MITSUBISHI ELECTRIC

Changes for the Better

MITSUBISHI CNC

MELSOFT

Instruction Manual

NC Designer

MELSOFT
Integrated FA Software

IB-1500040(ENG)-G

Introduction

This instruction manual describes how to use NC Designer. Incorrect handling may lead to unforeseen accidents, so make sure to read this instruction manual thoroughly before operation to ensure correct usage.
NC Designer supports the following NC series.

Appropriate NC
M700VW/M700VS/M70V/M700/M70/E70 series

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.

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.

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.
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.
Wind River Systems, Inc.® and Tornado® are either trademarks or registered trademarks of Wind River Systems, Inc. in the United States and/or other countries.
Intel® and Pentium® are either trademarks or registered trademarks of Intel Corporation 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.

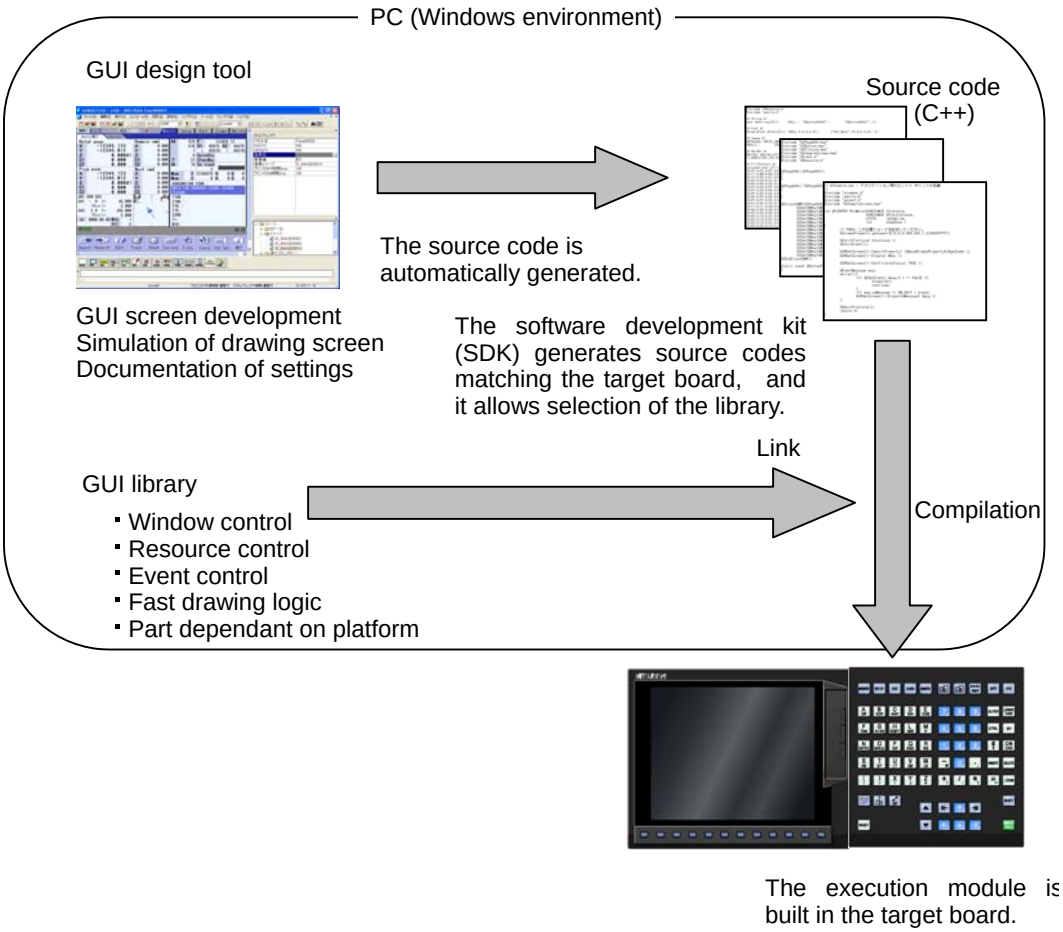
CONTENTS

1. Outline

This section describes an outline of NC Designer.

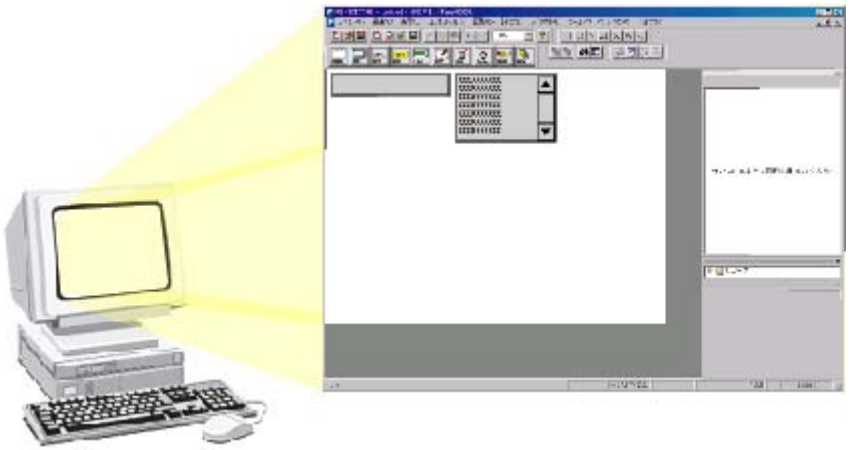
1.1 What Is NC Designer?

NC Designer provides you with a GUI development environment consisting of a GUI design tool for generating the source code of host equipment on the drawn GUI screen and a GUI library that does not depend on specific platforms.
The GUI library is compatible with every environment such as Windows CE.
NC Designer consists of a GUI design tool, which substantially reduces GUI development work-hours, and a fast and compact graphic library.



1.1.1 GUI Design Tool

With NC Designer, figures and parts having various functions are laid out on the screen during creation of screen data. Source codes (C++) matching the GUI library are automatically generated according to the created screen data.
NC Designer has various functions such as the simulation function for the PC, so that the development process which required many work-hours with conventional development methods are now automated, and re-working after assembly in the actual machine is minimized.



1.1.2 What Is the GUI Library?

The GUI library strongly supports development of the graphic user interface.
The GUI library contains functions for mouse and key events and window system, which are indispensable for the configuration of the GUI, as well as the drawing function.

1.1.3 Interpreter Method and Compilation Method

The screen development method includes two types: interpreter method (C++ language is not needed) and compilation method (C++ language is needed). The intended purposes etc. of these methods are the table below.

	Interpreter method	Compilation method
Purpose	Development of comparatively simple screen	Development of screen with more complex control operation
Programming	Unnecessary (with Macro function)	C++ language programming
Development (compilation) environment	Unnecessary	The compiler is necessary
Execution speed	Slower than the compilation method	Faster than the interpreter method

2. Features and Specifications of NC Designer

In this section, what can be done with NC designer is described for those who operate NC Designer for the first time. Specifications and functions are referred to in the description.

2.1 Features of NC Designer

NC Designer has the following features.

Ultimate GUI development framework applicable to various embedded systems (platforms)

- Automatic generation of source code
  Various source codes are automatically generated from the data created with NC Designer.
- Possible to load various fonts
- GUI library which does not depend on the platform
  The GUI library allows you to match every platform through repetitive generation.

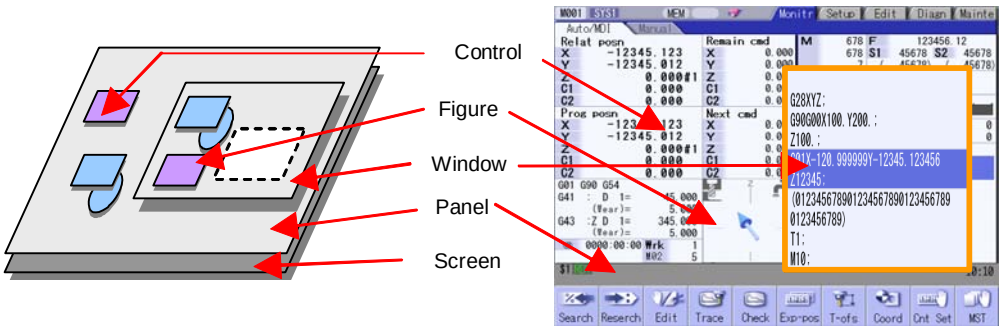
Substantial improvement of GUI development efficiency through embedded system

- Powerful editing functions (cut, copy, paste, rotation, alignment, zoom, etc.)
- Automatic generation of development document
  NC Designer automatically creates documents such as the screen list and property data in a file (rich text format).
- Simulation function for PC
  Screens created with NC Designer can be simulated.
  → Simple GUI simulation such as screen switching is realized.
  → Using the simulation tool, you can change properties of the GUI part.
  (Modification event history is also supported.)

2.2 Specifications of NC Designer

2.2.1 Screen Elements of NC Designer

The screen elements displayed (that is, created) with NC Designer include the followings.



Screen element		Description
Screen		Physical hardware display area. Available only one screen for each system. That is, only one screen for each project.
Page		Screen. The page includes the panel and the window.
	Panel	Screen displayed in full size on the screen.
	Window	Screen displayed in a window state on the screen.
Object		The object is a screen element arranged on the page. The object includes the control, figure and view frame.
	Control	The control is a group of GUI functions including buttons and pictures.
	Figure	The figure can be drawn with the basic drawing function of the GUI library. The figure includes rectangles, circles, lines, continuous lines, polygons, sectors and arcs.
	View frame	The view frame is a display area in the page having multiple pages. Controls and figures can be arranged on each page of the view frame.

2.2.2 Control

The control is an object having the following functions.

- Expression of GUI operation and retention of operation state
- Notification of GUI operation to user program

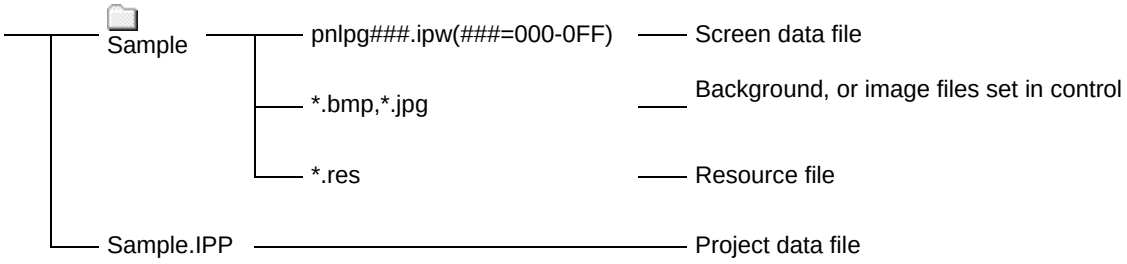
The control includes the following variations.

Name	Outline
Basic control	No function is provided for the basic control. The user creates display and event processes.
Button	The button realizes ON/OFF actions (changes in internal status) and switches the display according to the status.
Check box	The check box allows the user to select an item and retain the selected state.
Edit control	The edit control displays, inserts or overwrites character strings in multiple lines. Search and replace of a character string are also possible.
HTML browser	The HTML browser displays HTML files.
Label	The label displays a character string. The character string can be dynamically switched.
List	In the list, several character strings are displayed for selection by the user.
Picture	The picture switches the display according to the status. Simple animation is available by switching the design.
Progress bar	The progress bar expresses the progress of a process with filling.
Radio button	The radio button allows the user to select one of multiple selection items and retain the selected state.
Text box	Character strings and values are displayed and input in the text box. The value is converted into a character string or binary data.
Scroll bar	The scroll bar moves the display area. The scroll bar includes two variations: vertical scroll bar for vertical movement and horizontal scroll bar for horizontal movement.
Input box	Input box is a control that has the same function as the text box object but differs in the following points. - Operation function was added. - The input values can be reflected to other designated controls.
Ten-key	Ten-key is a control equipped with key buttons and displays keys you entered. The following operations are also available. - Operation function was added. - The values input to a designated control can be reflected.
NC data text box	NC data text box is a control that enables reading and writing from and to the NC's internal data.
PLC button	The PLC button control enables to read and write data from/to the PLC bit device of NC. It also enables to change the ON/OFF state of the button according to the state of the bit device.
PLC extension button	PLC extension button is a control equivalent to a PLC button control, but is different from the PLC button in the following points. ・"Actuator" has been added to the button types. ・"Interlock", "Disable" and "Blink" have been added to PLC devices. ・The group designation function has been added.
PLC text box	The PLC text box control enables to read and write data from/to the PLC device of NC.
PLC message	PLC message is a control that displays a message according to the status of PLC device in NC, by obtaining it from the message definition text file.
Table	The table control uses cells with the number (n) of rows and columns to control and display the character string data. Each row and column can have its own title.
Counter	The counter display part can display the current position, workpiece coordinate position, etc.

Name	Outline
CycleTime	The cycle time display part is used to display the automatic start-up time and the cycle time. Automatic start time (STL)... Total accumulated time during the automatic operation, from when the automatic start-up button is pressed in the memory (tape) mode or MDI to when either of the feed hold stop, block stop or reset button is pressed. Cycle time (CYC).. The automatic operation time from when the automatic start-up button is pressed in the memory (tape) mode or MDI to when either of the feed hold stop, block stop or reset button is pressed.This is preset to “0" by turning the power OFF.
Feedrate	The F display part shows the vector direction speed currently being moved in during interporation feed, the speed of the axis with highest speed during each axis independent feed. This part also shows dwell (code: G04). Setting the property (speed display type) enables the display of tool tip speed.
GModal M	The G modal display part (M system) is used to show each modal state.
GModal L	The G modal display part (L system) is used to show each modal state.
GModal Simple	The G modal display part (simple) is used to show each modal state.
LoadMater	The Load meter display part can display the spindle load and Z axis load in the bar graphs by using user PLC. (When the load meter is not set by user PLC, these are not displayed on the screen.) Two load meters are displayed by using four lines (the area of the spindle standby and the load meter) when the spindle standby is not displayed. One load meter is displayed by using two lines (the area of the spindle standby and the load meter) when the spindle standby is displayed.
Menu	The menu display part includes the menu ON, OFF and Disable statuses, which can change the menu selection state. Two types of settings are available for the menu: one-row menu (icon + menu name) and two-row menu (two rows of menu name).
MSTB	The MSTB display part can display each command of spindle function (S), miscellaneous function (M), tool function (T) and 2nd miscellaneous function (B).
ONB	The ONB display part displays the program No., sequence No. and block No. currently being executed. When a subprogram is being executed, the subprogram's program No., sequence No. and block No. are displayed.
ProgramBuffer	The program buffer display part displays the contents of the machining program currently being executed. The block being executed in the program currently is highlighted.
SPCommand	This part can display the spindle modal (S) and the value of actual spindle rotation speed.
FileInOut	The input/output control part is used to input and output NC files between the NC memory and an external device.
AlarmMessage	The alarm display part is used to display the alarm No. and alarm message character string when an alarm occurs. This part displays NC alarms and PLC alarms, but does not display the stop code.
MonitorStatus	The operation status display part is used to display the NC operation status. If the system controls multiple part systems, this part can display the operation status of each part system.
Time	The time display part is used to display the current time.

2.2.3 Configuration of Data File Created With NC Designer

NC Designer handles a group of screen data displayed on the target board, as a project. The project consists mainly of a folder of which screen files are stored and a project data file.
Suppose you save a project under the name "Sample", a "Sample" folder and a project data file named "Sample.IPP" are created in the designated folder.



(Note): ###: A three-digit hexadecimal value indicating the page number

To open an existing project, select the file having extension IPP.
The screen data file having extension ipw is in the original format of NC Designer. When the user builds screen data in the application, source codes are generated according to screen data and they are embedded in the user program.

2.2.4 Operating Environment of NC Designer

The system environment necessary for the operation of NC Designer is shown below.

OS	Windows 2000/XP/Vista/7
CPU	Pentium III 600MHz or better
HD	100MB minimum Additional disk capacity is necessary for various data files.
Memory	128MB minimum
Screen	Resolution should be SVGA (800 x 600) or better

NOTE

- For Windows Vista/7, the file operation is limited by the improved security function (UAC/User Account Control Function).
  Specify the installation destination folder other than C:/Program Files.

2.2.5 Specification List

Function	Outline
Screen	1 for each project
Panel/window	Max. 256 sheets of panels and windows in total for each project
View frame	Max. 10 frames for each panel or window
Screen size	Horizontal: 1 to 2560 dots Vertical: 1 to 1920 dots
Number of controls that can be created on each page	Max. 512
Number of controls that can be created in each frame	Max. 512. However, the maximum limit of the page (512) may not be exceeded inside the page of the frame.
Number of locales	Max. 32
Background image file	BMP or JPG file
Filling pattern	Up to 38 types can be registered.
Line pattern	Up to 8 types can be registered.

2.2.6 Function List

Function	Outline
Property window	Specify the properties of the control, figure, project, screen, window and frame.
Resource view	The resource data is displayed in a tree. Registration of the resource is available.
Source code generation	Various source codes are generated according to the created screen data.
Automatic generation of development document	Project and window data is output in a file (rich text format).
Simulation function	The appearance and movement of the created screen data is simulated on the PC.
Editing function	Various editing functions such copying, cutting, pasting, search and replacement can be performed.
Layout function	Positioning, resizing, rotation and other layout adjustment functions can be performed.
Locale switching	Up to 32 locales can be set for each control, and they are switched on NC Designer.
Error check	Various setting errors are checked.
Control list	A list of created controls and property settings are displayed.

2.3 Menu List

A list of pull-down menus of NC Designer and the usage of each item are described below.

2.3.1 File

Item	Function
New Project	Create a new project.
Open Project	Open an existing project.
Save Project	Overwrite the project being edited.
Save Project As	Save the project being edited, under a new name.
New Panel	Add a new panel to the project being edited.
New Window	Add a new window to the project being edited.
Open Panel/Window	Open the panel/window of the project being edited.
Close Panel/Window	Close the panel/window being edited.
Import Panel/Window	Copy the panel/window of another project to the project being edited.
Save Panel/Window	Save the panel/window being edited.
Save All	Overwrite all the project data being edited.
Source code generation	Convert the created data into source code of various formats.
Project convert	Convert the created data in an interpreter method file.
Document generation	Output project and window data into a file (rich text format).
Recent Projects	Read and display up to four recently edited projects.
Exit	Exit from NC Designer.

2.3.2 Editing

Item	Function
Undo	Abandon changes and restore the original state.
Redo	Execute the operation canceled with "undo."
Cut	Delete the selected object and copy it in the clipboard.
Copy	Copy the selected object and copy it in the clipboard.
Paste	Paste the object from the clipboard.
Delete	Delete the selected object (without copying it in the clipboard).
Find	Search for a control or caption.
Edit of a caption	Edit the caption of each part directly on the editing screen.
Select All	Select all objects or all controls or figures of the same type on the editing page.
Repeat	Copy the selected object by the designated number vertically or horizontally.

2.3.3 View

Item	Function
Toolbars	Select the tool bar displayed with NC Designer.
Resource	Display or hide the resource view.
Property	Display or hide the properties window.
Message	Display or hide the message window.
Image	Display or hide the image view.
Statusbar	Display or hide the status bar.
Switch Locale	Switch the locale being displayed.
Previous Screen	Display the previous page.
Next Screen	Display the next page.
Previous Frame Page	Select the previous view frame.
Next Frame Page	Select the next view frame.
Zoom	Specify the zoom ratio of the page.
Refresh	Redraw the page.

2.3.4 Control

Item	Function
Basic object	Select to draw the basic control.
Button object	Select to draw a button.
Checkbox object	Select to draw a check box.
Edit control object	Select to draw an edit control.
HTML browser object	Select to draw an HTML browser.
Label object	Select to draw a label.
List object	Select to draw a list.
Picture object	Select to draw a picture.
Progressbar object	Select to draw a progress bar.
Radiobutton object	Select to draw a radio button.
Textbox object	Select to draw a text box.
Vertical scroll bar object	Select to draw a vertical scroll bar.
Horizontal scroll bar object	Select to draw a horizontal scroll bar.
Input box	Select to draw an input box.
Ten-key	Select to draw a ten-key.
NC data text box	Select to draw an NC data textbox.
PLC button	Select to draw a PLC button.
PLC extension button	Select to draw a PLC extension button.
PLC textbox object	Select to draw a PLC textbox.
PLC message	Select to draw a PLC message.
Table object	Select to draw a table.
Counter	Select to draw a counter display part.
CycleTime	Select to draw a cycle time display part.
Feedrate	Select to draw a feedrate (F) display part.
GModal M	Select to draw an M system modal display part.
GModal L	Select to draw an L system modal display part.
GModal Simple	Select to draw a simple modal display part.
LoadMeter	Select to draw a load meter display part.
Menu	Select to draw a menu part.
MSTB	Select to draw a MSTB part.
ONB	Select to draw an ONB display part.
ProgramBuffer	Select to draw a program buffer display part.
SPCommand	Select to draw a spindle (S) display part.
FileInOut	Select to draw an input/output control part.
AlarmMessage	Select to draw an alarm message display part.
Monitor	Select to draw an operation status display part.
Time	Select to draw a time display part.
Frame	Select to draw a view frame.

2.3.5 Figure

Item	Outline
Rectangle	Select to draw a rectangle.
Circle&Oval	Select to draw a circle or ellipse.
Straight Line	Select to draw a line.
Poryline	Select to draw a continuous line.
Polygon	Select to draw a polygon.
Sector	Select to draw a sector.
Arc	Select to draw an arc.

