced write mode.

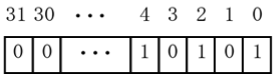
(i)LONG lSectionNum : Section No.
(i)LONG lSubSectionNum : Sub-section No.
(i)LONG lAxisFlag : Axis
(i) Variable-type Data : Setting data

Return value

ME_DATA_WRITE_ADDR (H80040290)	:	The specified address is not correct.
ME_DATA_WRITE_SECT (H80040291)	:	The section number is not correct.
ME_DATA_WRITE_SUBSECT (H80040292)	:	The sub-section number is not correct.
ME_DATA_WRITE_AXIS (H800402A0)	:	The specified axis is not correct.
ME_DATA_WRITE_DATASIZE (H80040296)	:	Data size over
ME_DATA_WRITE_DATATYPE (H80040297)	:	The specified data type is not correct.
ME_DATA_WRITE_READONLY (H8004029B)	:	Read only data
ME_DATA_WRITE_WRITE (H8004029E)	:	Cannot write data.

Details

Sets data that is specified for the Address, Section No., Sub-section No., and Axis.
For specifying an axis (AxisFlag), specify a flag (bit0 to bitF) corresponding to the axis. Data of multiple axes can be acquired at the same time by specifying multiple flags.
Axis specification example) When acquiring data of axis 1, 3, and 5.



For Variable-type Data, specify a variable type (including an array) other than the GMEM type.
When setting data of multiple axes, specify the target axis for "Axis", and set an array-type variable for "Setting data".

Example

```
LONG lData(3);
LONG lStatus;
LONG lAddress;
lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
'Set the data of each axis
lData(0)=100;
lData(1)=200;
lData(2)=300;
'Set the data of axis parameter [cutting feed clamp speed] 1st, 2nd and 5th axis (0x10011=H13)
lStatus = melSetData(lAddress, 127, 2002, H13, lData);
```

melGetData	Get NC data

Syntax

melGetData(lAddress, lSectionNum, lSubSectionNum, lAxisFlag, Data);

Argument

(i)LONG lAddress : Address
Specify the following items.
- Specify NC.
- Specify the basic part system or current part system during cross control.
- Specify the part system.
- Specify the ground (Fore/Back).

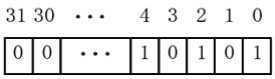
(i)LONG lSectionNum : Section No.
(i)LONG lSubSectionNum : Sub-section No.
(i)LONG lAxisFlag : Axis
(o)Variable-type Data : Obtained data

Return value

ME_DATA_READ_ADDR (H80040190)	:	The specified address is not correct.
ME_DATA_READ_SECT (H80040191)	:	The section number is not correct.
ME_DATA_READ_SUBSECT (H80040192)	:	The sub-section number is not correct.
ME_DATA_READ_DATASIZE (H80040196)	:	Data size over
ME_DATA_READ_READ (H8004019D)	:	Read disabled data
ME_DATA_READ_DATATYPE (H80040197)	:	The specified data type is not correct.
ME_DATA_READ_AXIS (H800401A0)	:	The specified axis is not correct.
ME_DATA_READ_WRITEONLY (H8004019F)	:	Write only data

Details

Gets the data that is specified using arguments.
For specifying an axis (AxisFlag), specify a flag (bit0 to bitF) corresponding to the axis. Data of multiple axes can be acquired at the same time by specifying multiple flags.
Axis specification example) When acquiring data of axis 1, 3, and 5.



For “Variable-type Data”, specify a variable type (including an array) other than the GMEM type.
When getting data of multiple axes, specify the target axis for "Axis", and set an array-type variable for "Obtained data”.

Example

```
LONG lData(3);
LONG lStatus;
LONG lAddress;
lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
'Get the data of axis parameter [cutting feed clamp speed] 1st, 2nd and 5th axis (0x10011=H13)
lStatus = melGetData(lAddress, 127, 2002, H13, lData);
```

melGetLumpData	Collectively get successive NC data items

Syntax

melGetLumpData(lAddress, lSectionNum, lSubSectionNum, lGetNos, Data);

Argument

(i)LONG lAddress : Address
Specify the following items.
- Specify NC.
- Specify the basic part system or current part system during cross control.
- Specify the part system.
- Specify the ground (Fore/Back).
- Specify the part system.

(i)LONG lSectionNum : Section No.
(i)LONG lSubSectTop : Head sub-section No.
(i)LONG lGetNos : Number of successive data items
(o)Variable-type Data : Obtained data

Return value

ME_DATA_READ_ADDR (H80040190)	:	The specified address is not correct.
ME_DATA_READ_SECT (H80040191)	:	The section number is not correct.
ME_DATA_READ_SUBSECT (H80040192)	:	The sub-section number is not correct.
ME_DATA_READ_DATASIZE (H80040196)	:	Data size over
ME_DATA_READ_READ (H8004019D)	:	Read disabled data
ME_DATA_READ_DATATYPE (H80040197)	:	The specified data type is not correct.

Details

Gets the data that is specified using arguments.
For "Variable-type Data", specify a variable type (including an array) other than the GMEM type.
When setting a value of 2 or greater as "Number of successive data items", specify the variable name of the array-type variable for "Obtained data".

Example

```
LONG lData(3);
LONG lStatus;
LONG lAddress;
lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
'Collectively get the common variables (#100 to #102)
lStatus = melGetLumpData(lAddress, 4, 800, 3, lData);
```

melSelectExecPrg	Search for operation program

Syntax

melSelectExecPrg(lAddress, pSelectPrg, lSequenceNum, lBlockNum);

Argument

(i)LONG lAddress : Address
Specify the following items.
- Specify NC.
- Specify the part system.

(i)STRING pSelectPrg : Program file name for which the operation search is performed

(i)LONG lSequenceNum : Sequence No. to be searched for

(i)LONG lBlockNum : Block No. to be searched for

Return value

ME_OPE_SELECTPRG_ADDR (H80001090)	:	The specified address is not correct.
ME_OPE_SELECTPRG_PRGFORMAT(H80001001)	:	The format of the program file name is not correct.
ME_OPE_SELECTPRG_NOTPRG (H80001002)	:	The specified program is not found.
ME_OPE_SELECTPRG_RUNNING (H80001003)	:	The program is currently running.
ME_OPE_SELECTPRG_RESET (H80001004)	:	In process of resetting
ME_OPE_SELECTPRG_FILESYSTEM(H80001043)	:	Some kind of error occurred in the file system.
ME_OPE_SELECTPRG_DATATYPE (H80001094)	:	The specified data type is not correct.
ME_OPE_SELECTPRG_SEARCHING(H8000100A)	:	Cannot perform search (currently carrying out another search).
ME_OPE_SELECTPRG_MODE (H80001091)	:	The specified operation mode is not correct.
ME_OPE_SELECTPRG_NOTSUPPORTED(H80001049)	:	A file of the unsupported device has been specified.

Details

Performs the operation search of the specified program.
When both "Sequence number" and "Block number" are set to 0, the operation search starts from the head of the program. If a value other than 0 is specified, the operation search starts from that position.