2.3.6 Setting

Item	Function
Project Properties	Enter the project property settings.
Panel/Window Properties	Enter the panel/window property settings.
Focus setup	Enter the focus destination setting.
Color setup	Enter the settings of the project color setting method.
Panel macro edit	Edit the macro used in the screen.
Project macro edit	Edit the macro used in the project.

2.3.7 Layout

Item	Function
Align/Distribution	Change the alignment or arrangement of selected multiple objects.
Make Same Size	Resize the horizontal or vertical width of selected multiple objects to the smallest or largest object size.
Order	Display the selected object at the far front or far back position.
Nudge	Move the selected object left, right, up or down by a dot or grid.
Rotate/Flip	Rotate the object or create a mirror image of it.
Modify	Deform the polygon or arc.
Group	Group multiple objects. Or cancel a group.
Grid	Enter the grid setting.

2.3.8 Tool

Item	Function
Screen Maintenance	Copy or delete the screen.
BSP selection	Select the BSP to be used.
Error check	Perform an error check of the created data.
Functional Object List	Display a list of controls and jump to the selected control.
Export of a character sequence resource	Export the character string resource in a CSV file.
Import of a character sequence resource	Import the character string in a CSV file into the character string resource.
Test	Perform an action test of the screen.
Options	Enter reference setting of common project data.

2.3.9 Window

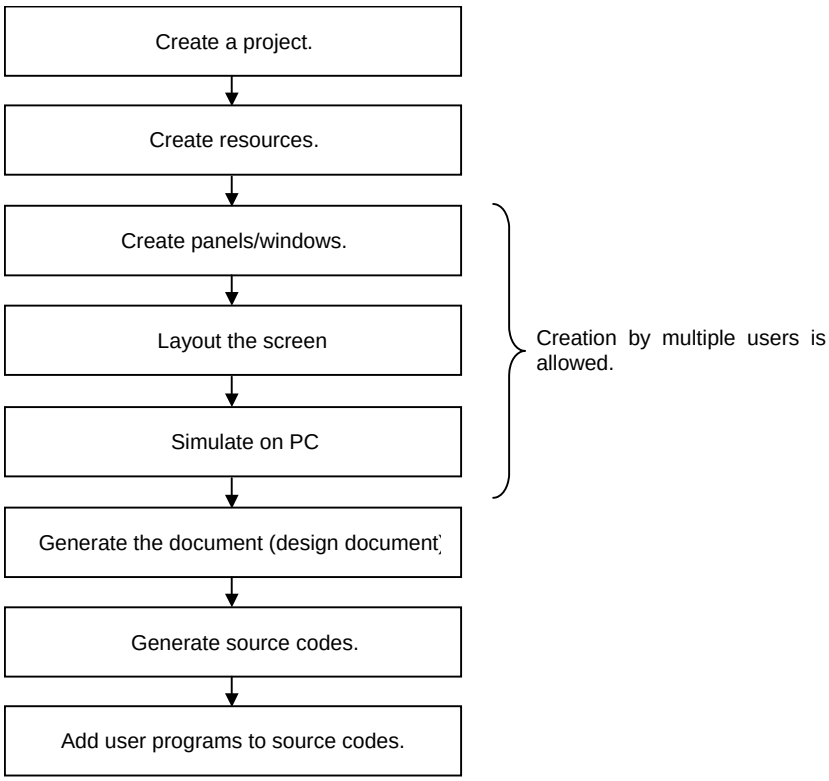
Item	Function
Cascade	Overlap screen editing windows.
Tile	Arrange screen editing windows like tiles.
Arrange Icons	Arrange minimized window icons.

2.3.10 Help

Item	Function
About NC Designer	Display the version of the product.

2.4 Flow of Development Using NC Designer

The flow of GUI development using NC Designer is shown below.



Resources such as the character strings and image data are controlled by a single user because they are common project data. Create the character string of each control and the window title character string in advance as resources. After creating resources, each screen layout can be arranged by multiple users.
After source codes are generated, add the user-specific program to the source code and perform application development.

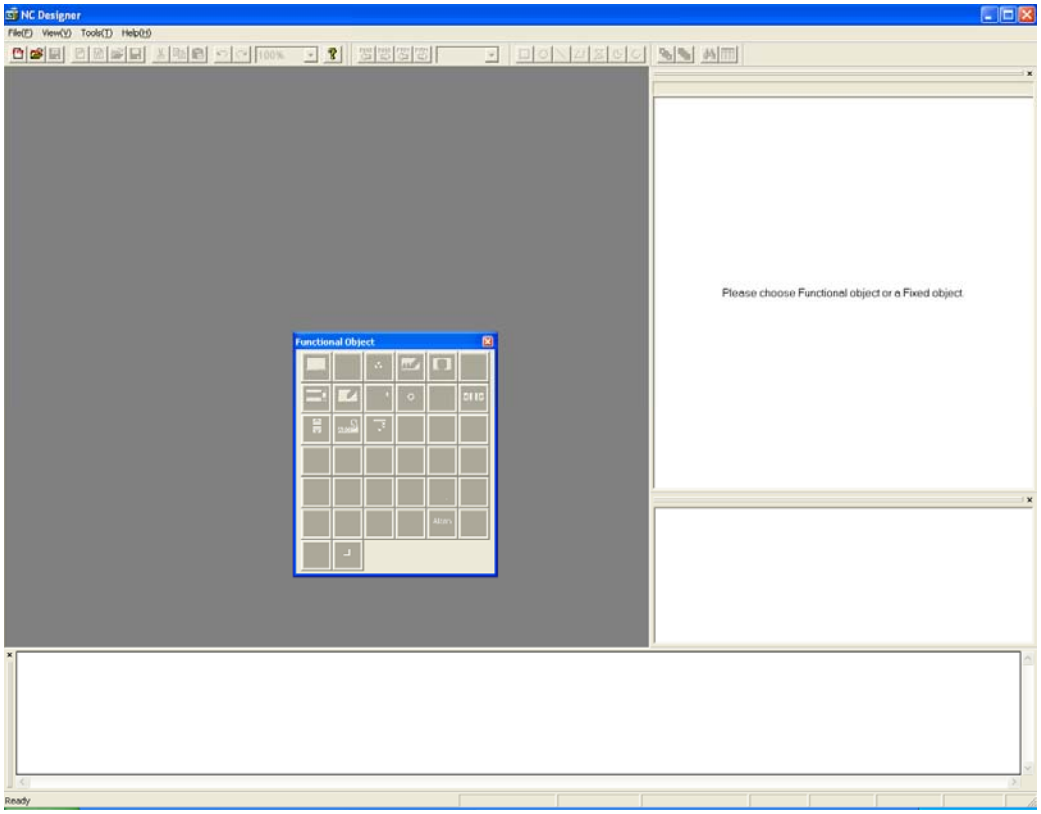
3. Startup and Termination of NC Designer

The startup and termination methods of NC Designer are described in this section.
Refer to "3.4 Installing NC Designer" for how to install NC Designer.

3.1 Starting NC Designer

To launch NC Designer, select the start button of Windows, followed by "Programs," → "MELSOFT application," → "NC Designer" and "NC Designer." (The displayed names may vary according to the "program folder" designated during installation.

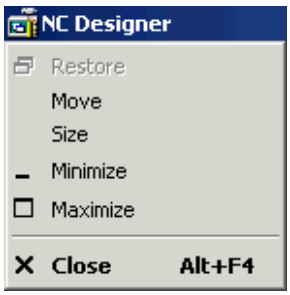
After NC Designer is launched, the main window shown below is displayed.



3.2 Exiting From NC Designer

Perform one of the following procedures to exit from NC Designer.

- Select [Exit] from the [File] menu.
- Click on the button at the upper right of the main window.
- Double click on the NC Designer icon at the upper left of the main window.
- Click on the NC Designer icon at the upper left of the main window and select [Close] from the displayed control menu box



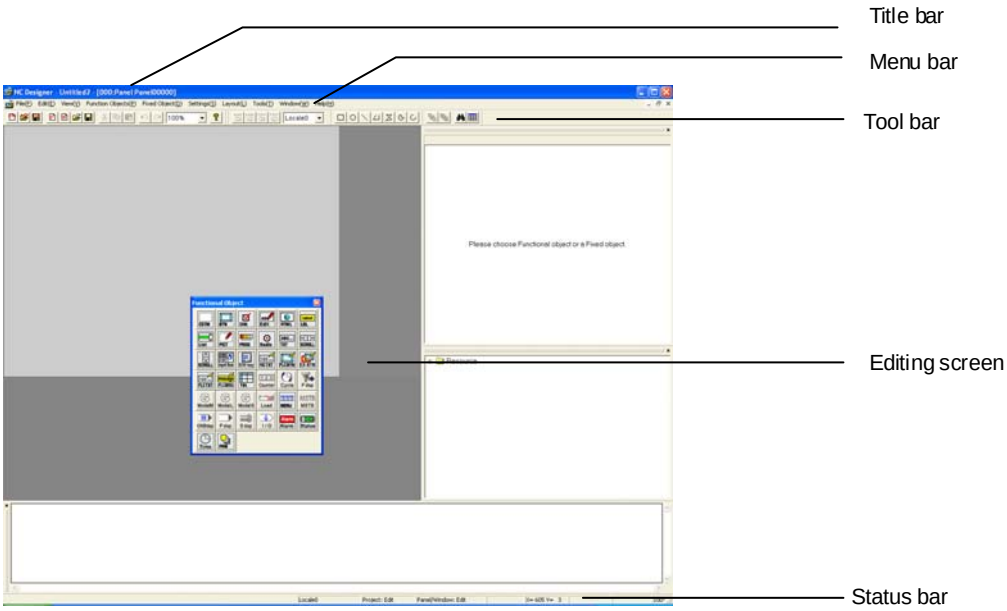
- While holding down the [Alt] key, press the [F4] key.

If the open project data has not been saved, a confirmation message is displayed.

3.3 User Interface

3.3.1 Function of Each Part of Basic Screen

The configuration of the operation screen of NC Designer and the name and function of each part are described.



Title Bar

The application name, project name, and panel/window name are displayed.

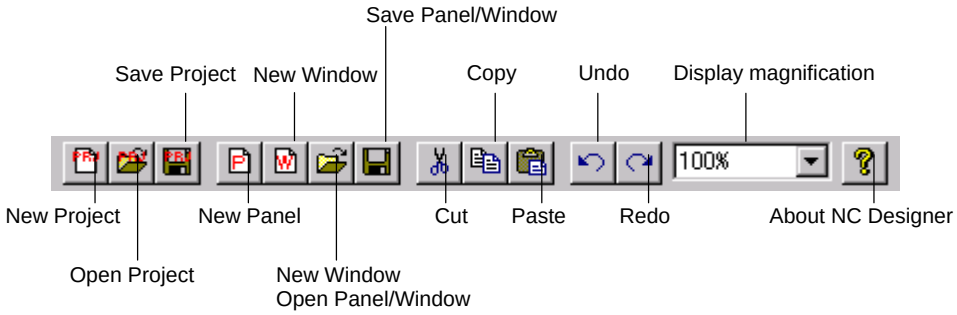
Menu Bar

Related functions are grouped.
The menu bar shows a list of group names, and each function in the group is displayed in the pull-down menu.

Tool Bar

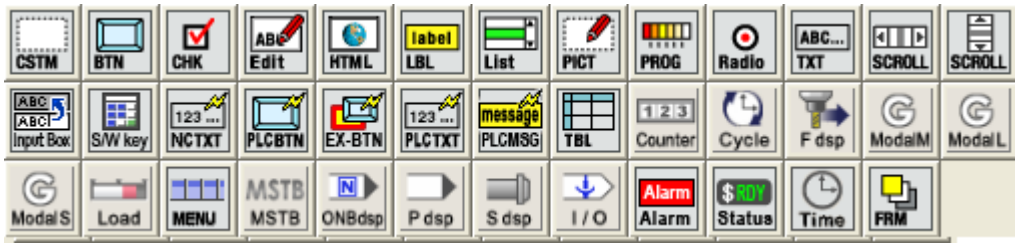
(1) Standard Tool Bar

Frequently used functions are displayed in icons.



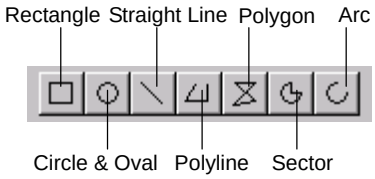
(2) Control Tool Bar

The drawing function of each control is displayed in an icon. (The image of each control is displayed in an icon.) Select the icon button corresponding to the desired control to draw the control.



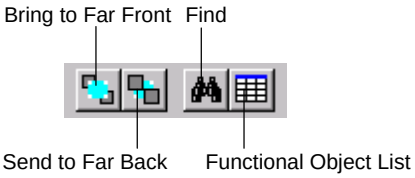
(3) Figure Tool Bar

The figure drawing function is displayed in icons. Select the icon button corresponding to the desired figure to draw the figure.



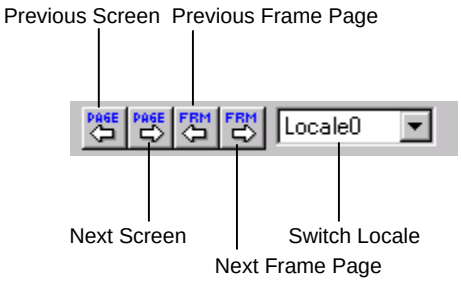
(4) Control Operation Tool Bar

Functions frequently used during operation of controls are displayed in icons.



(5) View Tool Bar

Functions frequently used in the [View] menu bar are displayed in icons.



Resource View

Character strings, images and other resources are displayed and entered.

Property Window

Properties of the project, window, view frame, control, figure and each resource are entered.

Image View

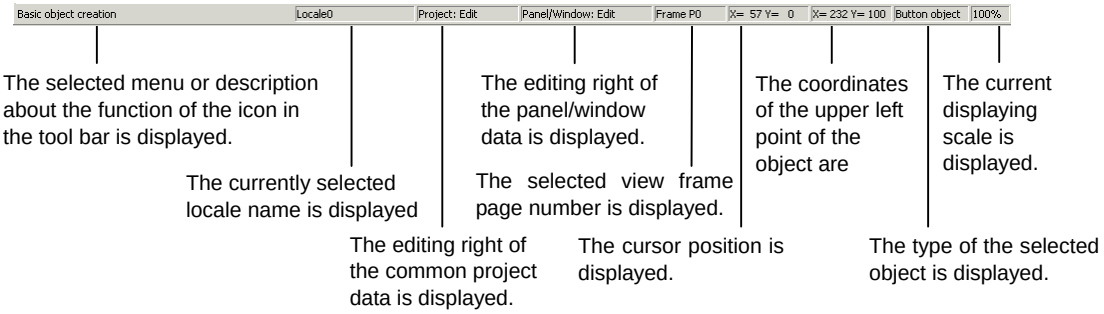
A preview of the image data set in the image resource selected in the resource view is displayed.

Drawing Screen

The screen displayed at the target is created.

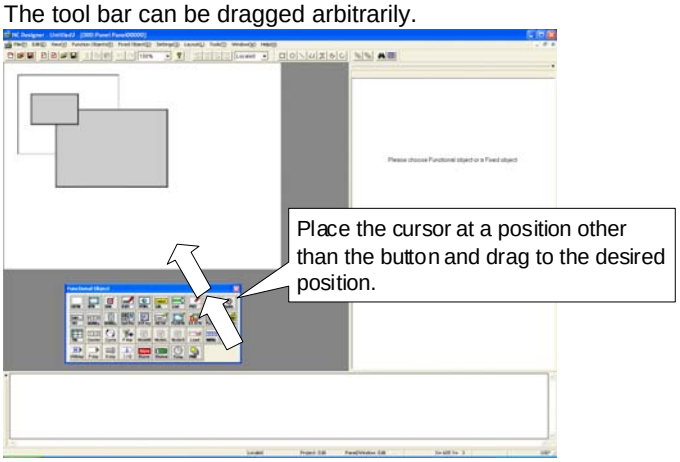
Status Bar

The position of the cursor and descriptions about the selected function or control are displayed.



NOTE

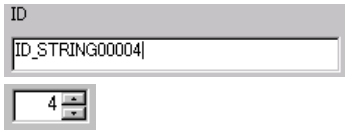
- The tool bar and status bar can be displayed or hidden upon selection.
  Tool bar : From the [View] menu, select [Tool Bar] - [Standard]/[Functional Object]/[Fixed Object]/[Operation]/[Display].
  Status bar : Select [Status Bar] from the [View] menu.
  The check mark placed before an item indicates that the item is displayed.
- The tool bar can be dragged arbitrarily.



3.3.2 Operation Method in Dialog Box and Property Window

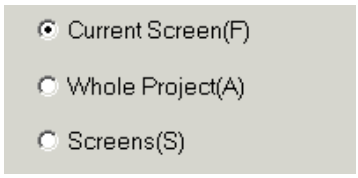
In the dialog box and property window, detail settings for the execution of each function of NC Designer can be entered.

Edit Box



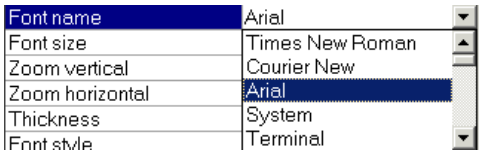
Enter a character string. With an edit box provided with spin buttons, click on the up "▲" or down "▼" button to increase or decrease the value.

Radio Button



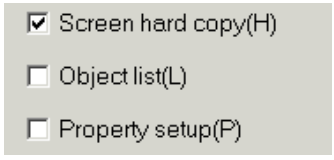
Selection items with a circle at the left side.
Only one option among several options can be selected. The selected option is indicated with a dot.

Combo Box



A list of selection items is displayed in a drop-down menu. You can select the desired one.

Check Box



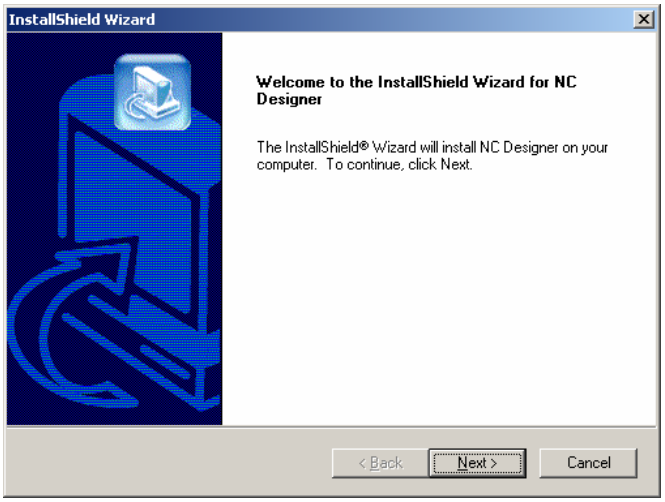
Selection items with a square at the left side.
Select the validity of the option. The selected option is marked with a check mark ✔.

3.4 Installing NC Designer

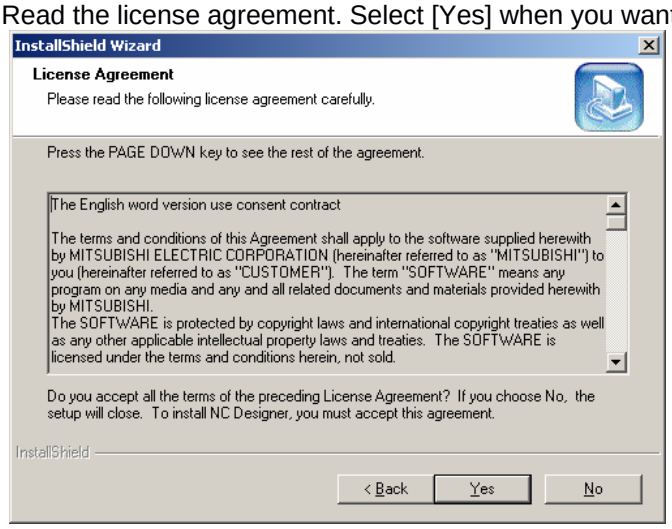
This section explains how to install NC Designer.

If the old version of NC Designer has already been installed, it will be uninstalled at the first execution of the installer.
Executing the installer again will install the new version of NC Designer.

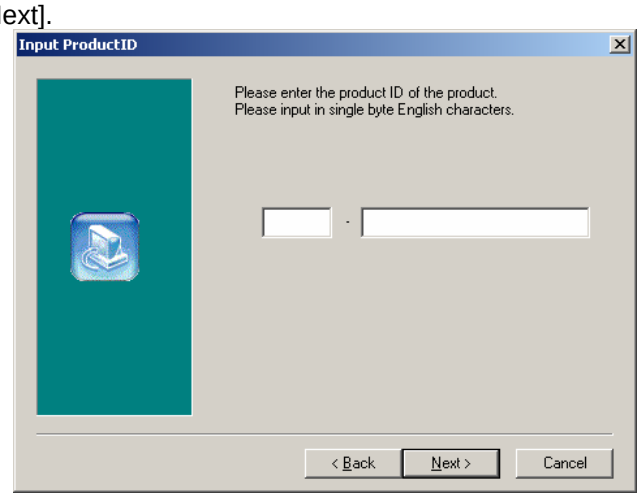
1. Execute "NCDesigner.exe" from CD-ROM. Then the installer will start up. Select [Next].



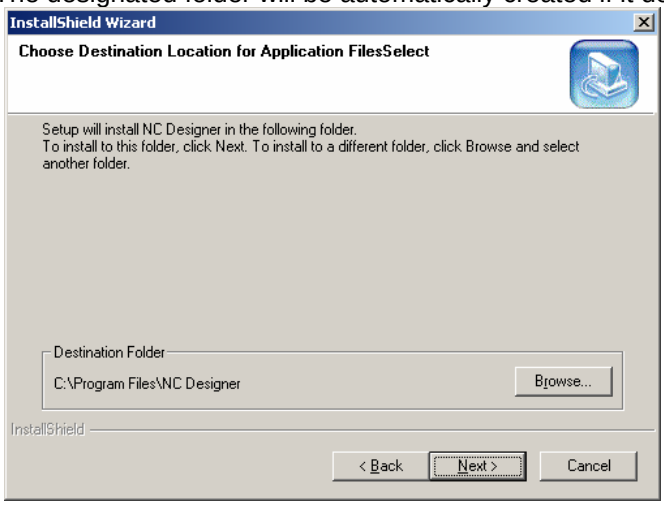
2. Read the license agreement. Select [Yes] when you want to agree.



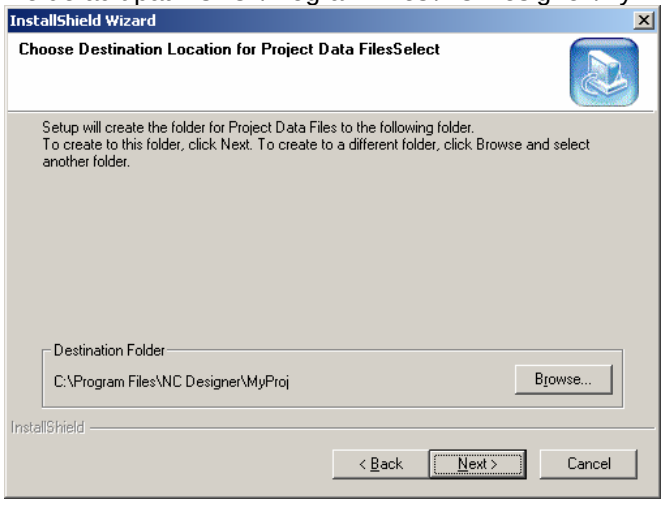
3. "Input ProductID" screen will appear. Enter the ID No. attached to your product and select [Next].



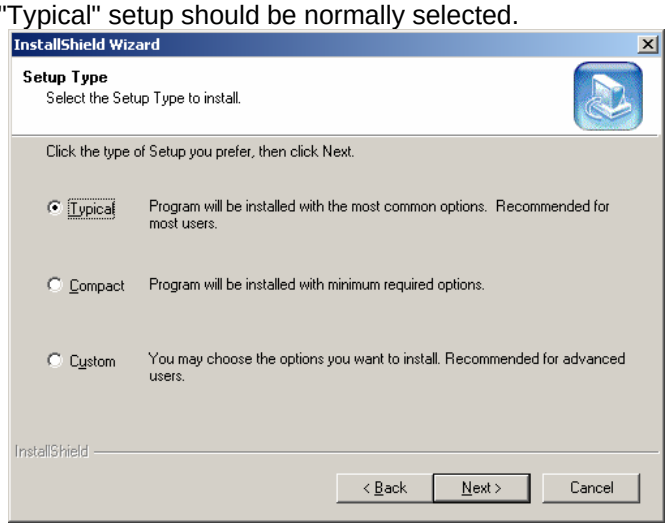
4. Select the destination for the application to be installed. Click [Next].
The default installation path is "C:\Program Files\NC Designer".
The designated folder will be automatically created if it does not exist.



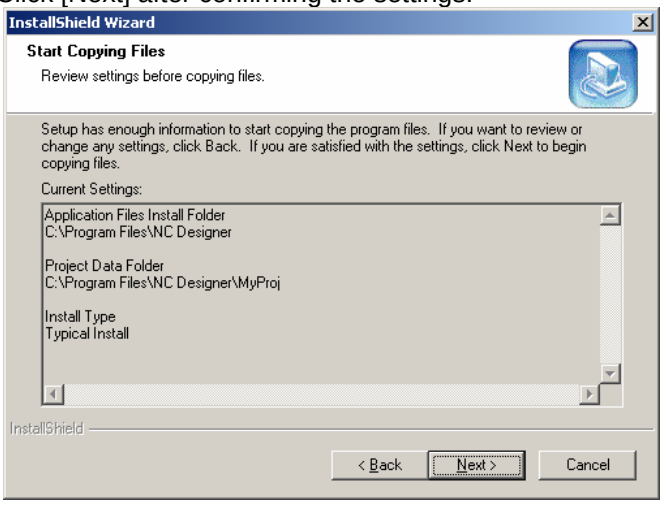
5. Designate the folder for project data files. Click [Next].
The default path is "C:\Program Files\NC Designer\MyProj".



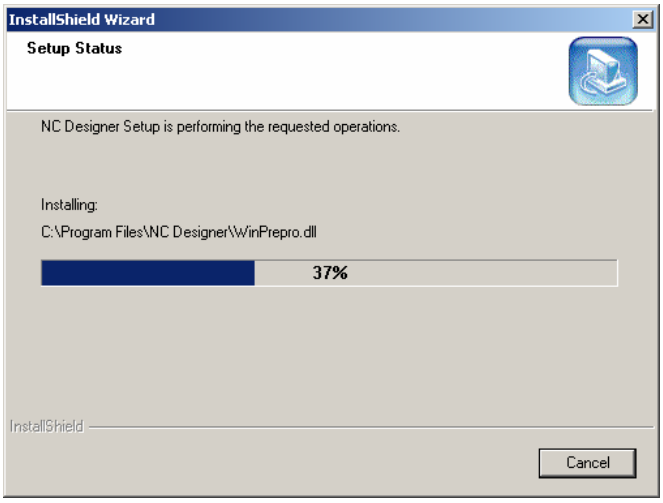
6. Select the setup type and click [Next].
"Typical" setup should be normally selected.



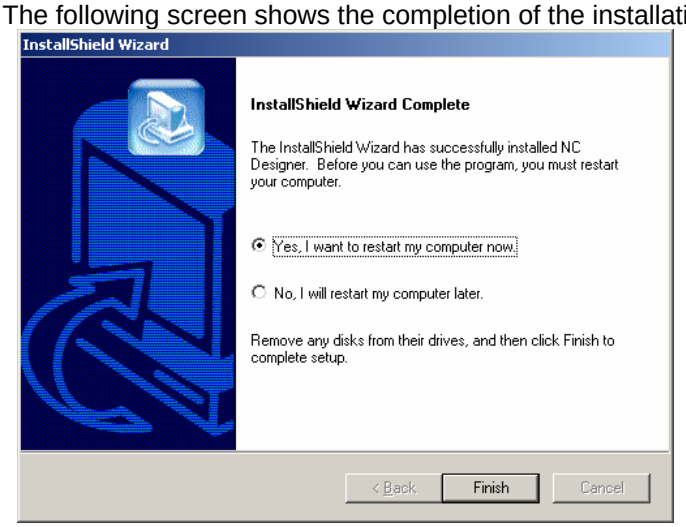
7. Click [Next] after confirming the settings.



8. The installation will start.



9. The following screen shows the completion of the installation. Select [Finish] to exit the installer.

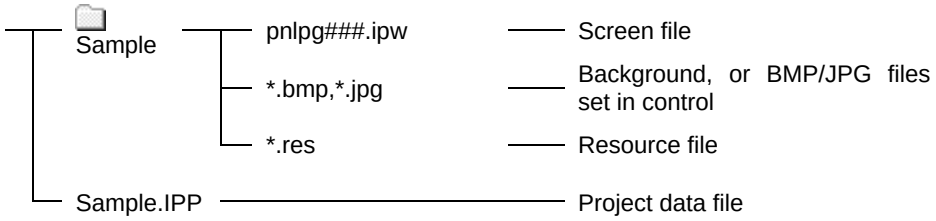


4. Creating a Project

Basic operations about the project where various pieces of created data are stored are described in this section.

4.1 What Is "Project"?

With NC Designer, a group of GUI screens used for a certain application is called "project."
The project mainly consists of a folder of which screen files are stored and a project data file.
When the project is saved under the name "Sample," for example, a "Sample" folder and a "Sample.IPP" project data file are created in the folder designated as a destination of saving.



(Note): ###: A three-digit hexadecimal value indicating the page number

To open an existing project, select the file having extension IPP.

IMPORTANT

- To move or copy project data on a floppy disk or to another PC, select both the folder and project data file. Because the project consists of these two pieces of data, the project does not open with only one of them.
- Settings related to the screen size and display scale are stored in the CONFIG.INI file. The CONFIG.INI file is created in the folder designated in the [BSP Selection] dialog box.

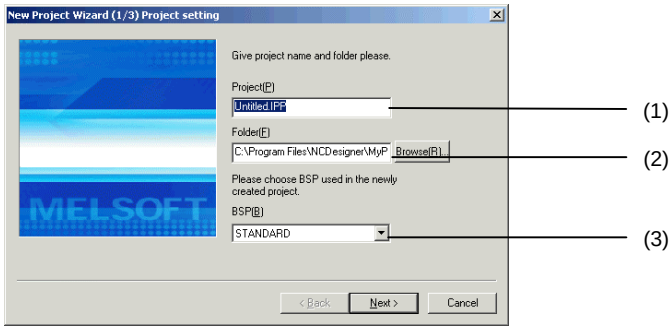
4.2 Creating a New Project

The procedure from creation of a new project to editing of the screen is described.

1. Select [New Project] from the [File] menu or select the [New Project] button in the tool bar.

2. The New Project wizard is displayed. Follow the instructions in the wizard to enter settings about the project.

Project setting BSP selection



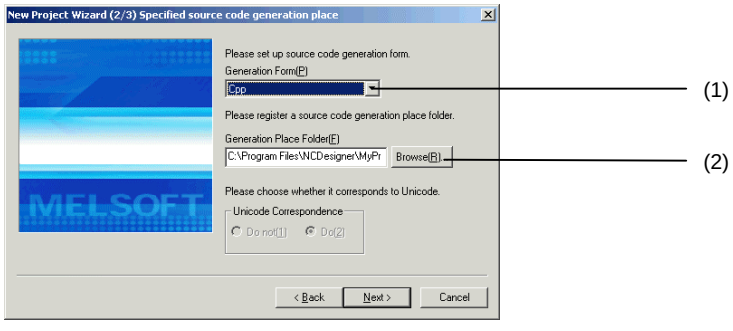
No.	Item	Description
(1)	Project	Enter the project name.
(2)	Folder	Designate the folder where the project is stored with a full path.
(3)	BSP	A list of BSP having been installed is displayed. Select the desired BSP.

NOTE

◆ In the compilation method, the following project names cannot be used.
- The same name as the panel/window name
- "SampleScreen"
- The combination of the panel/window name and the view frame name is same as the project name.

Example
- Project name: TestMonitor
- Panel/window name: Test
- View frame name: Monitor

Source code generation destination designation

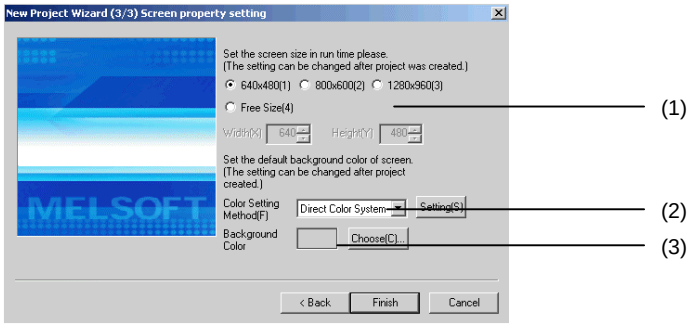


No.	Item	Description
(1)	Generation Form	Select the source code generation form from the list.
(2)	Generation Place Folder	Designate the folder which generates the source file with a full path.

NOTE

- If there is another source file for the other project in the folder designated as a destination folder, the file will be overwritten during source code generation.

Screen property setting



No.	Item	Description
(1)	Page size	Select the default screen size of the page.
(2)	Color Setting Method	Select the color setup method of the project, using "Direct Color System". Click on the Set button to display the [Color] dialog box.
(3)	Background Color	Designate the default background color of the page. Click on the Select button and select the desired color in the displayed [Color] dialog box.

3. Click on the [Finish] button to create the project.

NOTE

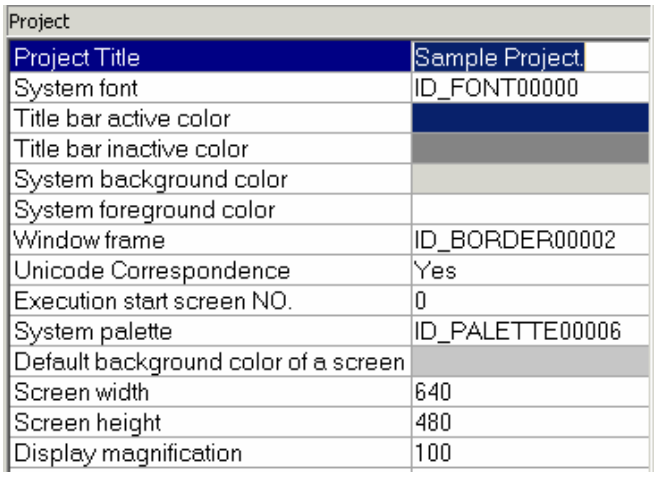
- After a project is created, the panel of page 0 is automatically created.
- If a new project is created while another project is being edited, a message urging to save the currently edited project is displayed.
- Settings related to the screen size and display scale are stored in the CONFIG.INI file. The CONFIG.INI file is created in the folder designated in the [BSP selection] dialog box.

4.3 Entering Project Properties

Enter the properties of the currently edited project.

1. Select [Project Properties] from the [Setting] menu.

2. Project properties are displayed in the property window.



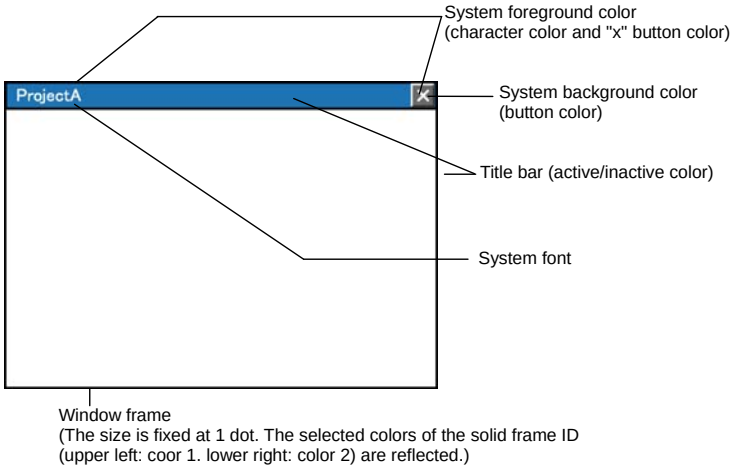
3. Enter each item.

Enter the following items.

Item	Description
Project Title	Enter a comment about the project.
System font	Select the resource ID of the character font displayed in the title bar of the window.
Title bar active color	Specify the color of the title bar of the active window.
Title bar inactive color	Specify the color of the title bar of the inactive window.
System background color	Specify the background color of the button displayed in the title bar of the window.
System foreground color	Specify the foreground color of the button displayed in the title bar of the window.
Window frame	Select the shape of the window frame from the solid frame resource ID.
Unicode Correspondence	Select "Yes" for Unicode character code of each locale.
Execution start screen No.	Specify the page number of the panel displayed first.
System palette	Select the palette used in the system, from the registered palette resource.
Default background color of a screen	Specify the default background color of the panel and window.
Screen width	Designate the screen width in dots (1 to 2560).
Screen height	Designate the screen height in dots (1 to 1920).
Display magnification	Designate the display scale of the application in percent (25 to 800).

The setting is confirmed when it is entered.

The system font and color settings are reflected as shown below.



NOTE

- The user who has opened the project to "Edit" common project data can change the project properties.
- If "Yes" is selected for "Unicode Correspondence," the source code of the character string resource is generated with Unicode during source code generation.
  If "No" is selected, source code of the character string resource is generated with the character code specified for each locale.

4.4 Changing the Color Setting Method

Change the color setting method to reflect the change on the color setting of the entire project.

1. Select [Color setup] from the [Setting] menu.
   A Save File confirmation message is displayed. To change the color setting method, click on the [OK] button to save the file.

2. A [Color setup] dialog box is displayed.
   For the [Color setup] dialog box, refer to Section 4.2 "Creating a New Project."

3. Click on the [OK] button. A confirmation message is displayed. Select the [OK] button to reflect the changes in all settings related to the color.

NOTE

- The user who has opened the project to "Edit" common project data can change the color setting method.
- The user who has opened the project to "Refer to" common project data must close the editing page when the color designation method is changed.
- If the "color setting method" is changed from "Index Color System" to "Direct Color System," the RGB values of the system palette are reflected on the existing color setting.
  If the "color setting method" is changed from the "Direct Color System" to "Index Color System," the colors with the most resembling RGB values are searched for in the system palette and their index numbers are specified for the existing color setting.
  If the "color setting method" is changed from "Direct Color System" to "Index Color System" to "Direct Color System," the original colors may not be restored because approximate colors are used during conversion into index color.
- If "The number of Colors" is changed and the number of colors is reduced when compared with that before the change, approximate colors are searched for from the system palette and their index numbers are assigned in color setting.
  If the number of colors increases when compared with that before the change, the index number is not converted.
- After the color setting is changed, all editing pages are closed.

4.5 Saving the Project

The method for saving the project file is described.

1. Select [Save Project] from the [File] menu or select the [Save Project] button in the tool bar.

NOTE

- If the edited project has not been saved, an overwrite confirmation dialog box is displayed when the project is closed.

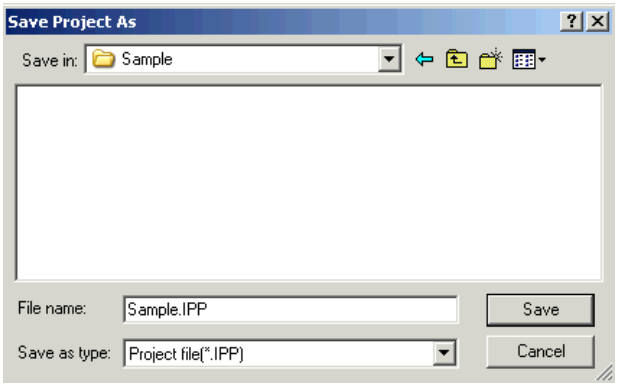
4.6 Saving the Project as ...

Save the currently edited project under another file name or at another location.

1. Select [Save Project as ...] from the [File] menu.

2. The [Save Project as] dialog box is displayed.

   Designate the destination directory and file name and click on the [Save] button.



NOTE

- If the project is saved under a new name, all the currently opened files are copied to the directory of the new project.
- For Windows Vista/7, the file operation is limited by the improved security function (UAC Function/User Account Control Function).
  Specify the installation destination folder other than C:/Program Files.
- In the compilation method, the following project names cannot be used.
  - The same name as the panel/window name.
  - "SampleScreen"
  - The combination of the panel/window name and the view frame name is same as the project name.

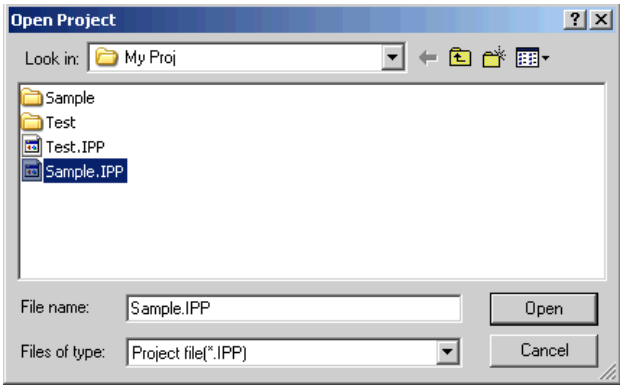
  Example
  - Project name: TestMonitor
  - Panel/window name: Test
  - View frame name: Monitor

4.7 Opening a Project

Open an existing project.

1. Select [Open Project] from the [File] menu or select the [Open Project] button in the tool bar.

2. The Open Project dialog box is displayed.
   Select a project file (extension: IPP) and click on the [Open] button.

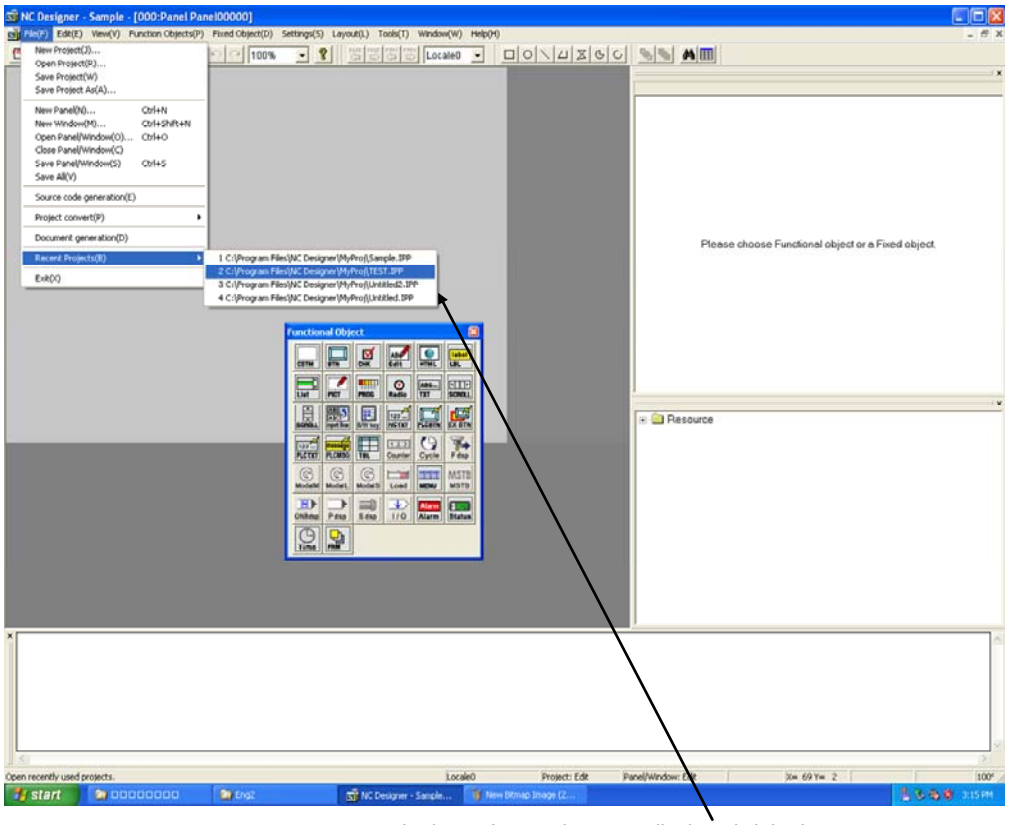


4.8 Opening a Recently Used Project

Up to four most recent projects can be opened directly.