Example

```
LONG lStatus;
LONG lAddress;
lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
'Perform the operation search of the program (machining program
No. 100 in NC memory)
lStatus = melSelectExecPrg(lAddress, "M01:\PRG\USER\100", 0, 0);
```

7. String Operation Functions

strcat	Concatenate character strings

Syntax	strcat(strDst, strSrc);
Argument	(i/o)STRING strDst : Destination character string (i)STRING strSrc : Source character string
Return value	Null character " ": Concatenation has failed. Concatenated character string: Concatenation has succeeded.
Details	Concatenates strSrc to the end of strDst, and returns the character string to strDst.

Example	STRING strDst;
	STRING strSrc;
	STRING strResult;
	
	strDst = "XYZ";
	strSrc = "ABC";
	strResult = strcat(strDst, strSrc);
	// Returns "XYZABC" to strDst, strResult.

strcmp	Compare character strings

Syntax	strcmp(strCmp1, strCmp2);
Argument	(i)STRING strCmp1 : Comparison character string 1 (i)STRING strCmp2 : Comparison character string 2
Return value	Less than 0 : strCmp1 is less than strCmp2. 0 : strCmp2 is the same as strCmp1. Greater than 0 : strCmp1 is greater than strCmp2.
Details	Compares strCmp1 with strCmp2, and returns the comparison result.

Example	STRING strCmp1;
	STRING strCmp2;
	LONG lResult;
	
	strCmp1 = "ABC";
	strCmp2 = "DEF";
	lResult = strcmp(strCmp1, strCmp2);
	// Returns a value less than 0 to lResult.

strcpy	Copy a character string
Syntax	strcpy(strDst, strSrc);
Argument	(o)STRING strDst : Destination character string (i)STRING strSrc : Source character string
Return value	Null character " ": Copy has failed. Destination character string: Copy has succeeded.
Details	Copies strSrc to strDst.

Example	STRING strDst;
	STRING strSrc;
	STRING strResult;
	
	strSrc = "ABC";
	strResult = strcpy(strDst, strSrc);
	// Returns "ABC" to strDst, strResult.

strlen	Get the number of characters in the target character string
Syntax	strlen(strTarget);
Argument	(i)STRING strTarget: Target character string
Return value	0 or greater: Number of characters
Details	Returns the number of characters in strTarget.

Example	STRING strTarget;
	LONG lResult;
	
	strTarget = “XYZ”;
	lResult = strlen(strTarget);
	// Returns 3 to lResult.

strncat	Concatenate character strings by the specified number of characters
Syntax	strncat(strDst, strSrc, lSize);
Argument	(i/o)STRING strDst : Destination character string (i)STRING strSrc : Source character string (i)LONG lSize : Number of characters to be added
Return value	Null character " ": Concatenation has failed. Concatenated character string: Concatenation has succeeded.
Details	Adds strSrc to strDst by lSize characters.

Example	STRING strDst;
	STRING strSrc;
	STRING strResult;
	
	strDst = “XYZ”;
	strSrc = “ABC”;
	strResult = strncat(strDst, strSrc, 2);
	// Returns "XYZAB" to strDst, strResult.
	strResult = strncat(strDst, strSrc, 5);
	// Returns "XYZABC" to strDst, strResult.

strncmp	Compare character strings by the specified number of characters
Syntax	strncmp(strCmp1, strCmp2, lSize);
Argument	(i)STRING strCmp1: Target character string 1 (i)STRING strCmp2: Target character string 2 (i)LONG lSize : Number of characters to be compared
Return value	Less than 0: strCmp1 is less than strCmp2. 0 : strCmp2 is the same as strCmp1. Greater than 0: strCmp1 is greater than strCmp2.
Details	Compares character string strCmp1 with strCmp2 by lSize characters, and returns the comparison result.

Example	STRING strCmp1;
	STRING strCmp2;
	LONG lResult;
	
	strCmp1 = "ABCDEF";
	strCmp2 = "ABCABC";
	lResult = strncmp(strCmp1, strCmp2, 3);
	// Returns 0 to lResult.

strncpy	Copy a character string by the specified number of characters
Syntax	strncpy(strDst, strSrc, lSize);
Argument	(o)STRING strDst : Destination character string (i)STRING strSrc : Source character string (i)LONG lSize : Number of characters to be copied
Return value	Null character " ": Copy has failed. Destination character string: Copy has succeeded.
Details	Copies strSrc to strDst by lSize characters.

Example	STRING strDst;
	STRING strSrc;
	LONG lSize;
	STRING strResult;
	
	lSize = 2;
	strSrc = "ABC";
	strResult = strncpy(strDst, strSrc, lSize);
	// Returns "AB" to strDst, strResult.

strchr	Search for a character from the beginning of a character string
Syntax	strchr(strTarget, lSearch);
Argument	(i)STRING strTarget : Character string to be searched (i)LONG lSearch : Search character (Specified using an ASCII code.)
Return value	0 or greater : Character detection position -1 : Not detected.
Details	Searches for strTarget, and returns the position of the first detected lSearch.

Example	STRING strTarget;
	LONG lSearch;
	LONG lPosition;
	
	strTarget = “ABCDEFG”;
	lSearch = H42; // “B”
	lPosition = strchr(strTarget, lSearch);
	// Returns 1 to lPosition.

strrchr	Search for a character from the end of a character string
Syntax	strrchr(strTarget, lSearch);
Argument	(i)STRING strTarget : Character string to be searched (i)LONG lSearch : Search character (Specified using an ASCII code.)
Return value	0 or greater : Character detection position -1 : Not detected.
Details	Searches for strTarget, and returns the position of the last detected lSearch.

Example	STRING strTarget;
	LONG lSearch;
	LONG lPosition;
	
	strTarget = "ABCDEFG";
	lSearch = H43; // "C"
	lPosition = strrchr(strTarget, lSearch);
	// Returns 2 to lPosition.

strpbrk	Search for characters included in a character group from the beginning of a character string

Syntax	strpbrk(strTarget, strCharSet);
Argument	(i)STRING strTarget : Character string to be searched (i)STRING strCharSet : Search character group
Return value	0 or greater : Character detection position -1 : Not detected.
Details	Searches for strTarget, and returns the position at which any character included in the character group specified in strCharSet is first detected.

Example	STRING strTarget;
	STRING strCharSet;
	LONG lPosition;
	
	strTarget = "ABCDEF";
	strCharSet = "FC";
	lPosition = strpbrk(strTarget, strCharSet);
	// Returns 2 to lPosition.

strstr	Search for a character string from the beginning

Syntax	strstr(strTarget, strSearch);
Argument	(i)STRING strTarget : Character string to be searched (i)STRING strSearch : Search character string
Return value	0 or greater : Character detection position -1 : Not detected.
Details	Searches for strTarget, and returns the position at which the character string specified in strSearch is first detected.

Example	STRING strTarget;
	STRING strSearch;
	LONG lPosition;
	
	strTarget = "ABCDEFGHI";
	strSearch = "DEF";
	lPosition = strstr(strTarget, strSearch);
	// Returns 3 to lPosition.

strspn	Search for characters not included in a character group from the beginning of a character string
Syntax	strspn(strTarget, strCharSet);
Argument	(i)STRING strTarget : Character string to be searched (i)STRING strCharSet : Search character group
Return value	0 or greater: Position at which characters not included in the search character group are detected Returns the number of characters in strTarget if the position is not detected.
Details	Searches for strTarget, and returns the position at which characters not included in the character group specified in strCharSet are first detected.

Example	STRING strTarget;
	STRING strCharSet;
	LONG lPosition;
	
	strTarget = "AABBCCDEF";
	strCharSet = "ABC";
	lPosition = strspn(strTarget, strCharSet);
	// Returns 6 to lPosition.

strcspn	Search for characters included in a character group from the beginning of a character string
Syntax	strcspn(strTarget, strCharSet);
Argument	(i)STRING strTarget : Character string to be searched (i)STRING strCharSet : Search character group
Return value	0 or greater: Position at which characters included in the search character group are detected Returns the number of characters in strTarget if the position is not detected.
Details	Searches for strTarget, and returns the position at which any character included in the character group specified in strCharSet is detected.

Example	STRING strTarget;
	STRING strCharSet;
	LONG lPosition;
	
	strTarget = "AABBCCDEF";
	strCharSet = "FED";
	lPosition = strcspn(strTarget, strCharSet);
	// Returns 6 to lPosition.

LTOA	Convert the integer value to a character string
Syntax	LTOA(lValue, lRadix);
Argument	(i)LONG lValue : Integer value (i)LONG lRadix : Radix (2 to 16)
Return value	Null character " ": Conversion has failed. Converted character string: Conversion has succeeded.
Details	Converts the integer value specified in lValue to a character string. The conversion method can be specified using lRadix. If a value other than 2 to 16 is specified in lRadix, it is converted as 10.

Example	STRING strResult;
	LONG lValue;
	
	lValue = 123456;
	strResult = LTOA(lValue, 10);
	// Returns a decimal character string "123456" to strResult.
	strResult = LTOA(lValue, 16);
	// Returns a hexadecimal character string "1E240" to strResult.

FTOA	Convert the real value to a character string
Syntax	FTOA(dbValue, Digit);
Argument	(i)DOUBLE dbValue : Real value (i)LONG Digit : Number of decimal places (0 to 10)
Return value	Null character " ": Conversion has failed. Converted character string: Conversion has succeeded.
Details	Converts the real value specified in dbValue to a character string. Rounds off digits not specified in Digit. If the number of significant digits is less than the number of decimal places, 0s are embedded.

Example	STRING strResult;
	DOUBLE dbValue;
	
	dbValue = 123.45678;
	strResult = FTOA(dbValue, 2);
	// Returns "123.46" to strResult.
	strResult = FTOA(dbValue, 10);
	// Returns "123.4567800000" to strResult.

8. Other Functions

Sleep	Interrupt the macro execution during a specified time.
Syntax	Sleep(lMilliseconds);
Argument	(i)LONG lMilliseconds : The time to interrupt the macro execution (ms) * A negative value is handled as 0.
Return value	None
Details	Interrupt the macro execution during the specified time. * This macro is available on D5 version or later.

Example	STRING strToolNo;
	LONG lStat;
	LONG lAddress;
	LONG lTnum;
	
	' Specify the tool number to search.
	strToolNo = "10/1";
	
	' Search the tool number.
	lAddress = ADR_MACHINE(1) | ADR_SYSTEM(1);
	lStat = melSetData(lAddress, 15, 2000005, 0, strToolNo);
	WHILE (lTnum == -1) DO 1;
	' Interrupt the macro execution and execute the NC process.
	Sleep(20);
	lStat = melGetData(lAddress, 15, 2000005, 0, strToolNo);
	lTnum = ATOL(strToolNo);
	END 1;

9. Error Message List

The error messages displayed in the "Macro Edit" dialog box and description are shown below.

Error message	Error code	Detail description
Format error	1	There is description that cannot be interpreted. It is not a variable name, function name or programming language.
Variable name error	2	Error in macro reserved word variable name.
(is missing	3	Starting parenthesis "(" is missing in the function or IF statement.
No. of() does not agree	4	Number of () does not match.
Position of , is incorrect	5	A comma (,) is in an incorrect position.
Function argument error	6	Error in function argument
=command error	7	Error in substituted part such as 3=VAR_A;
End of program is incomplete	8	The program is incomplete.
IF sentence error	9	There is an error in the description of IF, ELSE or ENDIF.
, or ; is missing	10	The comma (,) after the function argument is missing. Or the program is not deliminated with the semicolon (;).
Grammar error with FOR command	11	Not paired with NEXT, BREAK or CONTINUE.
Too many FOR command(Max 8)	12	Up to eight levels of FOR statement are allowed. Contain the depth of nest within eight.
Maximum number of lines exceeded	18	Up to 10000 lines are allowed in an event.
Too many IF command(Max 15)	19	Up to 15 levels of IF statement are allowed. Contain the depth of nest within 15.
GOTO sentence error	20	- Not paired with the sequence No. - The label of GOTO destination is not written after the GOTO statement. - The label of GOTO destination is in FOR-NEXT and WHILE-END.
WHILE DO sentence error	21	Not paired with END.
Too many WHILE command	22	Up to 27 levels of WHILE statement are allowed. Contain the depth of nest within 27.
Label error	23	There is a variable name of the No. designated by GOTO statement. (SHORT N100; ... GOTO 100;)
Array designation impossible	24	An array type is specified for the data type (GMEM) that cannot be declared with an array type.
Number of array designation elements exceeded the range	25	Specify the number of array elements in the range from 1 to 100.

Error message	Error code	Detail description
Declaration of number of array elements illegal	26	The number of array elements is invalid. Specify a numerical value for the number of elements.
Range of number of array elements illegal	27	A value out of the range is specified for the number of elements. Specify the number of elements in the range from "0" to "the specified number of elements -1".
Number of array variables per line exceeded	28	Up to 20 array variables can be specified per line.
Designation type of array element illegal	29	An integer constant or a value other than variables is specified as the array element.

The probable causes of the format error are described below.

Error message	Error condition
Format error	The macro is described at other than the location between header and footer of the macro ($ButtonXXX-OnClick and $End).
	Two variables are declared at a time. Example: SHORT A,B;
	A character string starting at other than a letter ('A' to 'Z' or 'a' to 'z') or underscore ('_') is designated as a variable name.
	32 or more characters are designated as a variable name.
	The same variable name as that of variable type (reserved words) is specified. Example: CHAR CHAR;, SHORT SHORT;, etc.
	The name of the control designated as an argument of the macro command is not enclosed with double quotation marks " ".