1. Select [Recent Projects] from the [File] menu.

2. The recently used projects are displayed with directory.
   Select the desired project.



The latest four projects are displayed. (The latest one appears at the top.)

5. Creating and Saving the Page

In this section, the page (screen) that can be displayed on the target board and the basic operations available in each page are described.

5.1 What Is Page?

The "page" indicates a screen displayed with NC Designer.
The page includes the following two types.

Item	Description
Panel	Screen displayed in full size on the screen.
Window	Screen displayed in a window state on the screen.

5.2 Creating a New Panel

Create a new panel in the currently edited project.

1. Select [New Panel] from the [File] menu or press the [New Panel] button in the tool bar.

2. A new panel is created.

NOTE

- After a new panel is created, the smallest page number among unused page numbers is automatically assigned to the new panel.

5.3 Creating a New Window

Create a new window in the currently edited project.

1. Select [New Window] from the [File] menu or select the [New Window] button in the tool bar.

2. A new window is created.

NOTE

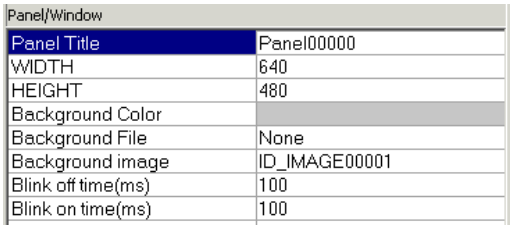
- After a new window is created, the smallest page number among unused page numbers is automatically assigned to the new window.

5.4 Entering Panel Properties

Enter the properties of the currently edited panel.

1. Select [Panel/Window Properties] from the [Setting] menu or select the [Panel/Window Properties] from the popup menu displayed upon a click of the right mouse button.

2. Panel properties are displayed in the property window.



3. Enter each item.

The setting items include the followings.

Item	Description
Panel Title	Enter a one-byte character string to specify the panel title (up to 31 characters). The first character must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_).
WIDTH	Specify the panel width in dots (1 to 2560).
HEIGHT	Specify the panel height in dots (1 to 1920).
Background Color	Specify the background color.
Background File	Select "Yes" to use the background, or select "No" to refrain from using it.
Background Image	Select the ID of the image resource used for the background. Use the resource view to register the image resource.
Blink off time	Enter the OFF interval (regular display) of the control blink in ms (100 to 60000).
Blink on time	Enter the ON interval of the control blink in ms (100 to 60000).

The setting is confirmed when entry is finished.

NOTE

- For the blink, refer to "Blink" in Section 7.1 "Common Control Functions."
- In the compilation method, the following project names cannot be used.
  - The same name as the project name
  - The same name as the window name
  - "SampleScreen"
  - The combination of the view frame name and the panel name is "SampleScreen".
  - The combination of the view frame name and the panel name is same as the project name.

  Example
  - Project name: TestMonitor
  - Panel name: Test
  - View frame name: Monitor

5.5 Entering Window Properties

Enter the properties of the currently edited window.

1. Select [Panel/Window Properties] from the [Setting] menu or select [Panel/Window Properties] from the popup menu displayed upon a click of the right mouse button.

2. Window properties are displayed in the property window.

Panel/Window	
Window Name	Window00001
X	0
Y	0
WIDTH	640
HEIGHT	480
Background Color	
Background File	None
Background image	ID_IMAGE00001
Title	
Existence of a title bar	Yes
Existence of a close button	Yes
Existence of a window frame	Yes
Blink off time(ms)	100
Blink on time(ms)	100

3. Enter each item.

The following items can be entered.

Item	Description
Window Name	Specify the window name (up to 31 characters). The first character must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_).
X	Specify the X coordinate of the upper left corner of the window in dots (0 to 2559).
Y	Specify the Y coordinate of the upper left corner of the window in dots (0 to 1919).
WIDTH	Specify the panel width in dots (1 to 2560).
HEIGHT	Specify the panel height in dots (1 to 1920).
Background Color	Specify the background color.
Background File	Select "Yes" to use the background image, or select "None" to retrain from using it.
Background Image	Select the ID of the image resource used as a background image. Use the resource view to register the image resource.
Title	Select the character string displayed in the title bar of the window from the character string resource, or enter a new one.
Existence of a title bar	Select "Yes" to display the window with a title bar, or select "None" to display it without a title bar.
Existence of a close button	Select "Yes" to display a close button in the title bar of the window, or select "None" to refrain from displaying the close button.
Existence of a window frame	Select "Yes" to display the window with the window frame, or select "None" to display the window without the window frame.
Blink off time	Specify the OFF interval (regular display) of the blink of the control in ms (100 to 60000).
Blink on time	Specify the ON interval of the blink of the control in ms (100 to 60000).

The setting is confirmed when entry is finished.

NOTE

- For the blink, refer to "Blink" in Section 7.1 "Common Control Functions."
- In the compilation method, the following project names cannot be used.
  - The same name as the project name
  - The same name as the window name
  - "SampleScreen"
  - The combination of the view frame name and the panel name is "SampleScreen".
  - The combination of the view frame name and the panel name is same as the project name.

  Example
  - Project name: TestMonitor
  - Panel name: Test
  - View frame name: Monitor

5.6 Saving the Panel/Window

The method for saving the currently edited panel/window is described.

1. Select [Save Panel/Window] from the [File] menu or select [Save Panel/Window] button in the tool bar.

Tool bar

NOTE

- The shortcut key for [Save Panel/Window] is [Ctrl] + [S].
- If the panel/window being edited has not been saved, a confirmation dialog box is displayed when the panel/window is closed. To save, click on the [Yes] button.

5.7 Save All

The entire project is saved.
After this procedure, not only the opened page but also the entire project is saved.

1. Select [Save All] from the [File] menu.

2. After the project is saved, a message dialog box is displayed.
   Click on the [OK] button.

NOTE

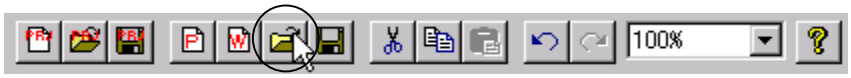
- Saving the entire project causes a change in the common project data. Only the user holding the right of editing the common project data can execute the function.

5.8 Opening a Panel/Window

Open the panel/window of the currently open project.

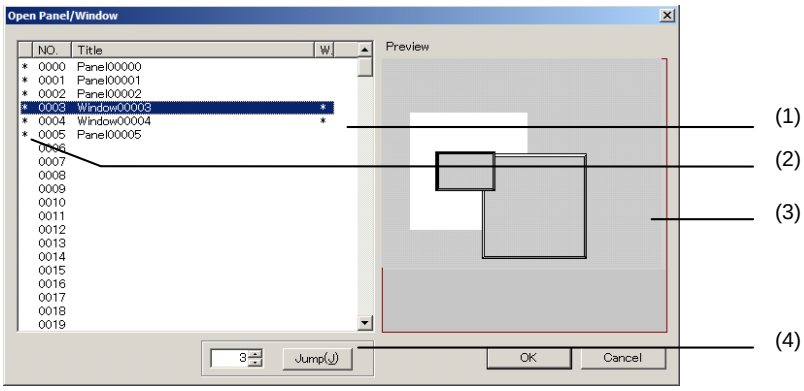
1. Select [Open Panel/Window] from the [File] menu or select the [Open Panel/Window] button in the tool bar.

Tool bar



2. An "Open Panel/Window] dialog box is displayed.

Select the desired screen and click on the [OK] button.



No.	Item	Description
(1)	Window	The window is marked with an asterisk (*).
(2)	Operation state	The page being used is marked with an asterisk (*).
(3)	Preview	A preview of the selected page is displayed.
(4)	Jump	Designate the page number and click on the [Jump] button to jump to the designated page.

3. The selected screen is displayed.

NOTE

- The shortcut key for [Open Panel/Window] is [Ctrl] + [0].
- If a page with an unused page number is opened, a new panel is created.
- The number of pages that can be edited simultaneously is 16.

5.9 Closing the Panel/Window

Close the open panel/window without closing the project.

1. Select [Close Panel/Window] from the [File] menu or click on the [x] mark at the upper right of the page view.

[In case of window]

Click

[In case of maximized window]

Click

5.10 Switching the Editing Window

The method for switching the editing window is described.

5.10.1 Previous Screen

The editing window of the previous page number is displayed.

1. Select [Previous Screen] from the [View] menu or select the [Previous Screen] button in the tool bar.

Tool bar

NOTE

- The editing window is the screen area created in the editing page (panel or window) units.
- If there is no panel or editing window one the page earlier than the currently opened panel or window, this function may not be used.
- An error is caused if a new screen is opened beyond the limit in the number of editing pages.

5.10.2 Next Screen

The screen of the next screen page number is displayed.

1. Select [Next Screen] in the [View] menu or select [Next Screen] button in the tool bar.

Tool bar

NOTE

- If there is no panel or editing window for the page numbers later than the currently opened panel or window, this function may not be used.
- An error is caused if a new screen is opened beyond the limit in the number of editing pages.

5.10.3 Cascade

The active window is brought to the front.

1. Select [Cascade] from the [Window] menu.

5.10.4 Tile

Open page windows are displayed in tiles.

1. Select [Tile] from the [Window] menu.

5.10.5 Arrange Icons

Minimized page windows are arranged neatly.
Minimized page windows are arranged from left to right below the application window.

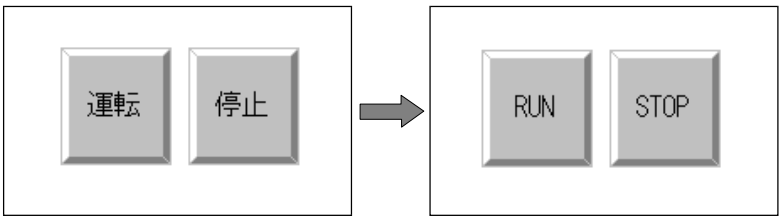
1. Select [Arrange Icons] from the [Window] menu.

NOTE

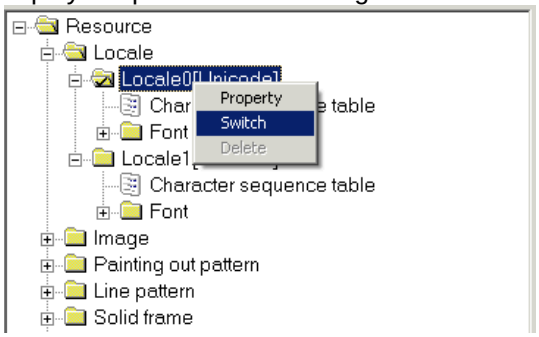
- This function may not be used if there is no minimized page window.

5.10.6 Switching Locale

When two or more locales are registered, the locale is switched.



1. Select [Switch Locale] from the [View] menu or select the desired locale in the tool bar.
   Or, select the desired locale in the source view, and select "Switch" from the popup menu displayed upon a click of the right mouse button.



NOTE

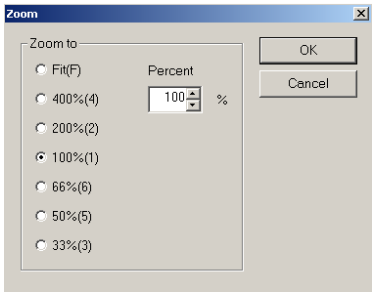
- The currently displayed locale is marked with a check mark in the icon.

5.10.7 Zoom

The editing window display is enlarged or reduced.
The scale can be selected in the range between 25% and 800%. Or select "fit" to enlarge or reduce according to the current window size.

1. Select [Zoom] from the [View] menu.

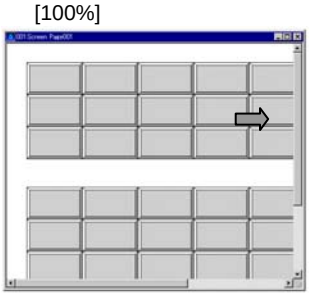
2. The [Zoom] dialog box is displayed.



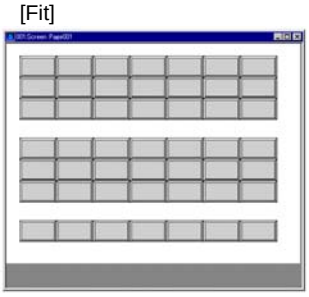
3. Select the scale and click on the [OK] button.

If [Fit] is selected, the window width fits the width of the editing window.

[100%]



[Fit]



The scale automatically changes so that the window is displayed in the full screen.

5.10.8 Refresh

Refresh the screen to erase garbage from the screen or display the current screen again.

1. Select [Refresh] from the [View] menu.

5.11 Creating View Frame

Use the view frame to switch a part of the displayed page.
The view frame consists of multiple view frame pages. Switch each view frame page according to the variable to switch the displayed image.
The following parts can be arranged in the view frame.

- Controls
- Figures

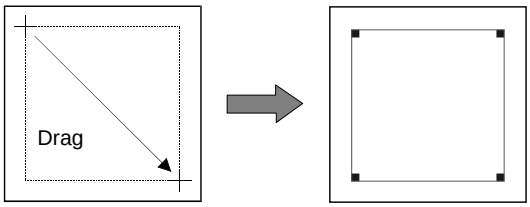
1. Select [Frame] from the [Control] menu or select the [Frame] button in the tool bar.

Tool bar

2. The cursor changes to the following shape.

3. Move the cursor to the starting point of the view frame.

4. Drag the cursor (move the mouse while holding down the left mouse button) to the end point of the display area of the view frame.



5. Select the view frame. The view frame properties are displayed.

Frame	
Frame Name	Frame00007
X	93
Y	86
WIDTH	323
HEIGHT	237
The number of the maximum frames	1
Edit Frame	0

6. Enter each item.

The setting items include the followings.

Item	Description
Frame Name	Specify the view frame name (up to 31 characters). The first character must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_).
X	0 to 2559
Y	0 to 1919
WIDTH	8 to 2560
HEIGHT	8 to 1920
The number of the maximum frames	Specify the maximum number of frames.
Edit Frame	Designate the frame page No. to be edited.

The setting is confirmed when entry is finished.

NOTE

- The view frame may not be created in another view frame.
  The frame of the view frame is highlighted with the reverse color of "Background color" that is specified in properties of panel or window.
- In the compilation method, the following project names cannot be used.
  - The combination of panel/window name and the view frame name is "SampleScreen"
  - The combination of the panel/window name and the view frame name is same as the project name.

  Example
  - Project name: TestMonitor
  - Panel/window name: Test
  - View frame name: Monitor

Next, the drawing method of each page of the view frame is described.

1. Double click on the view frame area.

2. The editing mode starts in the view frame.

3. Draw using the operation method similar to that of the regular page.

4. Edit other view frame pages.

5. Click on the area outside the view frame of the drawn page to return to the regular drawing mode.

5.11.1 Switching the View Frame

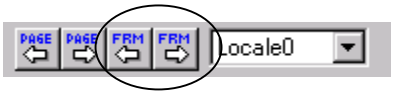
The method for switching the view frame page is described.

Switching to Previous or Next View Frame Page

Switch to the previous or next view frame page.

1. Select [Previous Frame] or [Next Frame] from the [View] menu or select [Previous Frame] or [Next Frame] button in the tool bar.

Tool bar



Switching to an Arbitrary View Frame Page

Display the desired view frame page.

1. Select a view frame and display view frame properties in the property window.

2. Designate the desired view frame page to be edited.

5.12 Importing a Page

The page import function enables to copy the desired page (panel/window) from another designated project to the currently editing project.

This function brings high productivity by reusing of panel/window for expanding the NC types and functions.

During the page import, the resource data, macro data and C++ source codes are imported in addition to the screen data.

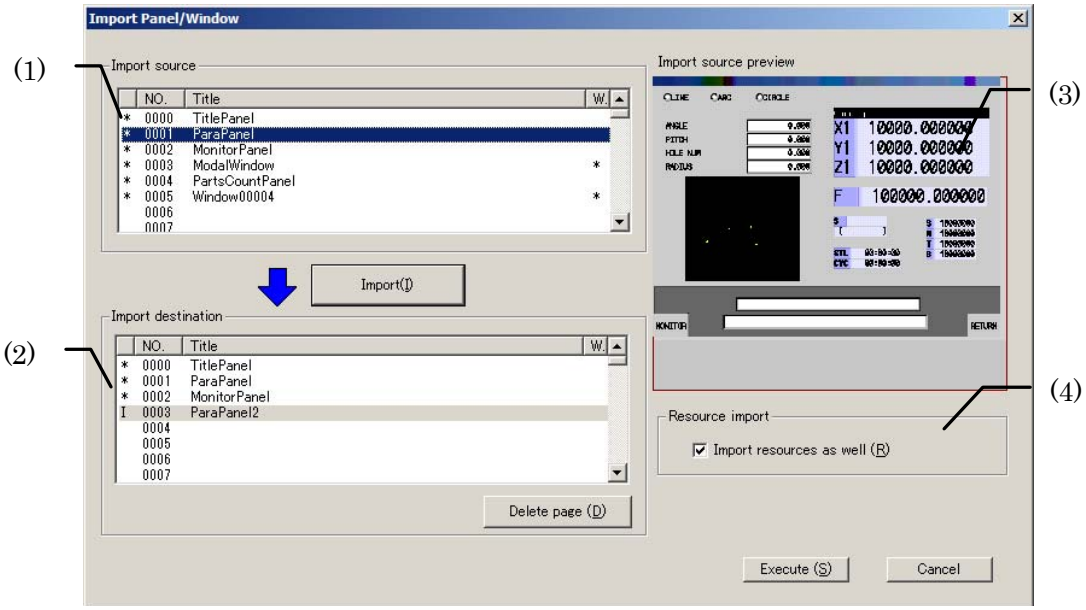
The page of the project created with NC Designer under the different version can be also imported.

5.12.1 Operation Screen

5.12.1.1 Page Import Dialog

The page import dialog enables to import the designated pages (panel/window) from a project of the import source location to the desired pages in the editing project. It is also possible to set whether to import the resource used in the pages.

Dialog Image



Display Item

The import dialog consists of the following items.

No.	Item	Detail
(1)	The page list of the project in the import source	Displays the page list of the project specified in the import source. "*" appears on the left side of the existing page. "*" appears on the right side of the window page.
(2)	The page list of the project in the import destination	Displays the page list of the currently edited project. "*" appears on the left side of the existing page. "I" appears on the imported page. "*" appears on the right side of the window page.
(3)	Import source preview	Previews the page selected in "The page list of the project in the import source" as in 1.
(4)	Resource import (R)	Sets whether to import the resource used in the page which has import setting. (Note 1) When [Import resources as well] is checked, the resource is imported, and the desired resource data name can be specified. (Note 2) When [Import resources as well] is not checked, the resource is not imported, and the different control from the import source may displayed.

Menu

The menus are displayed below.

No.	Item	Detail
(1)	Execute (S)	Saves the setting and closes the dialogue. Starts import.
(2)	Cancel	Cancels the setting and closes the dialogue.
(3)	× button	Cancels the setting and closes the dialogue.
(4)	Import (I)	Configures the import of the page specified in the import source to the page specified in the import destination. (Note 1) Pressing this button does not start import.
(5)	Delete page (D)	This is enabled when the cursor is on the page pointed by "I" in the import destination page list. The page to which import has been configured can be deleted from the project in the import destination.