Appendix

Appendix 1. Data Type Definitions

The definitions for various structure data types are given below.

```
#define   GColor          long            /* color variable                    */
#define   HGFONT          GFontHandle*    /* 2 font handle                     */
#define   HGDRAW          GDraw*          /* drawing handle                    */

/*****************************************************************************
 * Rectangle structure
 *****************************************************************************/
typedef struct _GRect{
        short             nXmin;          /* upper left X coordinate           */
        short             nYmin;          /* upper left Y coordinate           */
        short             nXmax;          /* lower right X coordinate          */
        short             nYmax;          /* lower right Y coordinate          */
}GRect;

/*****************************************************************************
 * Point structure
 *****************************************************************************/
typedef struct _GPoint{
        short             nX;             /* X coordinate                      */
        short             nY;             /* Y coordinate                      */
}GPoint;

/*****************************************************************************
 * Line structure
 *****************************************************************************/
typedef struct _GLine{
        short             nX1;            /* X coordinate for starting point   */
        short             nY1;            /* Y coordinate for starting point   */
        short             nX2;            /* X coordinate for ending point     */
        short             nY2;            /* Y coordinate for ending point     */
}GLine;

/*****************************************************************************
 * Polygon structure
 *****************************************************************************/
typedef struct _GPoly{
        short             nNumPoints;     /* number of vertices                */
        GPoint            *pgptPoints;    /* pointer to vertex data            */
        GRect             grBounds;       /* outline rectangle                 */
}GPoly;

/*****************************************************************************
 * Brush structure
 *****************************************************************************/
typedef struct _GBrush{
        short             nFillPattern;   /* fill pattern                      */
        GColor            gcForeColor;    /* fill foreground color             */
        GColor            gcBackColor;    /* fill background color             */
}GBrush;
```

```
/*****************************************************************************
 * Border structure
 *****************************************************************************/
typedef struct _GBorder{
        unsigned char       fBorder;            /* 3D border present or absent          */
        GColor              gcULColor;          /* upper left border color              */
        GColor              gcLRColor;          /* lower right border color             */
        GColor              gcLineColor;        /* line color                           */
        short               nSize;              /* 3D border size                       */
}GBorder;

/*****************************************************************************
 * Caption structure
 *****************************************************************************/
typedef struct _GCaption{
        GColor              gcColor;            /* caption character color              */
        unsigned char       ucHPosition;        /* horizontal display position          */
        unsigned char       ucVPosition;        /* vertical display position            */
        short               nLeftMargin;        /* left margin                          */
        short               nRightMargin;       /* right margin                         */
        short               nTopMargin;         /* top margin                           */
        short               nBottomMargin;      /* bottom margin                        */
}GCaption;

/*****************************************************************************
 * Cursor structure
 *****************************************************************************/
typedef struct _GCursor{
        unsigned char       ucType;             /* cursor type                          */
        GColor              gcColor;            /* cursor color                         */
}GCursor;

/*****************************************************************************
 * Font structure
 *****************************************************************************/
typedef struct _GFontSize{
        short               nAscent;            /* height from baseline to top line     */
        short               nDscent;            /* height from baseline to bottom line  */
        unsigned short      usWidth;            /* basic character width                */
        unsigned short      usHeight;           /* character height                     */
        unsigned short      usMaxWidth;         /* maximum character width              */
}GFontSize;
```

```
/*****************************************************************************
 * Simple font structure
 *****************************************************************************/
typedef struct _GSimpleFont{
        GTCHAR          szFontName[MAX_FONTNAME_LEN+1];   /* font name          */
        unsigned char   ucSize;                 /* font size                    */
        unsigned char   gptXScale;              /* horizontal scale             */
        unsigned char   gptYScale;              /* vertical scale               */
        unsigned char   ucWeight;               /* thickness                    */
        unsigned char   ucStyle;                /* shape                        */
}GSimpleFont;

typedef struct _GFont{
        short           nID;                    /* font ID                      */
        GTCHAR          szFontName[MAX_FONTNAME_LEN+1];   /* font name          */
        unsigned short  usCharacterSet;         /* character code               */
        short           nFontSizeCount;         /* number of font sizes         */
        GFontSize       *gfsSize;               /* pointer to font size array   */
        short           nFixedWidth;            /* fixed width font information */
        long            lFontSupport;         /* forms supported by the font (italics, bold, etc.) */
}GFont;

typedef struct _GFontAttribute{
        unsigned short  usWidth;                /* character width              */
        unsigned short  usHeight;               /* character height             */
        unsigned short  usWeight;               /* character thickness          */
        unsigned short  usItalic;               /* character italics            */
        unsigned short  usOutline;              /* character border             */
        void            *pData;                 /* additional information       */
        unsigned char   ucXScale;               /* horizontal scale             */
        unsigned char   ucYScale;               /* vertical scale               */
}GFontAttribute;

typedef struct _GFontPattern{
        GTCHAR          *pcChar;                /* pointer to character data    */
        GTCHAR          *pcNextChar;            /* pointer to next character data */
        short           nWidth;                 /* width                        */
        short           nHeight;                /* height                       */
        short           nBpp;                   /* number of dots per pixel     */
        char            *pcPattern;             /* pointer to pattern           */
        short           nGetSize;               /* get size flag                */
}GFontPattern;

typedef struct _GFontHandle{
        short           nID;                    /* font ID                      */
        short           nAttributeType;         /* use either pgfaAttribute or  */
                                                /* pnAttributeArray as attribute */
        GFontAttribute  *pgfaAttribute;         /* font attribute               */
        short           *pnAttributeArray;      /* font attribute (array)       */
}GFontHandle;
```

```
/*****************************************************************************
 * Image data structure
 *****************************************************************************/
typedef struct _GRFHeader{
        short           nType;          /* image type                   */
        short           nWidth;         /* width                        */
        short           nHeight;        /* height                       */
        short           nBpp;           /* number of bits per pixel     */
        long            lSize;          /* data size                    */
        unsigned char   *pData;         /* pointer to real data         */
}GRFHeader;
typedef struct _GRFHeaderDIB{
        short           nType;          /* image type                   */
        short           nWidth;         /* width                        */
        short           nHeight;        /* height                       */
        short           nBpp;           /* number of bits per pixel     */
        long            lSize;          /* data size                    */
        unsigned char   *pData;         /* pointer to real data         */
        unsigned char   *pPalette;      /* pointer to palette data      */
}GRFHeaderDIB;

/* image structure */
typedef struct _GImage{
        GRFHeader       *pImage;        /* pointer to image data structure  */
}GImage;


/*****************************************************************************
 * System time structure
 *****************************************************************************/
typedef struct _GSystemTime{
        unsigned long   ulLTime;        /* lower-side 32 bits for system time   */
        unsigned long   ulUTime;        /* upper-side 32 bits for system time   */
}GSystemTime;
```

```
/*****************************************************************************
 * Drawing environment structure
 *****************************************************************************/
/* GDraw structure */
typedef struct _GDraw{
        GRect           grLocalRect;        /* physical coordinate area                          */
        GPoint          gptLocalOrigin;     /* position of origin on physical coordinates        */
        GRect           grVirtualRect;      /* virtual coordinate area                           */
        GRect           grClipRect;         /* clipping rectangle                                */
        GPoint          gptPenPosition;     /* current position                                  */
        GColor          gcPenColor;         /* line color                                        */
        GPoint          gptPenSize;         /* line thickness                                    */
        short           nPenCap;            /* line end shape                                    */
        short           nPenJoin;           /* line contact shape                                */
        short           nPenDash;           /* line type No.                                     */
        short           nDashOffset;        /* line pattern offset                               */
        short           nTextMode;          /* text mode                                         */
        GColor          gcForeColor;        /* fill foreground color                             */
        GColor          gcBackColor;        /* fill background color                             */
        short           nFillPattern;       /* fill pattern No.                                  */
        short           nDrawCondition;     /* drawing condition                                 */
        short           nDrawingMode;       /* raster operation                                  */
        HGVRAM          hSystemVram;        /* display destination VRAM                          */
        HGVRAM          hDrawVram;          /* drawing destination VRAM                          */
        HGFONT          hFont;              /* font                                              */
        void            *vgdftDraw;         /* function table for drawing                        */
        char            cDashPatterns[NUM_DASH_PATS][DASH_PAT_SIZE];
                                            /* line pattern                                      */
        char            cFillPatterns[NUM_FILL_PATS][FILL_PAT_SIZE];
                                            /* fill pattern                                      */
        GDrawParamPoly  gpPoly;             /* polygon drawing parameter                         */
        GDrawParamWideLine  gpwWideLine;    /* wide line drawing parameter                       */
        GDrawParamOval  gpwOval;            /* circle, arc and sector drawing parameter          */
        unsigned short  usErrorCode;        /* previous error code                               */
        unsigned char   ucGradationType;    /* gradation type                                    */
                                            /* (0: up to down, 1: left to right)                 */
        GColor          gcGradationColor1;  /* color1                                            */
        GColor          gcGradationColor2;  /* color2                                            */
        unsigned short  usVertexPos;        /* gradation vertex position (0 to 100)              */
        unsigned short  usGradationLevel;   /* gradation level (0 to 256)                        */
        unsigned char   ucColorMode;        /* actual VRAM color
                                               environment information                        */
        GColor      gcRedMask;              /* direct color R value mask                         */
        GColor      gcGreenMask;            /* direct color G value mask                         */
        GColor      gcBlueMask;             /* direct color B value mask                         */
        char        cRedShift; /            * direct color R value shift value                  */
        char        cGreenShift;            /* direct color G value shift value                  */
        char        cBlueShift;             /* direct color B value shift value                  */
        GColor      gcForeColorOrg;         /* foreground color (original)                       */
        GColor      gcBackColorOrg;         /* background color (original)                       */
        GColor      gcPenColorOrg;          /* background color (original)                       */

        unsigned long ulExParam;            /* extension parameter                               */
        short       nExParam;               /* extension parameter                               */
}GDraw;
```

```
/*****************************************************************************
 * Memory management structure
 *****************************************************************************/
typedef struct _GMemory{
        unsigned char       fUseSpace;      /* shows use (0)/not use (1) status for memory space */
        unsigned char       cReserve[3];    /* reserved space (for 4 byte environment adjustment) */
        unsigned long       ulSize;                     /* memory space size                    */
        struct _GMemory     *pvPrevMemorySpace;
                                            /* pointer to GMemory in previous memory space */
        struct _GMemory     *pvSmallMemorySpace;/* pointer to space GMemory in small        */
                                            /* memory space                                 */
        struct _GMemory     *pvLargeMemorySpace;/* pointer to space GMemory in large        */
                                            /* memory space                                 */
}GMemory;

typedef struct _GMemorySpaceInformation{
        unsigned char       ucType;                     /* memory space type                */
        unsigned char       ucPlane;                    /* memory space plane No.           */
        unsigned char       cReserve[2];    /* reserved space (for 4 byte environment adjustment) */
        char                *pvMemorySpace;         /* pointer to memory space allocated by user */
        unsigned long       ulMemorySpaceSize;
                                    /* size of memory space allocated by user (multiples of 32) */
        GMemory             *pgmNoUseMemoryTree;
                                        /* pointer to unused two-branch memory management */
}GMemorySpaceInformation;
```

```
/*****************************************************************************
 * Control related structures
 *****************************************************************************/
/* Design structure */
typedef struct GDesign{
        GBrush          gbBrush;        /* fill brush                   */
        unsigned short  usImageID;      /* image resource ID            */
}GDesign;

/* Focus movement structure */
typedef struct GFocusObject{
        unsigned short  usKeyCode;      /* virtual key code             */
        unsigned short  usType;         /* focus movement method        */
        unsigned short  usID;           /* ID of object being moved     */
}GFocusObject;

/* Focus movement structure */
typedef struct GFocusInformation{
        unsigned short  usCount;        /* number of focus settings     */
        GFocusObject    *pFocusArray;   /* focus setting (array)        */
}GFocusInformation;

/* Value structure */
typedef union GValue{
        short           nValue;         /* short value                  */
        unsigned short  usValue;        /* unsigned short value         */
        long            lValue;         /* long value                   */
        unsigned long   ulValue;        /* unsigned long value          */
        float           fValue;         /* float value                  */
}GValue;

/* GBaseWindow Export/Import structure */
typedef struct GBaseWindowProperty{
        unsigned short  usType;         /* object type                  */
        unsigned short  usID;           /* object ID                    */
        short           nX;             /* X coordinate                 */
        short           nY;             /* Y coordinate                 */
        short           nWidth;         /* width                        */
        short           nHeight;        /* height                       */
        unsigned long   ulStyle;        /* object shape                 */
        GFocusInformation *pFocusInfo;  /* focus object                 */
}GBaseWindowProperty;
```

```
/*****************************************************************************
 * NC data access-related structure
 *****************************************************************************/
/* NC information structure */
typedef struct GNCControl{
        long            lMachine;                       /* NC No.                                                   */
                                                        /* Setting range: 1 to 255                                  */
        long            lSystem;                        /* Part system number                                       */
                                                        /* Setting range: 0 to 10                                   */
        long            lGround;        /* Ground                                                                   */
                                        /* 0 : Basic part system / Foreground                                       */
                                        /* 1 : Basic part system / Background                                       */
                                        /* 2 : Current part system during cross control / Foreground                */
                                        /* 3 : Current part system during cross control / Background                */
        unsigned long       ulAxis;                     /* Axis number                                              */
                                                        /* Setting range: 0 to 16                                   */
}GNCControl;


/* NC data structure */
typedef union GNCValue{
        char                cValue;                     /* One-byte integer value                                   */
        unsigned char       ucValue;                    /* Unsigned one-byte integer value                          */
        short               nValue;                     /* Two-byte integer value                                   */
        unsigned short      usValue;                    /* Unsigned two-byte integer value                          */
        long                lValue;                     /* Four-byte integer value                                  */
        unsigned long       ulValue;                    /* Unsigned four-byte integer value                         */
        double              dValue;                     /* Real number value                                        */
}GNCValue;
```