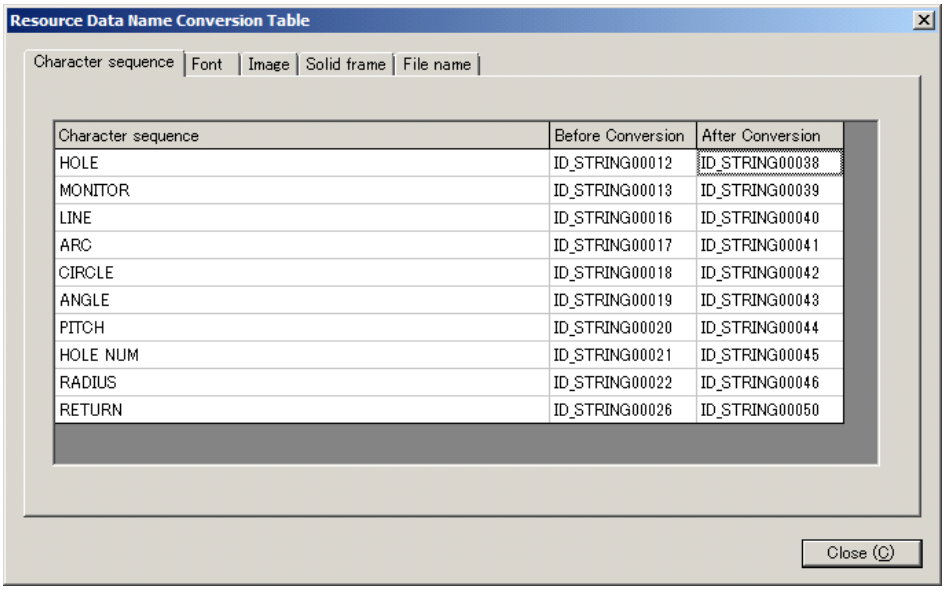
NOTE

- Even when no pages are available in the page list of the project in the import source or the page list of the project in the import destination, the page numbers 0000 to 0255 are still displayed in the [No.] field.
- The incorrect pages in the import destination are not displayed in the page list of the project in the import source.

5.12.1.2 Resource Data Name Conversion Table Dialog

The [Resource Data Name Conversion Table] dialog appears when import is performed when the [Import resources as well] is checked on the page import dialog. By editing the field in [After Conversion], you can rename the resource data after conversion to your desired name.

Dialog Image



Dialog Configuration

The Resource Data Name Conversion Table dialog is consisted of the following five tabs.

No.	Tab	Description
(1)	Character sequence	Displays a list of the character sequence resources to be imported.
(2)	Font	Displays a list of the font resources to be imported.
(3)	Image	Displays a list of the image resources to be imported.
(4)	Solid frame	Displays a list of the solid frame resources to be imported.
(5)	File name	Displays a list of the file name resources to be imported.

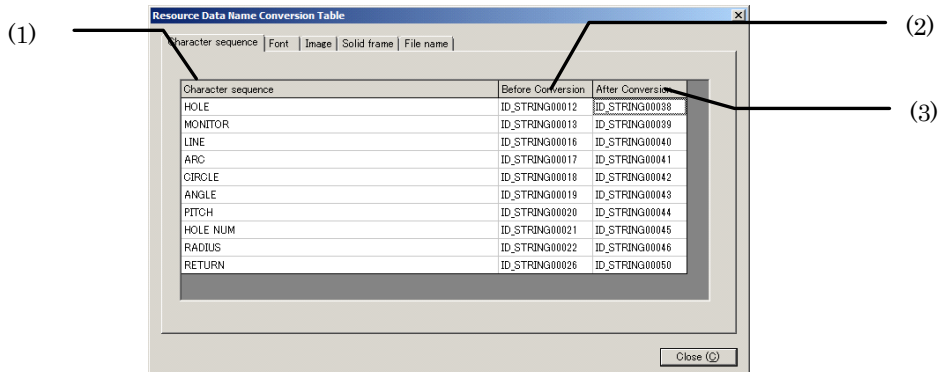
NOTE

- When multiple pages are imported, the list of all resource data to import is displayed.
- If there is no duplicated resource, even though [Import resource as well] is checked on the page import dialog, this dialog is not displayed. The [Open Panel/Window] dialog appears.

Display Item

The display items of each tab are described below.

(1) [Character sequence] tab

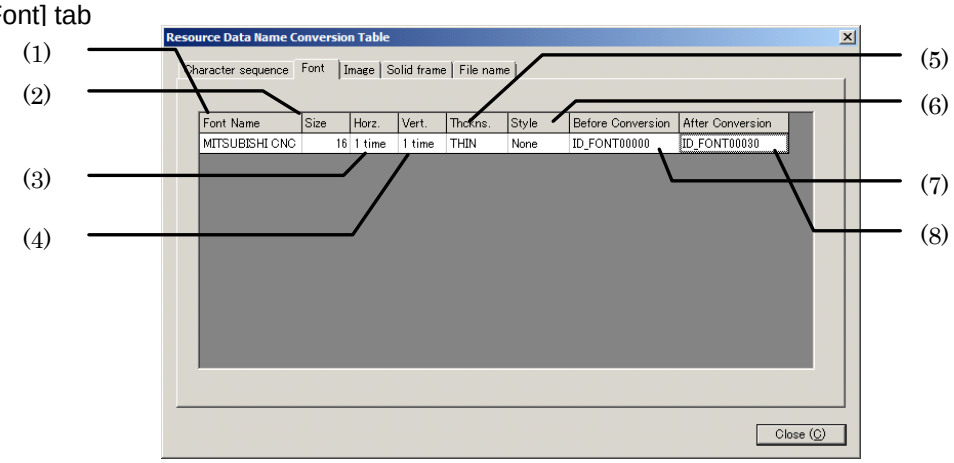


No.	Item	Description
(1)	Character sequence	Displays the character sequence of the selected locale.
(2)	Before Conversion	Displays the resource data name before the conversion
(3)	After Conversion	Displays the resource data name after the conversion.

NOTE

- The resource data name after the conversion is assigned from "ID_STRINGXXXXX(XXXXX: the total registered resource number plus 1)" in sequence.
- The character sequence properties of each control which use the character sequence resource are also converted.
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.4.1 Creating a New Character String Resource".
- If the same character sequence exists in the import source and import destination, it is registered with new resource data name. The character sequence and the resource data name in the import destination are maintained.
- If the number of locales in the import destination is smaller than the one in the import source, only the character sequence resource of the locales that exist in the import destination is copied. If the number of the locales in the import destination is larger than the one in the import source, the character sequence resource of Local0 is copied and imported for the insufficient locales.

(2) [Font] tab

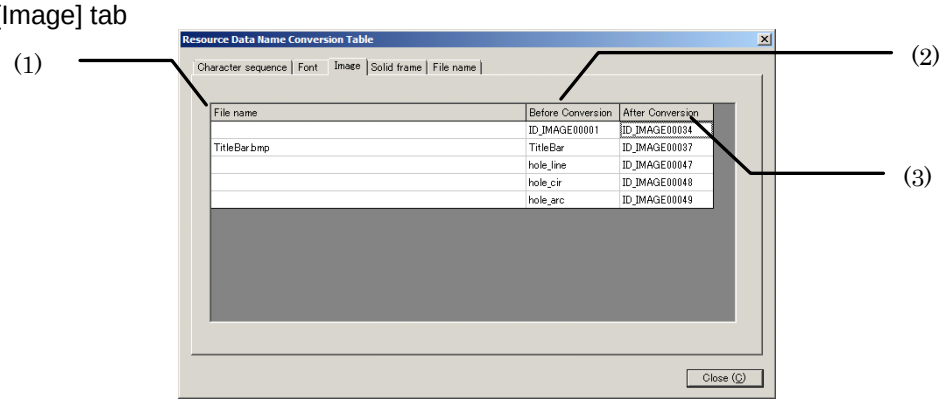


No.	Item	Description
(1)	Font Name	Displays the font name of the selected locale.
(2)	Size	Displays the font size of the selected locale.
(3)	Hoz.	Displays the horizontal zoom of the character size of the selected locale.
(4)	Vert.	Displays the vertical zoom of the character size of the selected locale.
(5)	Thckns.	Displays the character thickness of the selected locale.
(6)	Style	Displays the font style of the selected locale.
(7)	Before Conversion	Displays the resource data before conversion.
(8)	After Conversion	Displays the resource data after conversion.

NOTE

- The resource data name after the conversion is assigned from "ID_FONTXXXXX(XXXXX: the total registered resource number plus 1)" in sequence.
- The font properties of each control which use the font resource are also converted.
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.5.3 Specifying Font Resource".
- If the font resources of the same settings exist in the import source and the import destination, the font resource is not additionally registered. The information of the font resource which was not imported is saved in a log file.
- If the number of locales in the import destination is smaller than the one in the import source, only the character sequence resource of the locales that exist in the import destination is copied. If the number of the locales in the import destination is larger than the one in the import source, the character sequence resource of Local0 is copied and imported for the insufficient locales.
- When the font file which does not exist in the import destination（the file saved in the "C:\WINDOWS\Fonts" directory）is imported, the default font name of the import destination is displayed in the resource data name conversion table. The font file is not imported, therefore the dialog that appears after import may be different from the import source.

(3) [Image] tab

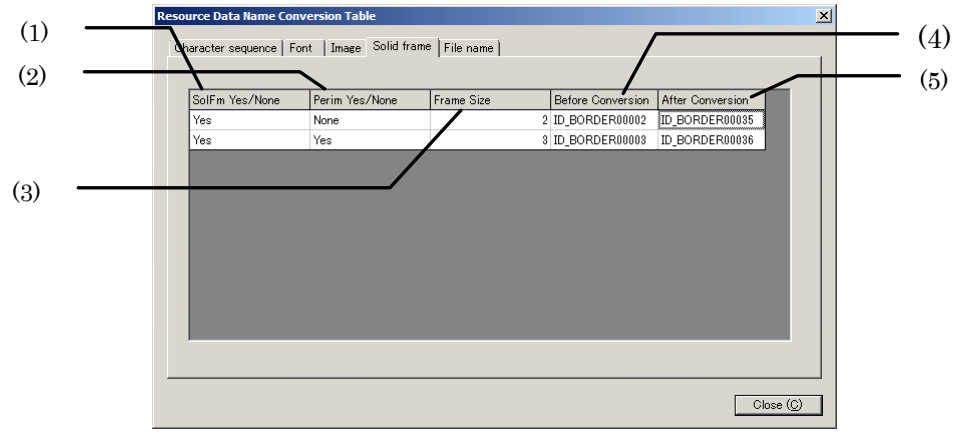


No.	Item	Description
(1)	File name	Displays the file name.
(2)	Before Conversion	Displays the resource data name before the conversion.
(3)	After Conversion	Displays the resource data name after the conversion.

NOTE

- The resource data name after the conversion is assigned from "ID_IMAGEXXXXX(XXXXX: the total registered resource number plus 1)" in sequence.
- When import is executed, the file is also imported to the project folder.
- The image properties of each control which use the image resource are also converted.
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.6.3 Specifying an Image File".
- If the same image exists in the import source and the import destination, it is registered with new resource data name. The image and the resource data name in the import destination are maintained.
- The empty file name is displayed in the following cases, but the file is not imported.
  (1) When the same file exists in the import source and the import destination.
  (2) When the file does not exist in the import source.
  The information of the image resource which was not imported is saved in a log file.

(4) [Solid frame] tab

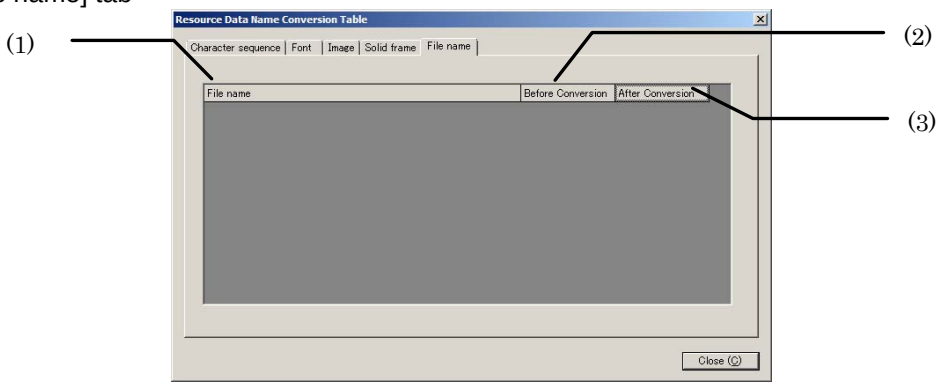


No.	Item	Description
(1)	SolFm Yes/None	Displays existence of a solid frame.
(2)	Perim Yes/None	Displays existence of a Perimeter line
(3)	Frame Size	Displays the frame size.
(4)	Before Conversion	Displays the resource name before conversion.
(5)	After Conversion	Displays the resource name after conversion.

NOTE

- The resource data name after the conversion is assigned from "ID_BORDERXXXXX(XXXXX: the total registered resource number plus 1)" in sequence.
- The solid frame properties of each control which use the solid frame resource are also converted.
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.7.3 Specifying a Solid Frame File".
- If the same solid frame exists in the import source and the import destination, it is registered with new resource data name. The solid frame and the resource data name in the import destination are maintained.

(5) [File name] tab



No.	Item	Description
(1)	File name	Displays the file name.
(2)	Before Conversion	Displays the resource data name before conversion.
(3)	After Conversion	Displays the resource data name after conversion.

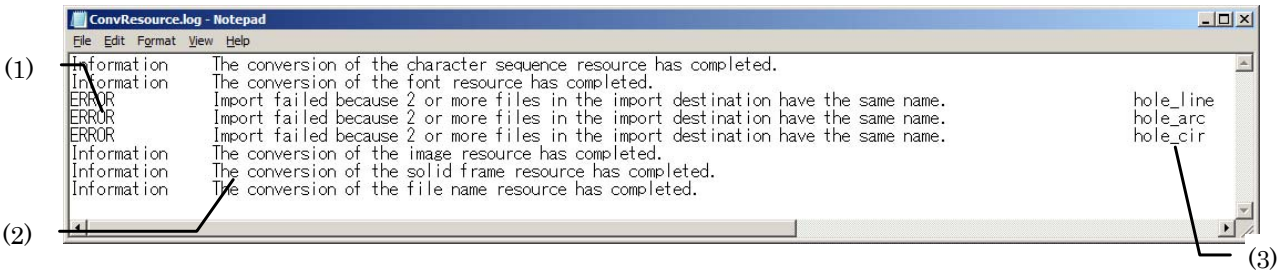
NOTE

- The resource data name after the conversion is assigned from "ID_FILEXXXXX(XXXXX: the total registered resource number plus 1)" in sequence.
- When import is executed, the file is also imported to the project folder
- The file name properties of each control which use the file name resource are also converted.
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.11.3 Specifying the File Name Resource".
- Only the [After Conversion] field can be edited. Up to 32 characters can be entered as a resource data name. For the usable characters, refer to "6.7.3 Specifying a Solid Frame File".
- If the same solid frame exists in the import source and the import destination, it is registered with new resource data name. The solid frame and the resource data name in the import destination are maintained.
- The empty file name is displayed in the following cases, but the file is not imported.
  (1) When the same file exists in the import source and the import destination.
  (2) When the file does not exist in the import source.
  The information of the image resource which was not imported is saved in a log file.

5.12.1.3 Message Log

The conversion state of the resource is output to a message log file.

Log File Output Image



Output Item

A log file is output in the following composition

No.	Item	Description
(1)	Message type	Displays the type of the message. The following three types are mainly used. Information: Each resource has converted WARNING：Not imported ERROR：Unable to import
(2)	Message	Displays messages.
(3)	Resource data name	Displays a resource data name.

Message List

The list of the messages to be output to a log file is described below.

Message
XXX Resource conversion has completed. XXX：A character sequence, font, image, solid frame or file name.
Did not import this font resource because another resource of the same property already exists.
Import failed because the specified file does not exist in the import source.
Import failed because 2 or more files in the import destination have the same name.

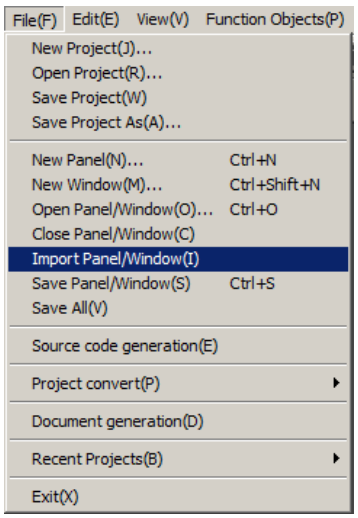
NOTE

- The log file (ConvResource.log) is saved in the folder at the same layer as the project information file (IPPfile).
- When the log file is in the following states, it is unable to write to the file.
  (1) The file is read-only.
  (2) Free disk space is insufficient.
  (3) The project was read from the network area and the file network connection was cut.
- Pressing the [Close] button on the resource name conversion dialog will display the resource conversion result on a message window.

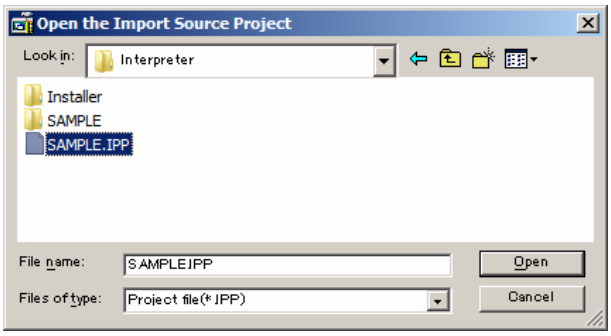
5.12.2 Flow of Import Operation

Selecting from File Menu while Editing Project

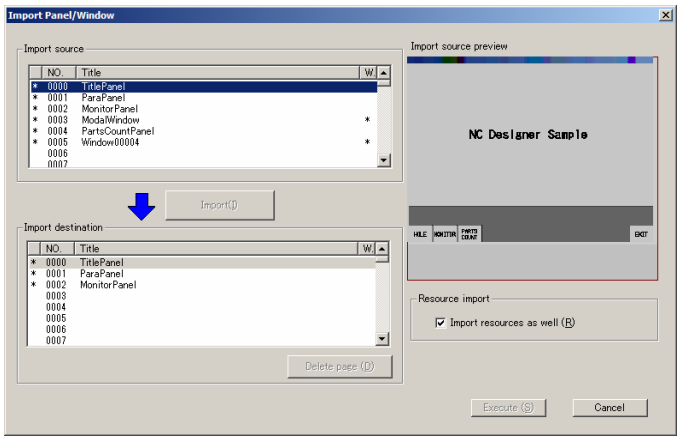
1. Select [Import Panel/Window] from the [File] menu.



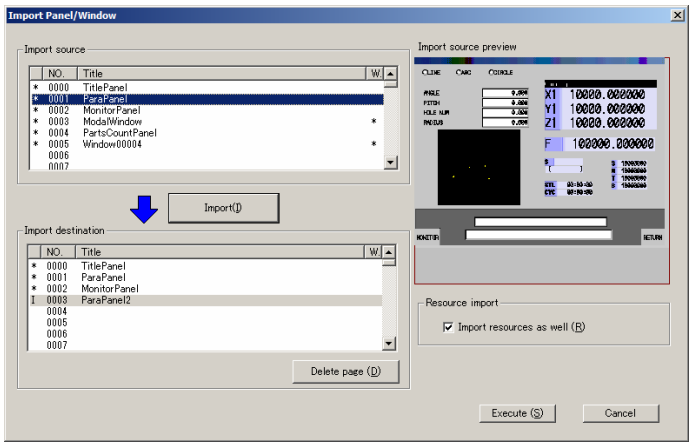
2. Specify the project in which contains the page to import, press the [Open] button.



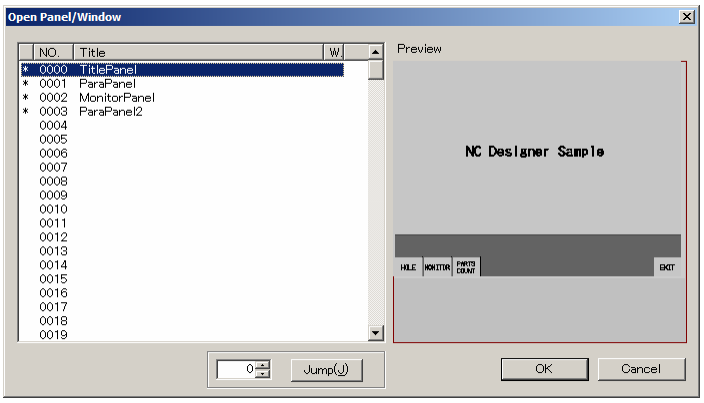
3. The [Import Panel/Window] screen appears.



4. After the setting, press the [Execute] button.



5. The [Open Panel/Window] dialogue appears. Select the panel/window to open, and press the OK button.



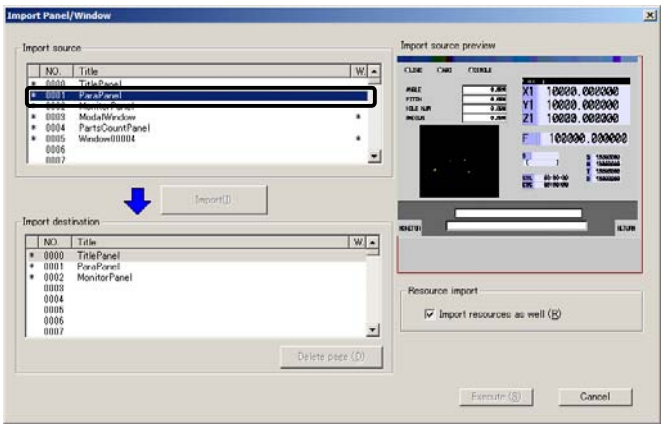
6. The selected panel/window appears.

NOTE

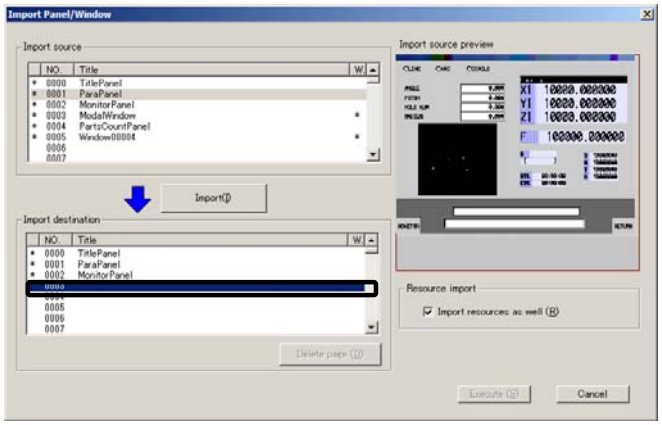
- If the project in the import destination is being referred to, or any one of the panel/window in the project in the import destination is being referred to, [Import Panel/Window] of the [File] menu is grayed out, and it cannot be selected.
- If the project in the import source is being referred to, or any one of the panel/window in the project in the import source is being referred to, an error dialogue of process execution appears when the project is specified in the "Open Project" dialog, and then the [Open] button is pressed. This disables the import process.

5.12.3 Selecting the Import Source/Destination

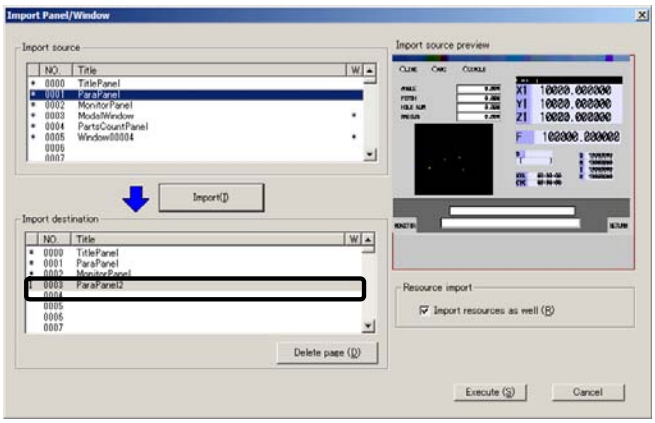
1. Select the page to import from the page list of the project in the import source.



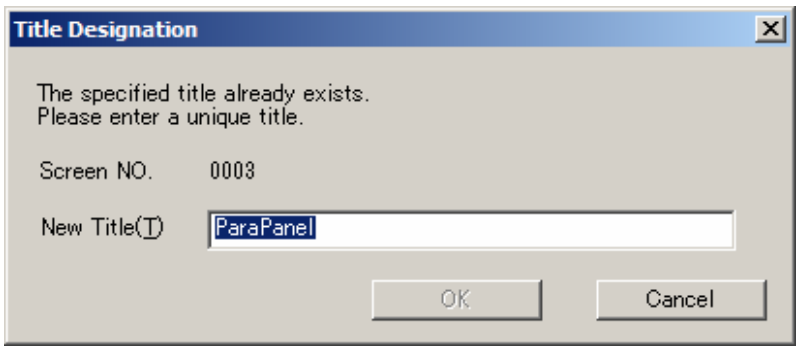
2. Select the location to import from the page list of the project in the import destination.



3. Pressing the [Import] button inserts the panel/window name.
   Repeat the step 1 to step 3 to import multiple pages.



If the page of the same name exists in the import destination, the Title Designation dialog appears. Specify another page name which does not overlap with other pages, then press the OK button.



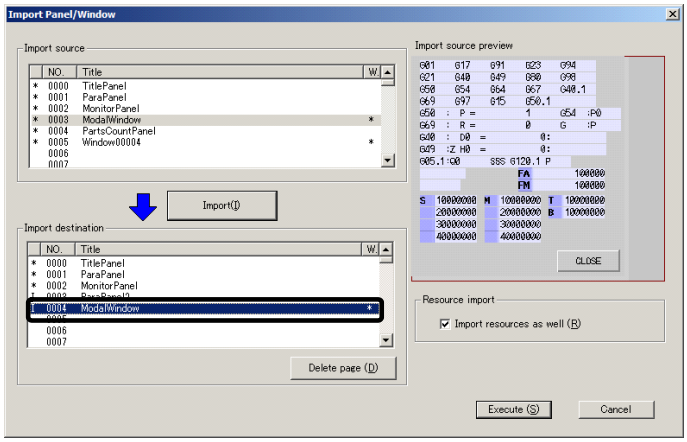
4. Pressing the [Execute] button executes the import for the designated panel/window.

NOTE

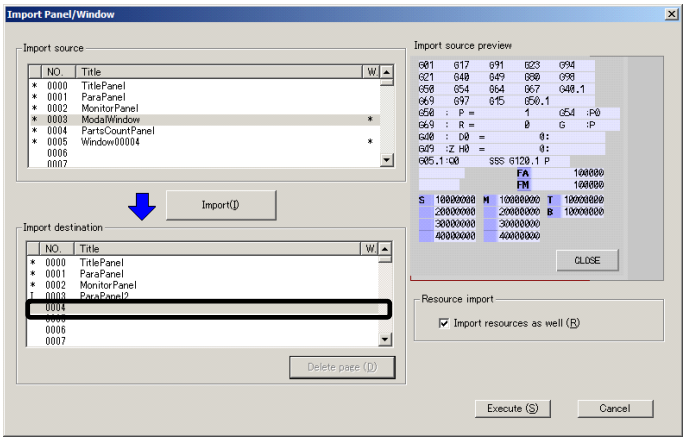
- "I" appears on the left end of the page that has the import setting.
- If the Cancel button is pressed after the page import is set, the import is not executed.
- The page that has the import setting can be deleted. Follow [Procedure how to delete pages] to delete the added pages.

5.12.4 Deleting a Page in the Import Source

1. Click the page to delete in the page list of the project in the import destination.



2. Pressing the [Delete page] button deletes the panel/window name.

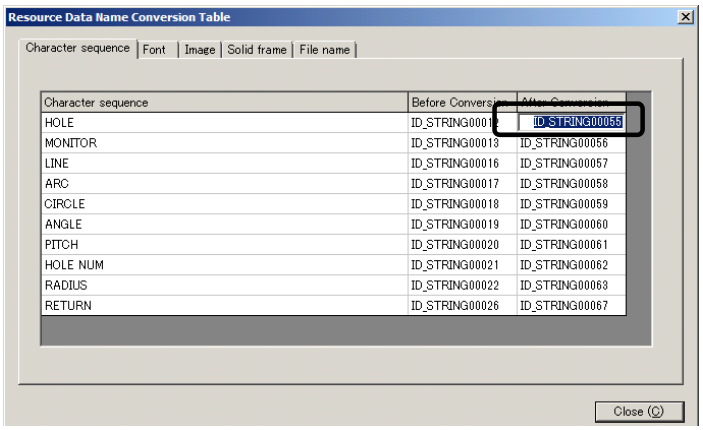


NOTE

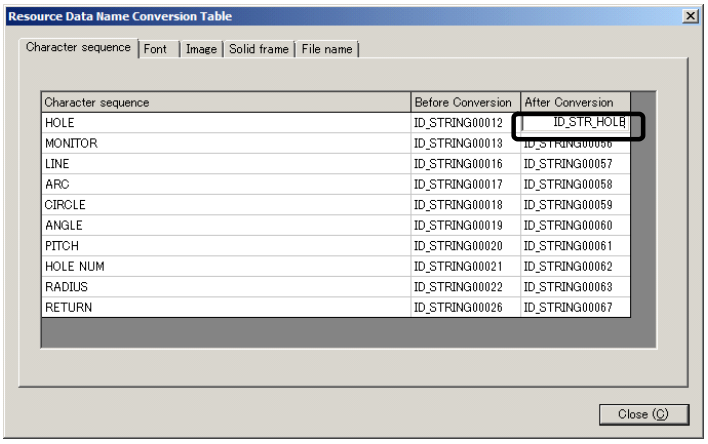
- The [Delete page] button is enabled when the cursor is moved to the page that has the import setting (the page marked with "I" appears on the left end).

5.12.5 Changing a Resource Data Name

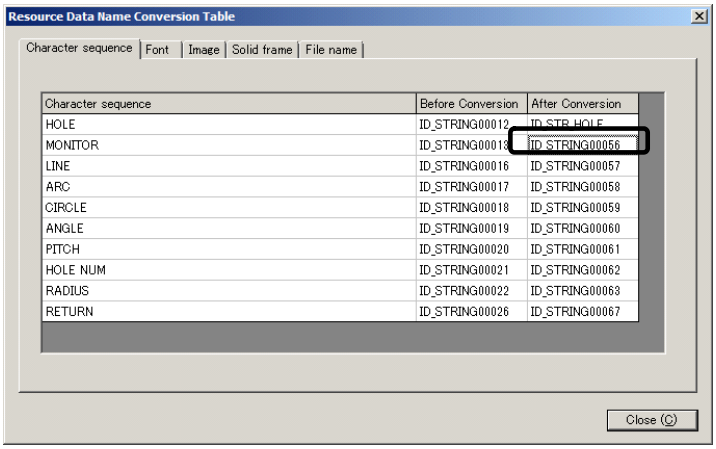
1. Click the position or press the ENTER key to change the resource data name.



2. Change the resource data name in the [After Conversion] field.



3. Pressing the Enter key changes the data name.



5.12.6 Restrictions

1. Painting-out pattern resources and line pattern resources are not imported.
2. The HTML file and PLC message file described in the macro are not imported. The file registered to the file name resource is imported. However, files are not imported in the following cases.
    1) When the same file exists in the import source and the import destination.
    2) No files exist in the import source.
3. The screen No. and the resource ID used in the macro are not converted.
4. If the screen files that have the same extension but different data formats exist in the import source, the page does not appear in the import source page list.

6. Registering the Resource

With NC Designer, the settings and data used commonly in the controls are registered in advance as resources.
In this section, the resource registration method is described.

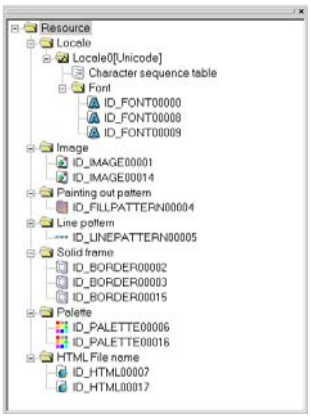
6.1 Resource

With NC Designer, the settings and data used commonly in the controls can be registered as resources.
The following data is handled as resources with NC Designer.

Item	Description
Locale	Character strings used for controls and window title, and font data
Image	Figure data (BMP or JPG) used for background image and picture controls
Painting out pattern	Filling pattern data used for controls and figures
Line pattern	Line pattern data used for figures
Solid frame	External frame specified for controls
Palette	Palette used for color designation
File name	File data used for HTML browser controls

6.2 Resource View

Resources registered in the currently opened project are displayed. Resources can be added, deleted or the settings of them can be changed.

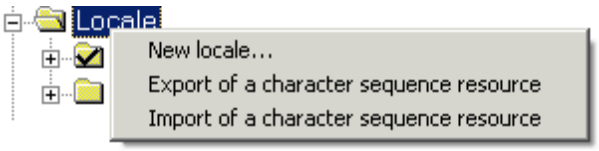


6.3 Locale

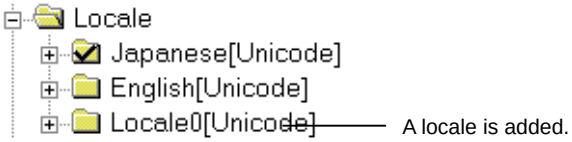
The set of a character string resource and font resource is called locale.
Up to 32 locales can be registered for each project.
Because two or more locales can be registered for each project, there is no need to divide a project according to the language, but screen data and settings can be handled as an integral group.

6.3.1 Creating a New Locale

1. Move the mouse cursor to "Locale" in the resource view and click the right mouse button. Select "New locale" from the displayed popup menu.



2. A new locale is registered.



6.3.2 Deleting a Locale

1. Move the mouse cursor to the desired locale name and select "Delete" from the popup menu displayed upon a click of the right mouse button.

6.3.3 Locale Setup

1. Double click on the desired locale name or move the mouse cursor to the local name and select "Properties" from the popup menu displayed upon a click of the right mouse button.

2. Locale setting items are displayed in the property window.

Item	Description
Locale name	Specify the locale name. The first character of the locale name must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The locale name must be within 16 characters.
Character code	Select the character code of the locale. If "Yes" is selected for "Unicode Correspondence" in project properties, the "character code" is available only in Unicode.
Language discerment character sequence	Select the language being used.

NOTE

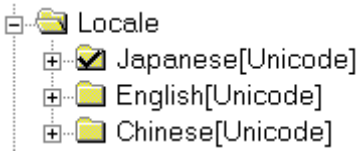
- The character code selected with "Character code" and the language discrimination character string that can be selected with "Language discerment character string" vary according to BSP being used.
- The locale of all languages has already been registered according to the language specification. Thus, do not change the setting of the locale.

6.3.4 Switching the Locale

The displaying locale can be switched on NC Designer.

1. Move the mouse cursor to the desired locale name and select "Switch" from the popup menu displayed upon a click of the right mouse button.

2. The locale is switched. The currently selected locale is marked with a check mark.



NOTE

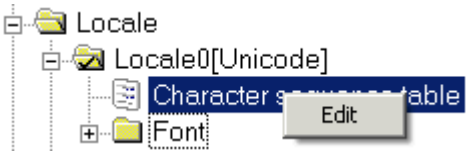
- When the lang parameter of NC is switched, the language is automatically switched because the interpreter method and the compilation method (DLL method) synchronize with NC. The compilation method (execution file method) should embed the language switch function in the source code.

6.4 Character Sequence Resource

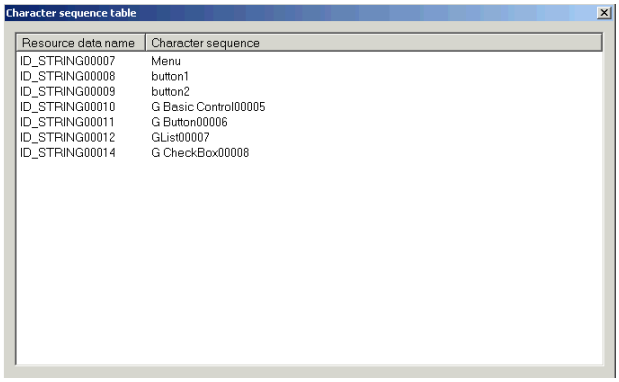
Register the character sequence set in the control.
Up to 5000 character sequence resources can be registered in the character string resource.

6.4.1 Creating a New Character String Resource

1. Move the mouse cursor to "Character sequence table" under "Locale" in the resource view and double click on it or select "Edit" from the popup menu displayed upon a click of the right mouse button.

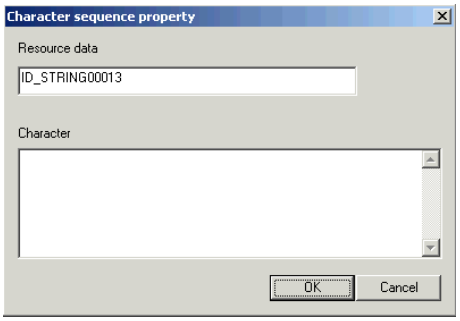


2. The [character sequence table] is displayed.



3. To register a new character sequence resource, select an arbitrary resource and select [Create] from the popup menu displayed upon a click of the right mouse button, or double click on the area where no resource is displayed.

The [character sequence property] dialog box is displayed.



4. Enter the "resource data name" and "character sequence" and click on the [OK] button.

NOTE

- The first character of the resource data name of the character string resource must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data must be within 32 characters.
- Up to 256 one-byte characters (each two-byte character is equivalent to two characters) can be entered as a character string.
- The character string resource is automatically created in the following case.
  - The character string displayed in the control is entered at control properties setup.
- If two or more locales are registered and a character string resource is registered to one of those locales, the character string resource data name and character strings are reflected on the character string table of the other locales.
- During multi-user development, the character string resource is controlled by a single user because it is common project data.

6.4.2 Editing or Creating Character String Resource

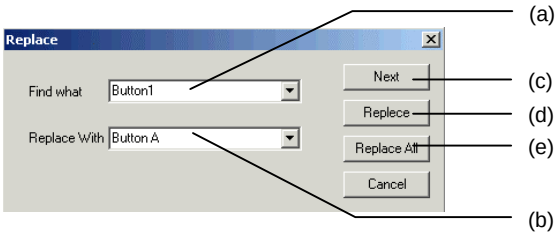
1. To edit the character string resource, select desired resource data and select [Edit] from the popup menu displayed upon a click of the right mouse button, or double click on the resource data.

2. The [Character String Properties] dialog box is displayed. Change the setting.

6.4.3 Replacing the Character String Resource

1. To replace registered character resource data, select the desired resource data and select [Replace] from the popup menu displayed upon a click of the right mouse button.

2. The [Replace] dialog box is displayed.



No.	Item	Description
(a)	Find what	Designate the character string to be replaced.
(b)	Replace With	Designate the character string to replace.
(c)	Next	Search for the character string designated in "Find what".
(d)	Replace	Replace the found character string each time a match is found.
(e)	Replace All	Replace all the target character string at once.

NOTE

◆ The target character string is the one included in the currently edited locale.

6.4.4 Deleting Character String Resource

1. To delete character string resource, select the desired resource data and select [Delete] from the popup menu displayed upon a click of the right mouse button.

NOTE

◆ If two or more locales are registered, the character string resource is deleted from the character string table of the other locales, too.

6.4.5 Importing or Exporting Character String Resource

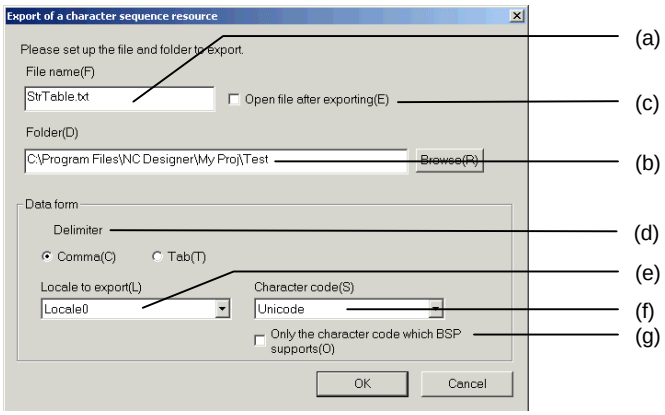
Export the character string resource in a text file and edit it with spreadsheet software or the like, then import the result to enter the character string resource at a time.

Exporting Character String Resource

Export the character string resource into a text file.

1. Select [Export Character String Resource] from the [Tool] menu. Or move the cursor to "Locale" in the resource view and select "Export of a character sequence resource" from the popup menu displayed upon a click of the right mouse button.

2. The [Export of a character sequence resource] dialog box is displayed.



No.	Item	Description
(a)	File name	Designate the export file name (default file name: StrTable.txt).The file name is up to 64 one-byte characters (each two-byte character is equivalent to two characters).
(b)	Folder	Designate the destination of the file. (The maximum number of characters of the folder and file names is 200.)
(c)	Open file after exporting	Place a check mark here to launch the application associated with the extension of the file designated in the "file name" field and open the file.
(d)	Delimiter	Select either comma or tab as a delimiter used in the export file.
(e)	Locale to export	Select the export locale among each locale and "all locales."
(f)	Character code	Select the character code corresponding to the export file. Select among Unicode, character code supported by BSP, and language identification character string of Windows.
(g)	Only the character code which BSP supports	Place a check mark here to limit the character code selected at "character code" to only the character code of the language supported by BSP.

3. Click on the "OK" button to start to export.

NOTE

- Before exporting the character string resource, save the project ([File] - [Save Project]).
- The language identification character string of Windows that can be selected at "character code" includes the followings.

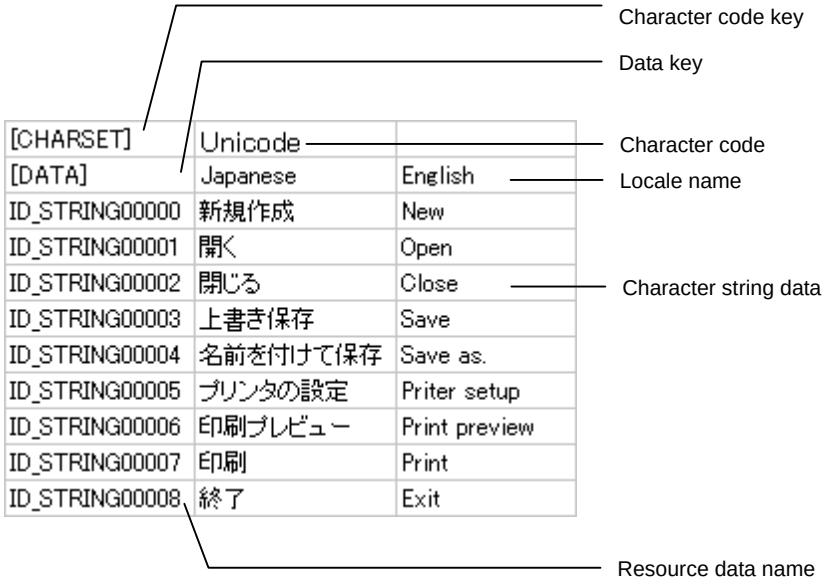
Language identification character string	Main language	Auxiliary language	Character code
chinese	Chinese	Chinese	big5
chinese-simplified	Chinese	Chinese (simplified)	gb2312
chinese-traditional	Chinese	Chinese (traditional)	big5
czech	Czech	Czech	windows-1250
danish	Danish	Danish	windows-1252
dutch	Dutch	Dutch (default)	windows-1252
english	English	English (default)	windows-1252
finnish	Finnish	Finnish	windows-1252
french	French	French (default)	windows-1252
german	German	German (default)	windows-1252
greek	Greek	Greek	windows-1253
hungarian	Hungarian	Hungarian	windows-1250
icelandic	Icelandic	Icelandic	windows-1252
italian	Italian	Italian (default)	windows-1252
japanese	Japanese	Japanese	Shift-JIS
korean	Korean	Korean	ks_c_5601-1987
norwegian	Norwegian	Norwegian (default)	windows-1252
polish	Polish	Polish	windows-1250
portuguese	Portuguese	Portuguese (default)	windows-1252
russian	Russian	Russian (default)	windows-1251
slovak	Slovak	Slovak	windows-1250
spanish	Spanish	Spanish (default)	windows-1252
swedish	Swedish	Swedish	windows-1252
turkish	Turkish	Turkish	windows-1254

Editing Text File

Use spreadsheet software or text editor to edit the exported text file. The character code name, resource data name of the character string resource, and character string data in the text file can be edited.