Appendix 2. Macro Event Arguments

Arguments of some macro events are passed on execution.

The arguments are stored in LLPARAM and LUPARAM, and the contents are different depending on the macro event.

Macro event	LLPARAM	LUPARAM
OnKeyPress	Key code	Auxiliary key code
OnKeyRelease	Key code	Auxiliary key code
OnPress	-	XY coordinates
OnRelease	-	XY coordinates
OnClick	-	-
OnDraw	(Reserve)	-
OnTimer	(Reserve)	-
OnSetFocus	-	-
OnKillFocus	-	-
OnCreate	-	-
OnDelete	-	-
OnScroll	Scroll method	Scroll position
OnScrollFinish	-	-
OnSelectChange	Selected line	-

Appendix 2.1 OnKeyPress/OnKeyRelease Arguments

The following list shows the key entry, the key code stored in LLPARAM, and the auxiliary key code stored in LUPARAM.

Key entry	Key code	Auxiliary key code
A	65	-
B	66	-
C	67	-
D	68	-
E	69	-
F	70	-
G	71	-
H	72	-
I	73	-
J	74	-
K	75	-
L	76	-
M	77	-
N	78	-
O	79	-
P	80	-
Q	81	-
R	82	-
S	83	-
T	84	-
U	85	-
V	86	-
W	87	-
X	88	-
Y	89	-
Z	90	-
[	219	-
]	221	-
(	56	BIT0
)	57	BIT0
MONITOR	112	BIT0
SET UP	113	BIT0
Edit	114	BIT0
DIAGN	115	BIT0
MAINTE	116	BIT0
Tab changeover left	120	BIT1
Tab changeover right	121	BIT1
Part system changeover	112	BIT1
SFP	120	BIT0
F0	121	BIT0
LIST	119	BIT1
Window display	114	BIT1
Window changeover	115	BIT1
F1	112	
F2	113	-
F3	114	-
F4	115	-
F5	116	-
F6	117	-
F7	118	-

Key entry	Key code	Auxiliary key code
0	48	-
1	49	-
2	50	-
3	51	-
4	52	-
5	53	-
6	54	-
7	55	-
8	56	-
9	57	-
-	189	-
+	107	-
/	191	-
*	186	BIT0
.	190	-
@	192	-
_	226	BIT0
,	188	-
!	49	BIT0
:	186	-
\	220	-
<	188	BIT0
>	190	BIT0
PAGE UP	33	-
PAGE DOWN	34	-
←	37	-
↑	38	-
→	39	-
↓	40	-
|←	9	BIT0
→|	9	-
ALTER	-	-
CTRL	-	-
SHIFT	-	-
SP	32	-
#	51	BIT0
$	52	BIT0
CB	36	BIT0
CAN	27	-
INS	45	-
DEL	46	-
=	189	BIT0
~	222	BIT0
EOB(;)	187	-
INPUT	13	-
RESET	-	-

Key entry	Key code	Auxiliary key code
F8	119	-
F9	120	-
F10	121	-
CANCEL	122	-
Page feed	123	-

Key entry	Key code	Auxiliary key code

NOTE

- The list above describes the key code that can be entered with ABC array NC keyboard.
- F1 to 10, cancel, and page feed keys correspond to the buttons at the bottom of the display unit.
- A logical sum with the auxiliary key code stored in LUPARAM has the following meaning (key entry).

Auxiliary key code	Value	Meaning
BIT0	0x01	SHIFT
BIT1	0x02	CONTROL
BIT2	0x04	ALTER

- For LUPARAM, BIT0 is turned ON by entering Shift and BIT1 is turned ON by entering Ctrl.
  (Note) Shift etc. indicated here are not the key names (buttons) on the NC keyboard, but indicated the ones by the key entry.
  Ex.) When the MONITOR key is pressed, 112 is stored in LLPARAM and BIT0 for LUPARAM is turned ON.

Appendix 2.2 OnPress/OnRelease Arguments

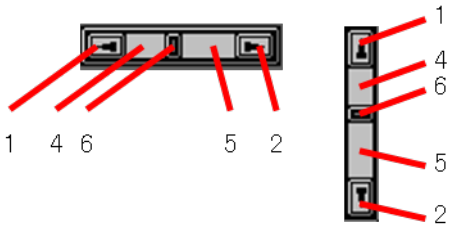
The mouse status stored in LLPARAM is always "0".

As for the XY coordinates stored in LUPARAM, the upper two bytes are the Y coordinate and the lower two bytes are the X coordinate. They can be acquired by the following macro code.

Item	Acquisition
X coordinate	LUPARAM & H0000FFFF
Y coordinate	(LUPARAM & HFFFF0000) >> 16

Appendix 2.3 OnScroll Arguments

The scroll method stored in LLPARAM shows where in the scroll bar is used to scroll. The scroll methods are as follows.



Value	Scroll method
1	Left-most, top arrow
2	Right-most, bottom arrow
4	Left, upper background
5	Right, lower background
6	Pinch

The scroll position stored in LUPARAM shows the top line on the page after the scroll.

Revision History

Date of revision	Manual No.	Revision details
Jul. 2017	IB(NA)1501500-A	First edition created.
Feb. 2018	IB(NA)1501500-B	• Corresponded to NC Designer2 Ver.A4. • The following sections were changed. Precautions for Safety ・Items in "Caution" were added. 5.22 CycleTime ・"9: Cut time" was added. 5.29 NCAlarmList • The following functions were added. GCSNCAlarmListGetSelectLine GCSNCAlarmListGetAlarmInfo
Oct. 2018	IB(NA)1501500-C	• Corresponded to NC Designer2 Ver.A5. • The following sections were added. 8. Other Functions • The following sections were changed. Introduction ・E80 was added to supported models 5.2 Control Common Command ・Changed "Details" and "Example" of GCSCreateGWindow 5.21 Counter ・Added "26: PLC axis position" to "Details" of GCSCounterSetCounterType
Sep. 2020	IB(NA)1501500-D	• Corresponded to NC Designer2 Ver.A6. • The following section was added. 5.30 Extension Menu • The following sections were changed. 5.14 NCPLCButton 5.15 NCPLCTextbox 5.29 AlarmList • Mistakes were corrected.
Oct. 2021	IB(NA)1501500-E	• Corresponded to NC Designer2 Ver.A7. • The following sections were changed. 5.15 NCPLCTextbox 5.26 InputBox • Mistakes were corrected.
Jul. 2022	IB(NA)1501500-F	• Corresponded to NC Designer2 Ver.A8. • The following sections were changed. Introduction (Continue to the next page)

Date of revision	Manual No.	Revision details
		(Continued) 4.3 Programming Language 5.15 NCPLCTextbox 5.18 FileInOut 6. NC Data Access Function • Mistakes were corrected.
Mar. 2023	IB(NA)1501500-G	• Corresponded to NC Designer2 Ver.A9. • The following sections were added. Appendix 2. Macro Event Arguments Appendix 2.2 OnPress/OnRelease Arguments Appendix 2.3 OnScroll Arguments • The following sections were changed. 4.1 Macro Editing Area ・NOTE was added. 4.3 Programming Language ・The number of nesting levels of conditional branch was changed from 8 to 15. 5.12 ScrollBar ・The following functions were added. GCSScrollbarexSetScrollPosition GCSScrollbarexGetScrollPosition 9. Error Message List ・Error messages were added. Appendix 2.1 OnKeyPress/OnKeyRelease Arguments ・The section No. and the title were changed. • Mistakes were corrected.