1. Use spreadsheet software or text editor to open the exported text file.

   If the "Open file after exporting" checkbox is marked when the file is exported, spreadsheet software or the like is launched automatically to load the generated text file.



[CHARSET]	Unicode	
[DATA]	Japanese	English
ID_STRING00000	新規作成	New
ID_STRING00001	開く	Open
ID_STRING00002	閉じる	Close
ID_STRING00003	上書き保存	Save
ID_STRING00004	名前を付けて保存	Save as.
ID_STRING00005	プリンタの設定	Priter setup
ID_STRING00006	印刷プレビュー	Print preview
ID_STRING00007	印刷	Print
ID_STRING00008	終了	Exit

2. Edit the text file.

NOTE

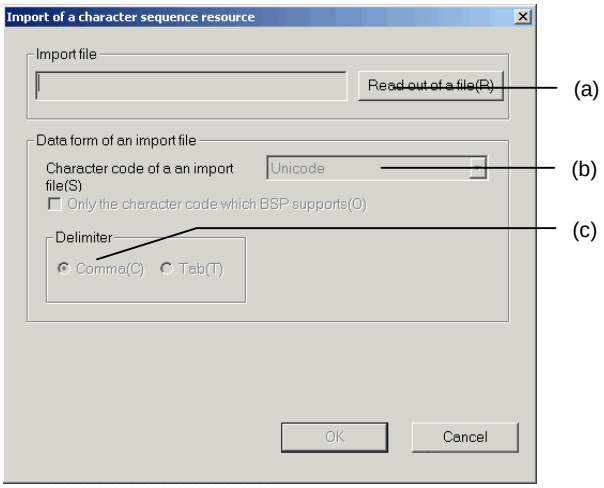
- Do not delete [CHARSET] and [DATA].
- The first character of the locale and resource data names must be a one-byte letter (A to Z or a to z). The second and later characters must be letters, numbers or underscores (_). The locale name must be within 16 characters.
- The character string registered in the character string resource is within 256 one-byte characters (each two-byte character is equivalent to two characters).
- When editing the text file to be imported, enclose each character string including a comma, tab, line feed and carrier return codes, or double quotation mark in double quotation marks (" "). Specify two repetitive double quotation marks for each double quotation mark.
  Example
  - To import [AB, CD], enter [AB, CD].
  - To import [AB" CD], enter [AB"" CD].

Importing Character Sequence Resource

Import the edited text file.
An error check is performed when the file is imported.

1. Select [Import of a character sequence resource] from the [Tool] menu. Or move the mouse cursor to "Locale" in the resource view, and select "Import of a character sequence resource " from the popup menu displayed upon a click of the right mouse button.

2. The "Import of a character sequence resource" dialog box is displayed.



No.	Item	Description
(a)	Import file	Designate the file to be imported. Click on "Load File" and designate the desired file in the displayed file selector.
(b)	Dataform of an import file	Designate the data format of the file to be imported.
	Character code of an import file	Select the character code of the file to be imported. Place a check mark at "Only the character code which BSO supports" to select only the character codes supported by BSP.
(c)	Dilimiter	Select the delimiter of the file to be imported.

3. Click on the OK button to start to import.

NOTE

- If a duplicate resource data name is found in the importing text file, the duplicate data will overwrite the original data in the application. Check for duplicate resource data name before starting to import.
- The character string resource can be imported only if the project is opened with the edit flag.
- If importation is interrupted during importation, the character string data having been imported up to the timing is reflected on the character string resource.

Error Check

If an error is detected during importation, the message view displays an error list.

Message	Remedy
Invalid locale name "XXX" was discovered by the file under import. The information on this column is not imported.	Correct the locale name.
The character sequence with a blank locale name was discovered by the file under import. The information on this column is not imported.	Correct the locale name.
Invalid resource data name "XXX" was discovered by the file which is under import. The information on this line is not imported.	Correct the resource data name.
The column with a blank resource data name was discovered by the file under import. The information on this line is not imported.	Correct the resource data name.
The line to which a locale name overlaps the file under import was discovered. The information on this column is not imported.	Enter an unused locale name.
The character sequence to which length exceeds 256 characters in the file under import was discovered. It is not imported after 256 characters of this character sequence.	Reset the character string registered in the character string resource so that it is within 256 one-byte characters (each two-byte character is equivalent to two characters).
Since the number of the maximum registration of a locale name is 32, it is ignored after it.	The number of locale names must be within 32.
Since the number of the maximum registration of the character sequence resource ID is 5000, the character sequence resource after it is disregarded.	The maximum number of registered resources must be within 5000.
In the file under import, since a data key does not exist, a file cannot be imported.	Add a data key ([DATA]) in the imported file.

NOTE

- The first character of the locale and resource data names must be a one-byte letter (A to Z or a to z). The second and later characters must be letters, numbers or underscores (_).The locale name must be within 16 characters.
- Show a preview in the [Import Character String] dialog box to display errors in red. Check for errors before starting to import.

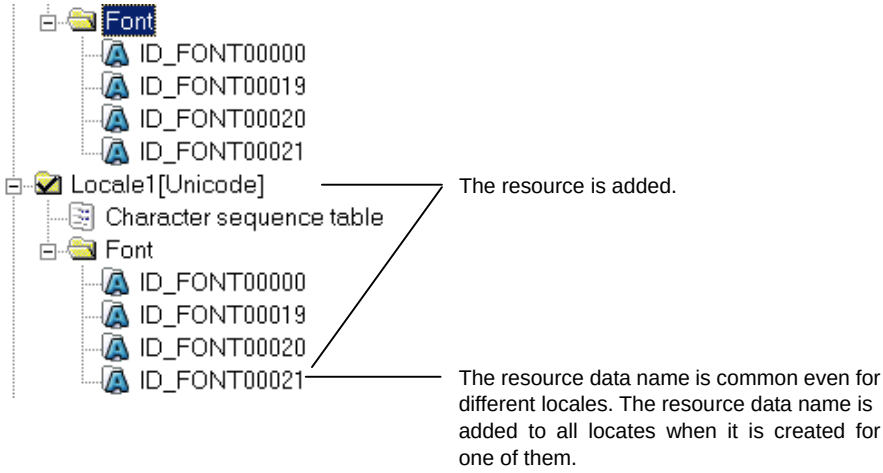
6.5 Font Resource

Register the font, style and other character attributes to be specified for the control.
Up to 5000 font sources can be registered.

6.5.1 Creating a New Font Resource

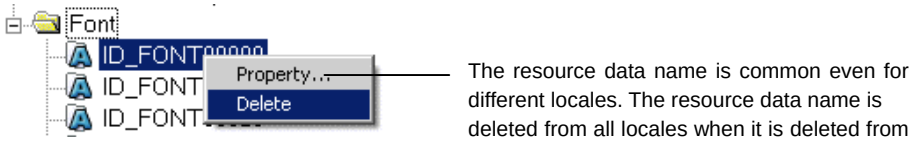
1. Move the mouse cursor to "Font" in the resource view and select "New font" from the popup menu displayed upon a click of the right mouse button.

2. The new font resource is registered.



6.5.2 Deleting Font Resource

1. Move the mouse cursor to the font resource data name to be deleted, and select "Delete" from the popup menu displayed upon a click of the right mouse button.



6.5.3 Specifying Font Resource

1. Double click on the desired font resource data name or move the mouse cursor to the font resource data name and select "Property" from the popup menu displayed upon a click of the right mouse button.

2. The font resource setting items are displayed in the property window.

Item	Description
Resource data name	Specify the resource data name. The first character of the resource data name must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data name must be within 32 characters.
Font name	Specify the font name.
Font size	Specify the font size. The setting is valid if the selected font is a vector font.
Zoom horizontal	Specify the horizontal zoom of the character size. The setting is valid if the selected font is a raster font.
Zoom vertical	Specify the vertical zoom of the character size. The setting is valid if the selected font is a raster font.
Thickness	Select the character thickness among "THIN," "NORMAL" and "BOLD."
Font style	Select the character style among "None" and "Italics."

NOTE

◆ The font name that can be selected depends on the BSP that is used.

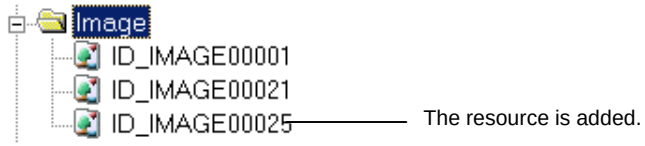
6.6 Image Resource

Register image data used for controls or background images.
The image data that can be registered is either BMP or JPG files.
Up to 5000 image resources can be registered.

6.6.1 Creating a New Image Resource

1. Move the mouse cursor to "Image" in the resource view and select "New image" from the popup menu displayed upon a click of the right mouse button.

2. The new image resource is registered.



6.6.2 Deleting an Image Resource

1. Move the mouse cursor to the desired image resource data name and select "Delete" from the popup menu displayed upon a click of the right mouse button.

6.6.3 Specifying an Image File

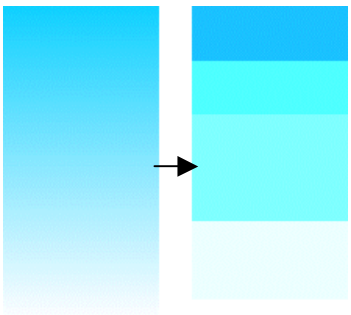
1. Double click on the desired image resource data name or move the mouse cursor to the image resource data name and select "Property" from the popup menu displayed upon a click of the right mouse button.

2. The image resource setting items are displayed in the property window.

Item	Description
Resource data name	Specify the resource data name. The first character of the resource data must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data name must be within 32 characters.
File name	Specify an image file. Click on the [...] button to display a file selector. Specify an image file. The file is either a BMP or JPG file.
Palette Specification	Select whether or not to designate a palette used for the image resource. To designate the palette, select "Yes." To refrain from designating the palette, select "None." The setting is valid only for the "Direct Color System" color setting method.
Palette	Select the palette used for the image resource from the palette resource. The setting is valid only for the "Direct Color System" color setting method.
The number of Colors	Select the number of display colors of the image resource among "2 Colors", "4 Colors", "16 Colors" and "256 Colors".

NOTE

- Use the palette to reduce the data size (ROM consumption).
- If the "index color system" is selected as a color setting method, the system palette is applied without using the designated palette. While the number of colors follows the image resource properties setting, it must be within the number of system colors specified at [Color Setup] in the [Setting] menu. If the "direct color system" is selected as a color setting method, property settings (palette designation, palette and number of colors) of each image resource are followed.
- When the palette is used, the number of colors becomes smaller and fine gradation may be disabled.
  Example: Specifying a 256-color palette resource to a 24-bit color image resource



24-bit color 256-color palette

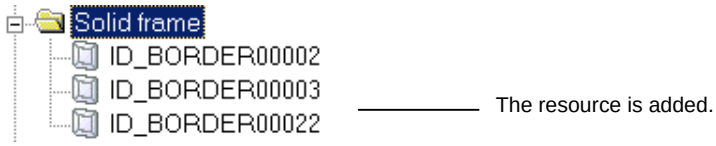
6.7 Solid Frame Resource

Register the solid frame used for controls.

6.7.1 Creating a New Solid Frame Resource

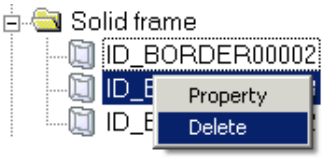
1. Move the mouse cursor to "Solid frame" in the resource view and select "New solid frame" from the popup menu displayed upon a click of the right mouse button.

2. The new solid frame resource is registered.



6.7.2 Deleting a Solid Frame Resource

1. Move the mouse cursor to the desired solid frame resource data name and select "Delete" from the popup menu displayed upon a click of the right mouse button.

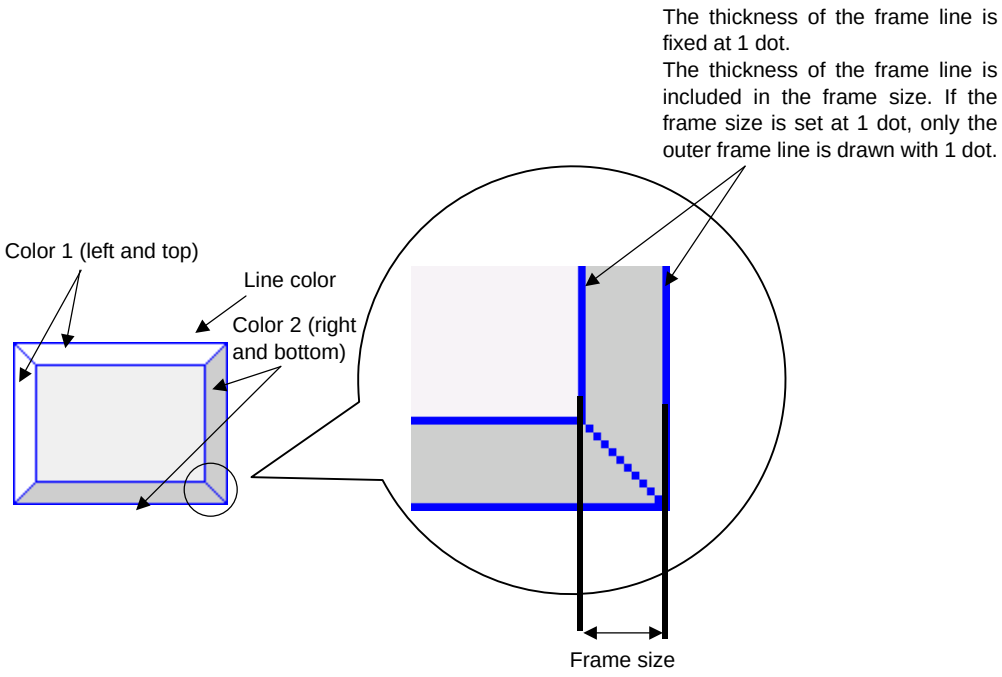


6.7.3 Specifying a Solid Frame File

1. Double click on the desired solid frame resource data name or move the cursor to the solid frame resource data name and select "Property" from the popup menu displayed upon a click of the right mouse button.

2. The solid frame resource setting items are displayed in the property window.

Item	Description
Resource data name	Specify a resource data name. The first character of the resource data name must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data name must be within 32 characters.
Existence of a solid frame	Select the existence of the solid frame between [Yes] and [None].
Color 1	Specify the color of the upper left side of the solid frame.
Color 2	Specify the color of the lower right side of the solid frame.
Line color	Specify the color of the frame line.
Frame size	Specify the size of the solid frame in dots (1 to 960).

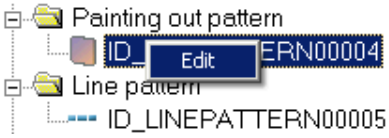


6.8 Filling Pattern Resource

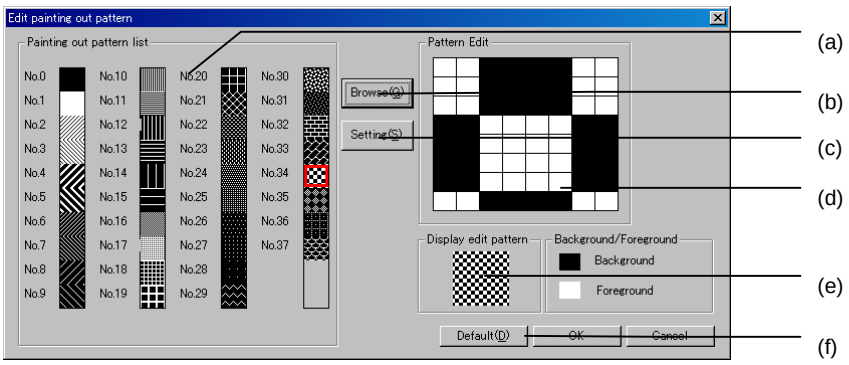
Register the filling pattern used for controls and figures.
38 filling patterns are provided in the initial state.

6.8.1 Editing the Filling Pattern

1. Move the mouse cursor to the resource data name of "Painting out pattern" in the resource view and double click on it or select "Edit" from the popup menu displayed upon a click of the right mouse button.



2. The [Pattern Edit] dialog box is displayed. Edit the pattern.



No.	Item	Description
(a)	Painting out pattern list	Select the filling pattern.
(b)	Browse	Copy the currently selected filling pattern to the pattern editing area.
(c)	Setting	Specify the filling pattern being edited in the current pattern edit area to the selected filling pattern list.
(d)	Pattern Edit	Edit the filling pattern in dots. Each time the mouse button is clicked, the foreground and background are switched over.
(e)	Display edit pattern	Display a sample of the edited filling pattern.
(f)	Default	Restore the default filling pattern list.

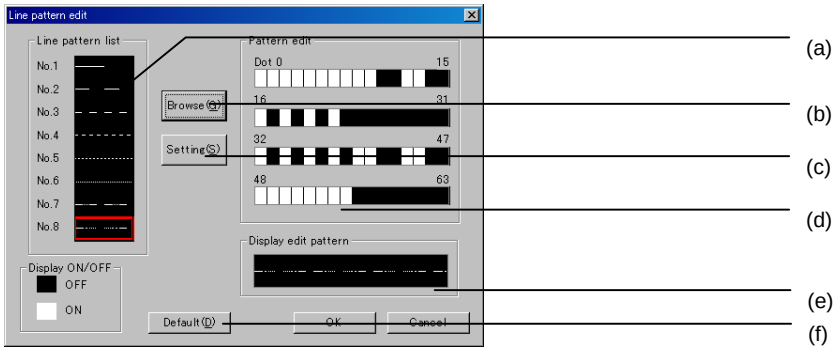
6.9 Line Pattern Resource

Register the line pattern used for figures.
Eight line patterns are provided in the initial state.

6.9.1 Editing the Line Pattern

1. Move the mouse cursor to the resource data name of "line pattern" in the resource view, and double click on it, or select "Edit" from the popup menu displayed upon a click on the right mouse button.

2. The [Line pattern edit] dialog box is displayed. Edit the pattern.



No.	Item	Description
(a)	Line pattern list	Select the line pattern.
(b)	Browse	Copy the currently selected line pattern to the pattern edit area.
(c)	Setting	Specify the line pattern edited in the current pattern editing area to the selected pattern list.
(d)	Pattern edit	Edit the line pattern in dots. Each time the mouse button is clicked, ON and OFF are switched over. The left end, center and right end patterns are displayed, starting at the top.
(e)	Display edit pattern	Display a sample of the line pattern being edited.
(f)	Default	Restore the default line pattern list.

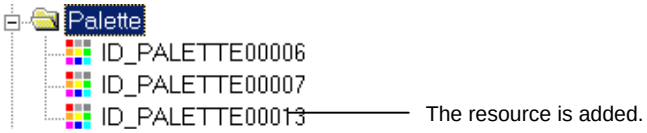
6.10 Palette Resource

Register the palette used for color designation of control and so on.
Up to 5000 palette resources can be registered.

6.10.1 Creating a New Palette Resource

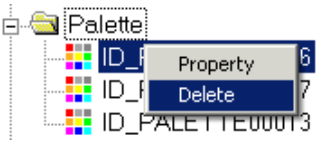
1. Move the mouse cursor to "Palette" in the resource view, and select "New palette" from the popup menu displayed upon a click of the right mouse button.

2. The new palette resource is registered.



6.10.2 Deleting the Palette Resource

1. Move the mouse cursor to the desired palette resource data name, and select "Delete" from the popup menu displayed upon a click of the right mouse button.



6.10.3 Palette Setting

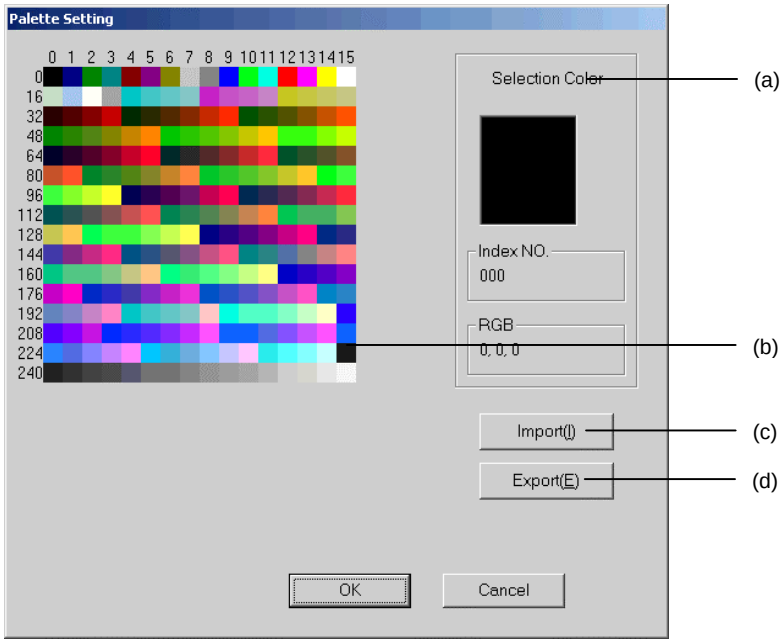
1. Double click on the desired palette resource data name, or move the mouse cursor to the desired palette resource data name, and select "Property" from the popup menu displayed upon a click of the right mouse button.