Global Service Network

AMERICA

MITSUBISHI ELECTRIC AUTOMATION INC. (AMERICA FA CENTER)
Central Region Service Center (Chicago)
500 CORPORATE WOODS PARKWAY, VERNON HILLS, ILLINOIS 60061, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-847-478-2650
Minneapolis, MN Service Satellite
Detroit, MI Service Satellite
Grand Rapids, MI Service Satellite
Milwaukee, WI Service Satellite
Cleveland, OH Service Satellite
Indianapolis, IN Service Satellite
St. Louis, MO Service Satellite

South/East Region Service Center (Georgia)
1845 SATELLITE BOULEVARD STE. 450, DULUTH, GEORGIA 30097, U.S.A.
TEL +1-678-258-4529 / FAX +1-678-258-4519
Charleston, SC Service Satellite
Charlotte, NC Service Satellite
Raleigh, NC Service Satellite
Dallas, TX Service Satellite
Houston, TX Service Satellite
Hartford, CT Service Satellite
Knoxville, TN Service Satellite
Nashville, TN Service Satellite
Baltimore, MD Service Satellite
Pittsburg, PA Service Satellite
Tampa, FL Service Satellite
Syracuse, NY Service Satellite
Orlando, FL Service Satellite
Lafayette, LA Service Satellite
Philadelphia, PA Service Satellite

Western Region Service Center (California)
5900-B KATELLA AVE. - 5900-A KATELLA AVE. CYPRESS, CALIFORNIA 90630, U.S.A.
TEL: +1-714-699-2625 / FAX: +1-847-478-2650
San Jose, CA Service Satellite
Seattle, WA Service Satellite
Denver, CO Service Satellite

Canada Region Service Center (Toronto)
4299 14TH AVENUE MARKHAM, ONTARIO L3R OJ2, CANADA
TEL: +1-905-475-7728 / FAX: +1-905-475-7935
Edmonton, AB Service Satellite
Montreal, QC Service Satellite

Mexico Region Service Center (Queretaro)
Parque Tecnológico Innovación Querétaro, Lateral Carretera Estatal 431, Km 2+200, Lote 91 Modulos 1 y 2
Hacienda la Machorra, CP 76246, El Marqués, Querétaro, México
TEL: +52-442-153-6050
Monterrey, NL Service Satellite
Mexico City, DF Service Satellite

BRAZIL

MITSUBISHI ELECTRIC DO BRASIL COMÉRCIO E SERVIÇOS LTDA.
Votorantim Office
AV. GISELE CONSTANTINO,1578, PARQUE BELA VISTA, VOTORANTIM-SP, BRAZIL CEP:18.110-650
TEL: +55-15-3023-9000
Blumenau, Santa Catarina Office

EUROPE

MITSUBISHI ELECTRIC EUROPE B.V.
European Service Headquarters (Dusseldorf, GERMANY)
Mitsubishi-Electric-Platz 1 40882 RATINGEN, GERMANY
TEL: +49-2102-486-5000 / FAX: +49-2102-486-5910

South Germany Service Center (Stuttgart)
SCHELMENWASENSTRASSE 16-20, 70567 STUTTGART, GERMANY
TEL: + 49-711-770598-0 / FAX: +49-711-770598-141

France Service Center (Paris)
2 RUE DE L'UNION, 92565 RUEIL-MALMAISON CEDEX, FRANCE
TEL: +33-1-41-02-83-13 / FAX: +33-1-49-01-07-25

France Service Satellite (Lyon)
240, ALLEE JACQUES MONOD 69800 SAINT PRIEST FRANCE
TEL: +33-1-41-02-83-13 / FAX: +33-1-49-01-07-25

Italy Service Center (Milan)
VIA ENERGY PARK 14, VIMERCATE 20871 (MB) ITALY
TEL: +39-039-6053-342 / FAX: +39-039-6053-206

Italy Service Satellite (Padova)
VIA G. SAVELLI, 24 - 35129 PADOVA, ITALY
TEL: +39-039-6053-342 / FAX: +39-039-6053-206

U.K. Service Center
TRAVELLERS LANE, HATFIELD, HERTFORDSHIRE, AL10 8XB, U.K.
TEL: +44-1707-288-780 / FAX: +44-1707-278-695

Spain Service Center
CTRA. RUBI, 76-80 8174 SAINT CUGAT DEL VALLES, BARCELONA, SPAIN
TEL: +34-935-65-2236 / FAX: +34-935-89-1579

Poland Service Center
UL.KRAKOWSKA 50, 32-083 BALICE, POLAND
TEL: +48-12-347-6500 / FAX: +48-12-630-4701

Hungary Service Center
BUDAÖRS OFFICE PARK, SZABADSÁG ÚT 117., 2040 BUDAÖRS, HUNGARY
TEL: +48-12-347-6500 / FAX: +48-12-630-4701

Turkey Service Center
MITSUBISHI ELECTRIC TURKEY ELEKTRİK ÜRÜNLERİ A.Ş
SERIFALI MAHALLESI KALE SOKAK. NO.41 34775
UMRANIYE, ISTANBUL, TURKEY
TEL: +90-216-969-2500 / FAX: +90-216-661-44-47

Czech Republic Service Center
AutoCont Control Systems s.r.o (Service Partner)
KAFKOVA 1853/3, 702 00 OSTRAVA 2, CZECH REPUBLIC
TEL: +420-59-5691-185 / FAX: +420-59-5691-199

Russia Service Center
MITSUBISHI ELECTRIC RUSSIA LLC
LETNIKOVSKAYA STREET 2, BLD.1, 5TH 115114 MOSCOW, RUSSIA
TEL: +7-495-721-2070 / FAX: +7-495-721-2071

Sweden Service Center
HAMMARBACKEN 14, P.O.BOX 750 SE-19127, SOLLENTUNA, SWEDEN
TEL: +46-8-6251200 / FAX: +46-8-6251014

Bulgaria Service Center
AKHNATON Ltd. (Service Partner)
4 ANDREJ LJAPCHEV BLVD. POB 21, BG-1756 SOFIA, BULGARIA
TEL: +359-2-8176009 / FAX: +359-2-9744061

Ukraine Service Center (Kiev)
CSC Automation Ltd. (Service Partner)
4 B, YEVHENA SVERSTYUKA STR., 02002 KIEV, UKRAINE
TEL: +380-44-494-3344 / FAX: +380-44-494-3366

Belarus Service Center
TECHNIKON Ltd. (Service Partner)
NEZAVISIMOSTI PR.177, 220125 MINSK, BELARUS
TEL: +375-17-393-1177 / FAX: +375-17-393-0081

South Africa Service Center
Adroit Technologies (Service Partner)
20 WATERFORD OFFICE PARK, WATERFORD DRIVE, CNR OF WITKOPPEN ROAD,
FOURWAYS JOHANNESBURG SOUTH AFRICA
TEL: +27-11-658-8100 / FAX: +27-11-658-8101

ASEAN

MITSUBISHI ELECTRIC ASIA PTE. LTD. (ASEAN FA CENTER)
Singapore Service Center
307 ALEXANDRA ROAD MITSUBISHI ELECTRIC BUILDING SINGAPORE 159943
TEL: +65-6473-2308 / FAX: +65-6476-7439

PHILIPPINES

MELCO FACTORY AUTOMATION PHILIPPINES INC.
Head Office
128 LOPEZ RIZAL STREET, BRGY., HIGHWAY HILLS, MANDALUYONG CITY , MM PHILIPPINES 1550
TEL: +63-2-8256-8042 / FAX: +632-8637-2294

Philippines Service Center
KM.23 WEST SERVICE ROAD SSH, CUPANG ,MUNTINLUPA CITY, PHILIPPINES
TEL: +63-2-8807-0420 / FAX: +63-2-8842-5202

VIETNAM

MITSUBISHI ELECTRIC VIETNAM CO., LTD.
Vietnam Ho Chi Minh Service Center
11TH & 12TH FLOOR, VIETTEL TOWER B, 285 CACH MANG THANG 8 STREET, WARD 12, DISTRICT 10,
HO CHI MINH CITY, VIETNAM
TEL: +84-28-3910-5945 / FAX: +84-28-3910-5947

Vietnam Hanoi Service Center
14TH FLOOR, CAPITAL TOWER, 109 TRAN HUNG DAO STREET, CUA NAM WARD,
HOAN KIEM DISTRICT, HA NOI CITY, VIETNAM
TEL: +84-24-3937-8075 / FAX: +84-24-3937-8076

INDONESIA

PT. MITSUBISHI ELECTRIC INDONESIA
Indonesia Service Center (Cikarang)
JL. KENARI RAYA BLOK G2-07A, DELTA SILICON 5, LIPPO CIKARANG - BEKASI 17550, INDONESIA
TEL: +62-21-2961-7797 / FAX: +62-21-2961-7794

MALAYSIA

MITSUBISHI ELECTRIC SALES MALAYSIA SDN. BHD.
Malaysia Service Center (Kuala Lumpur Service Center)
LOT 11, JALAN 219, P.O BOX 1036, 46860 PETALING JAYA, SELANGOR DARUL EHSAN, MALAYSIA
TEL: +60-3-7626-5032
Johor Bahru Service Satellite
Pulau Pinang Service Satellite

THAILAND

MITSUBISHI ELECTRIC FACTORY AUTOMATION (THAILAND) CO., LTD.
Thailand Service Center (Bangkok)
101, TRUE DIGITAL PARK OFFICE, 5TH FLOOR, SUKHUMVIT ROAD, BANGCHAK, PHRA KHANONG,
BANGKOK, 10260 THAILAND
TEL: +66-2-092-8600 / FAX: +66-2-043-1231-33

INDIA

MITSUBISHI ELECTRIC INDIA PVT., LTD.
CNC Technical Center (Bangalore)
PLOT NO. 56, 4TH MAIN ROAD, PEENYA PHASE 3,
PEENYA INDUSTRIAL AREA, BANGALORE 560058, KARNATAKA, INDIA
TEL : +91-80-4655-2121
Chennai Service Satellite
Coimbatore Service Satellite
Hyderabad Service Satellite

North India Service Center (Gurgaon)
PLOT 517, GROUND FLOOR, UDYOG VIHAR PHASE-III, GURUGRAM 122008, HARYANA, INDIA
TEL : +91-124-463-0300
Ludhiana Service Satellite
Panthnagar Service Satellite
Delhi Service Satellite
Jamshedpur Service Satellite
Manesar Service Satellite

West India Service Center (Pune)
ICC-Devi GAURAV TECHNOLOGY PARK, UNIT NO.402, FOURTH FLOOR, NORTH WING,
SURVEY NUMBER 191-192 (P), NEXT to INDIAN CARD CLOTHING COMPANY Ltd,
OPP. VALLABH NAGAR, PIMPRI, PUNE- 411 018, MAHARASHTRA, INDIA
TEL : +91-20-6819-2274
Kolhapur Service Satellite
Aurangabad Service Satellite
Mumbai Service Satellite

West India Service Center (Ahmedabad)
204-209, 2ND FLOOR, 31FIVE, CORPORATE ROAD PRAHLADNAGAR,
AHMEDABAD -380015, GUJARAT, INDIA
TEL : + 91-79-6777-7888
Rajkot Service Satellite

CHINA

MITSUBISHI ELECTRIC AUTOMATION (CHINA) LTD. (CHINA FA CENTER)
CNC Call Center
TEL: +86-400-921-5130

Shanghai Service Center
NO.1386 HONG QIAO ROAD, CHANG NING QU, SHANGHAI 200336, CHINA
TEL: +86-21-2322-3030 / FAX: +86-21-2322-3000*8422
Qingdao Service Center
Suzhou Service Center
Wuhan Service Center
Ningbo Service Center
Hefei Service Center
Beijing Service Center
Tianjin Service Center
Xian Service Center
Dalian Service Center
Chengdu Service Center

Shenzhen Service Center
LEVEL8, GALAXY WORLD TOWER B, 1 YABAO ROAD, LONGGANG DISTRICT,
SHENZHEN 518129, CHINA
TEL: +86-755-2399-8272 / FAX: +86-755-8229-3686
Dongguan Service Center
Xiamen Service Center

KOREA

MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD. (KOREA FA CENTER)
Korea Service Center
8F GANGSEO HANGANG XI-TOWER A, 401 YANGCHEON-RO, GANGSEO-GU,
SEOUL 07528 KOREA
TEL: +82-2-3660-9631 / FAX: +82-2-3664-8668
Korea Daegu Service Satellite

TAIWAN

MITSUBISHI ELECTRIC TAIWAN CO., LTD. (TAIWAN FA CENTER)
Taiwan Taichung Service Center
NO. 8-1, GONGYEQU 16th RD., XITUN DIST., TAICHUNG CITY 40768 , TAIWAN
TEL: +886-4-2359-0688 / FAX: +886-4-2359-0689

Taiwan Taipei Service Center
11F, NO.88, SEC.6, ZHONGSHAN N. RD., SHILIN DIST., TAIPEI CITY 11155, TAIWAN
TEL: +886-2-2833-5430 / FAX: +886-2-2833-5433

Taiwan Tainan Service Center
11F.-1, NO.30, ZHONGZHENG S. RD., YONGKANG DIST., TAINAN CITY 71067, TAIWAN
TEL: +886-6-252-5030 / FAX: +886-6-252-5031

OCEANIA

MITSUBISHI ELECTRIC AUSTRALIA PTY. LTD.
Oceania Service Center
348 VICTORIA ROAD, RYDALMERE, N.S.W. 2116 AUSTRALIA
TEL: +61-2-9684-7269/ FAX: +61-2-9684-7245

Notice

Every effort has been made to keep up with software and hardware revisions in the contents described in this manual. However, please understand that in some unavoidable cases simultaneous revision is not possible.

Please contact your Mitsubishi Electric dealer with any questions or comments regarding the use of this product.

Duplication Prohibited

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : TOKYO BLDG.,2-7-3 MARUNOUCHI,CHIYODA-KU,TOKYO 100-8310,JAPAN

MODEL	NC Designer2
MODEL CODE	100-617
Manual No.	IB-1501500

IB(NA)1501500(ENG)-G(2303)MEE
Made in JAPAN