2. The palette resource setting items are displayed in the property window.

Item	Description
Resource data name	Specify the resource data name. The first character of the resource data name must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data name must be within 32 characters.
Palette	Specify the palette. Click on the [...] button to open the [Palette Setting] dialog box.

6.10.4 Editing the Palette

1. Click on the [...] button at [Palette]. The [Palette Setting] dialog box is displayed.

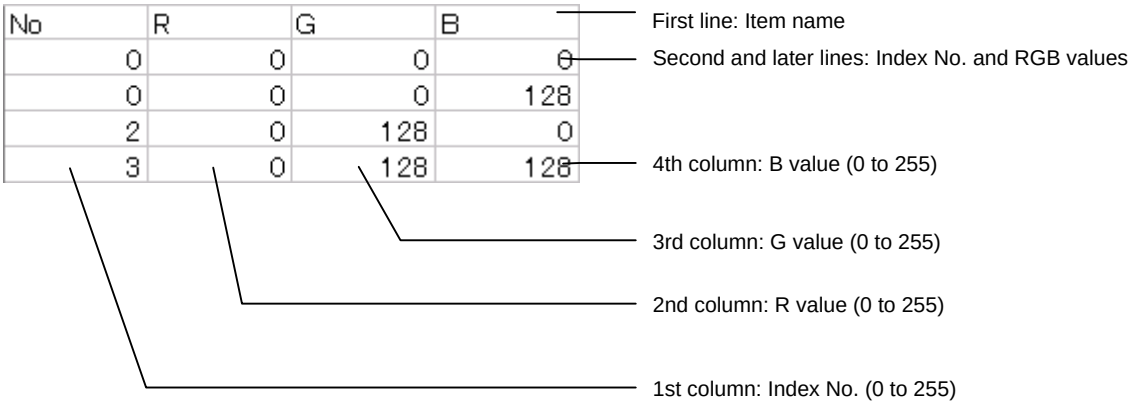


No.	Item	Description
(a)	Selection Color	The index number of the selected color and the RGB values are displayed. The RGB values are displayed in decimals.
(b)	Color Matrix	Specify the color corresponding to each index number. Click on each cell and specify the desired color in the [Color] dialog box.
(c)	Import	Import color setting from a CSV file. Specify the source CSV file.
(d)	Export	Export the current setting to a CSV file. Specify the destination CSV file.

6.10.5 Importing/Exporting Palette Setting

Edit the CSV file created with exported palette setting, using spreadsheet software. Import the result to change the color at a time.

The CSV file is in the following format.



NOTE

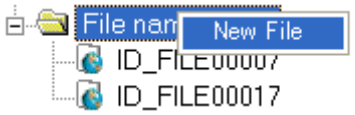
- Specify values in decimals.
- If an error is caused during importation, importation is interrupted. Check and correct the followings and import again.
  - Incorrect index number and/or RGB value (out of permissible range, character string or blank)
  - Duplicate index number

6.11 File Name Resource

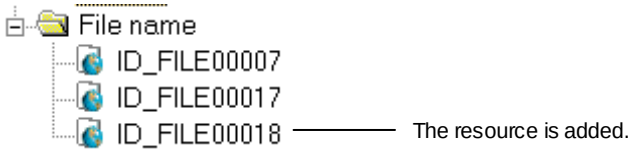
Register the file name displayed first when the browser control is operated.
Up to 5000 files can be registered.

6.11.1 Creating a New File Name Resource

1. Move the mouse cursor to "File name" in the resource view, and select "New File" from the popup menu displayed upon a click of the right mouse button.

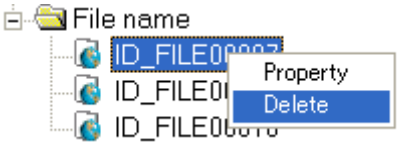


2. The new file name resource is registered.



6.11.2 Deleting the File Name Resource

1. Move the mouse cursor to the desired file name resource data name, and select "Delete" from the popup menu displayed upon a click of the right mouse button.



6.11.3 Specifying the File Name Resource

1. Double click on the desired file name resource data name, or move the mouse cursor to the file name resource data name, and select "Property" from the popup menu displayed upon a click of the right mouse button.

2. The file name resource setting items are displayed in the property window.

Item	Description
Resource data name	Specify the resource data name. The first character of the resource data name must be a one-byte letter (A to Z or a to z). The second and later characters must be one-byte letters, numbers or underscores (_). The resource data name must be within 32 characters.
File name	Select the file. Click on the [...] button to display a file selector. Specify the file displayed first.

7. Creating Controls

Each control and property settings are described in this section.
The control is an object having the following functions.

- Expression of GUI operation and retention of operation state
- Notification of GUI operation to user program

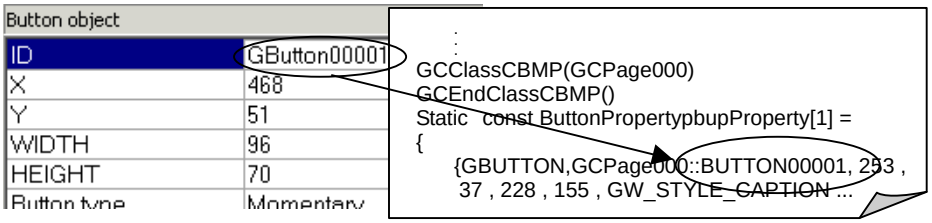
7.1 Common Functions of Controls

7.1.1 Control Name

Specify the name of the control

Item	Description
ID	Specify the name of the control. The first character of the control name must be a letter (A to Z or a to z) or underscore (_). The second and later characters must be letters, numbers or underscores (_). The control name must be within 31 characters.

The control name specified here becomes a variable name during source code generation.



NOTE

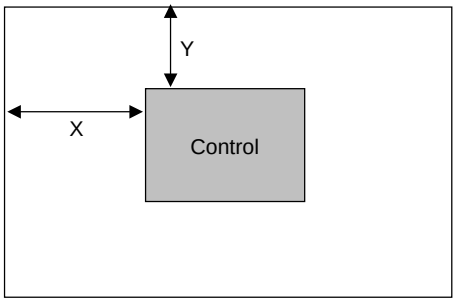
- One of the following control names is automatically given to a new control.
  Basic control object : GBasicControlxxxxx
  Button object : GButtonxxxxx
  Check box object : GCheckBoxxxxxx
  Edit control object : Geditxxxxx
  HTML browser object : GHtmlBrowserxxxxx
  Label object : GLabelxxxxx
  List box object : GListxxxxx
  Picture object : GPicturexxxxx
  Progress bar object : GProgressBarxxxxx
  Radio button object : GRadioButtonxxxxx
  Text box object : GTextBoxxxxxx
  Scroll bar object : GscrollBarExxxxxx
  Input box : GInputBoxxxxxx
  Ten-key : GSoftKeyxxxxx
  NC data text box: GNCDataTextBoxxxxxx
  PLC button : GNCPLCButtonxxxxx
  PLC extension button: GNCPLCExButtonxxxxx
  PLCtextbox : GNCPLCTextBoxxxxxx
  PLC message: GNCPLCMessagexxxxx
  Table : GNCTablexxxxx
  Counter : GNXCounterxxxxx
  CycleTime : GNXCycleTimexxxxx
  Feedrate : GNXFeedratexxxxx
  GModal M : GNXGModalxxxxx
  GModal L : GNXGModal_Lxxxxx
  GModal Simple : GNXGModalSimplexxxxx
  LoadMeter : GNXLoadMeterxxxxx
  MSTB : GNXMSTBxxxxx
  ONB : GNXONBxxxxx
  ProgramBuffer : GNXPrgBuffxxxxx
  SPCommand : GNXSPCommandxxxxx
  Menu : GNXMenuxxxxx
  FileInOut : GNXFileInOutxxxxx
  AlarmMessage : GNXAlarmMessagexxxxx
  MonitorStatus : GNXMonitorStatusxxxxx
  Time : GNXTimexxxxx
  xxxxx: Between 00000 and 99999. Automatically assigned from the smallest number. The automatically given control name can be changed later.
- Specify the control name while avoiding duplication with other controls in the same page.
- If there are duplicate control names, source code generation is not in order. Use the error check function to check for duplication.

7.1.2 Position/Size

Specify the displaying position and size of the control

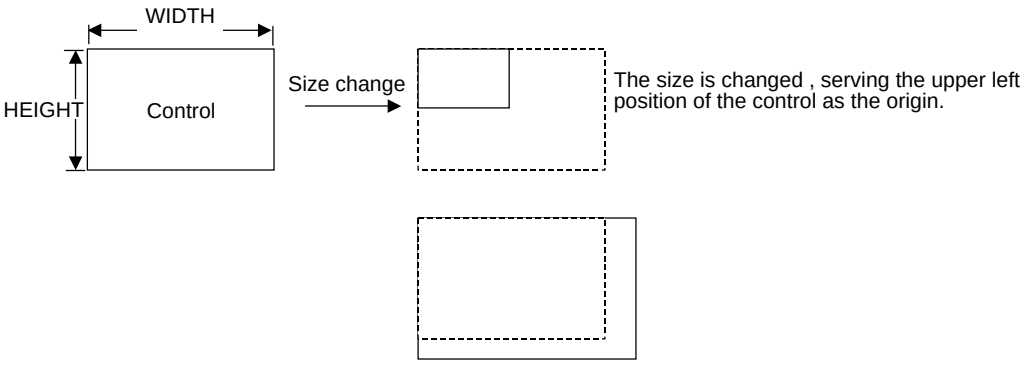
Item	Description
X	Specify the horizontal position from the upper left of the page/view frame of the control (X coordinate) in dots (0 to 2559).
Y	Specify the vertical position from the upper left of the page/view frameof the control (Y coordinate) in dots (0 to 1919).
WIDTH	Specify the width of the control in dots (8 to 2560).
HEIGHT	Specify the height of the control in dots (8 to 1920).

Position



Page/view frame

Size

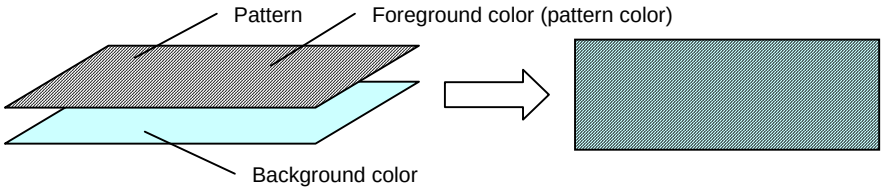


7.1.3 Color/Pattern

Specify the color/pattern of the control

Item	Description
Background color	Specify the background color of the control.
Foreground color	Specify the foreground color (pattern color) of the control.
Pattern	Select the filling pattern among "background," "foreground," "pattern 0" to "pattern 37," and "none."

The relationship among the background color, foreground color and pattern is as shown in the figure below.



NOTE

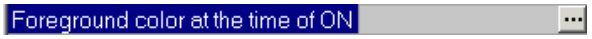
- The color or pattern may be specified for each control state for some controls such as the button for which "foreground color at the time of ON," "background color at the time of OFF" and "pattern at the time of focus" may be specified.
- 38 types of patterns are provided for NC Designer in the default state. For the pattern type, refer to Appendix.

7.1.4 Color Setup

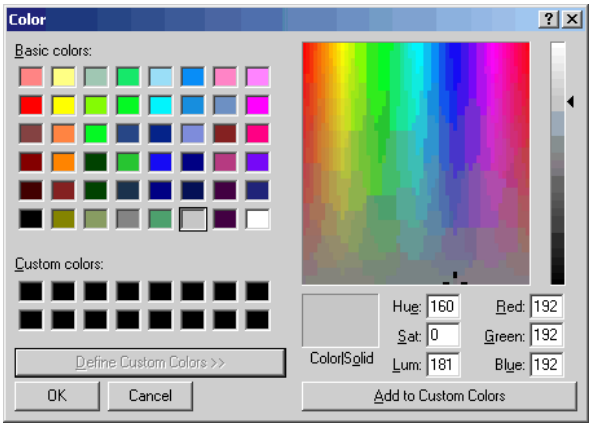
The setting method varies according to the [Color Setup] given in the [Settings] menu.

Direct Color System

1. Click on the setting item related to color in the property window.



2. Click on the [...] button. The [Color] dialog box is displayed.



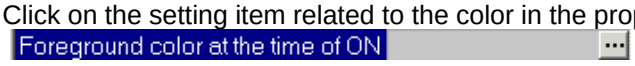
3. Designate the desired color in RGB, and click on the [OK] button.

NOTE

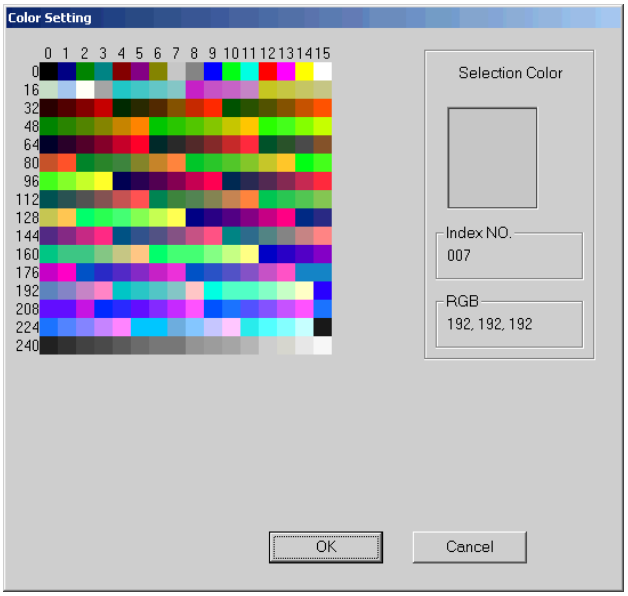
- Colors are reduced to 256 colors on NC Designer. (This is for displaying. The setting (RGB values) does not change.)

Index Color System

1. Click on the setting item related to the color in the property window.



2. Click on the [...] button. The [Color Setting] dialog box is displayed.



3. Select the desired color and click on the [OK] button.

7.1.5 Image

Specify the image displayed on the control.

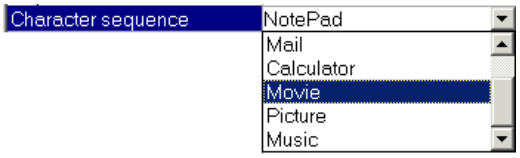
Item	Description
Design	Select the image resource data name to be displayed on the control.

NOTE

- The image resource may be either a BMP or JPG file. For details, refer to Section 6 "Registration of Resources."
- The image may be specified for each control state for some controls such as the button for which "Design at the time of ON," "Design at the time of OFF" and "Design at the time of focus" must be specified.

7.1.6 Character String

Specify the caption character string displayed on the control.

Item	Description
Caption existence	Select whether or not the caption is displayed.
Character sequence	Specify the character string displayed on the control. There are the following two methods for the entry of the character string. ・Select from registered character string resources. Click on the ▼ button at the right of the entry area and select one of registered character string resources.  ・Enter a new character string. Click on the entry area and enter a character string directly. 

NOTE

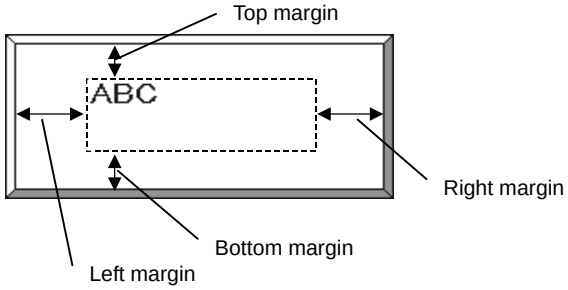
- The new character string is registered as a character string resource.
- The character string is common data for the project. A single user should enter the new character string.
- If [Edit Caption] is selected from the [Edit] menu while a control is selected, you can enter the character string directly on the control.

7.1.7 Character Attribute

Specify the character attribute of the caption.

Item	Description
Character color	Specify the character color.
Horizontal position	Select the horizontal character position among "Align left," "Center" and "Align right."
Vertical position	Select the vertical character position among "Align top," "Center" and "Align bottom."
Font	Select the ID of the font resource for displaying the caption.
Margin left	Designate the starting position of the caption in dots from the left end of the control (0 to 2560).
Margin right	Designate the starting position of the caption in dots from the right end of the control (0 to 2560).
Margin top	Designate the starting position of the caption in dots from the top of the control (0 to 1920).
Margin bottom	Designate the starting position of the caption in dots from the bottom of the control (0 to 1920).

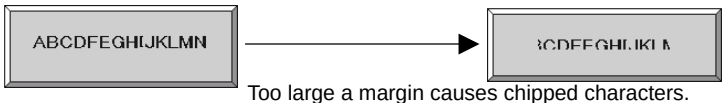
The relationship between the margin and character string position is as shown below.



The solid frame is not included in the margin. The rectangle indicated with dot lines indicates the area where the character string is displayed.

NOTE

◆ Avoid reserving too large a margin. Otherwise the character string may not be displayed completely.



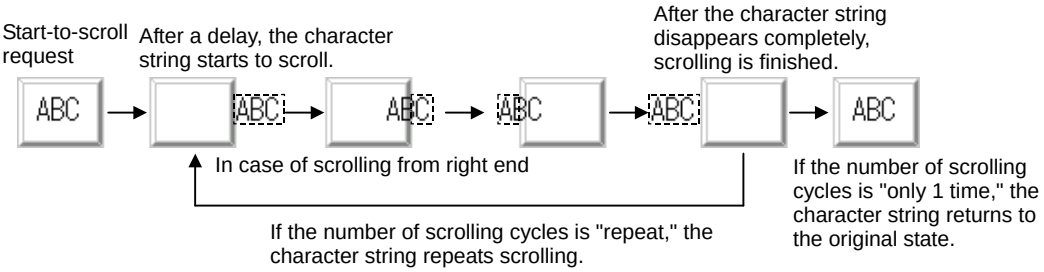
Too large a margin causes chipped characters.

7.1.8 Scrolling Caption Character String

Scroll the caption character string displayed at the control.

Item	Description
Scroll	Select the scroll of the caption character string among "Yes," "No" and "When a string overflows."
The number of times of scrolling	Select the scrolling frequency between "Only 1 time" and "Repeat."
Start delay (ms)	Specify the delay from the start-to-scroll request to the start of scrolling in ms (0 to 60000).
Updating interval (ms)	Specify the scroll refreshment interval in ms (0 to 5000). Specify "0" to refrain from scrolling.
Amount of movements (dot)	Specify the amount of movement in scrolling in dots (0 to 100).
Scroll start position	Select the starting position between "The present position" and "From a right end."

The relationship between the scroll settings and action is as shown below.



NOTE

- The scrolling direction of the caption character string is from right to left (fixed).
- If a carriage return is included in the caption character string, the character string scrolls in a single line.

Before scrolling The character string is in a single line while it scrolls.

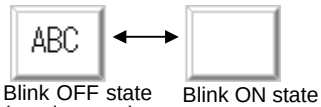
- The caption character string scrolls in the character string display area, allowing for the margins. For the margin, refer to "Character Attribute."

7.1.9 Blink

Blink the caption character string of the control.

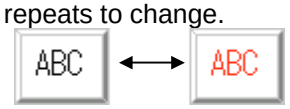
Item	Description
Blink	Select whether the character string blinks or not.
The blink method	Select the caption character string blinking method among "Switch Show/Hide character," "A character color is changed," and "A whole color is changed."
The character color/whole color at the time of blink	Specify the blinking character color. The setting is valid if "A character color is changed" or "A whole color is changed" is selected for "The blink method."

If "Switch Show/Hide character" is selected, the caption character string continues to blink.



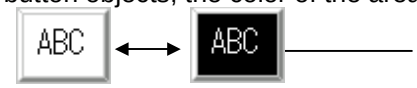
Blink OFF state (regular state) Blink ON state

If "A character color is changed" is selected, the character color of the caption character string repeats to change.



Blink OFF state Blink ON state

If "A whole color is changed" is selected, the button, label, text box and picture objects change the color of the whole control repeatedly, except for the solid frame. With the check box and radio button objects, the color of the area where the caption character string is displayed changes.

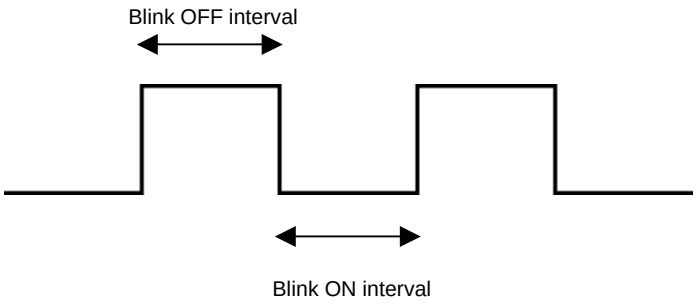


XOR between regular color and "the character color/whole color at the time of blink"

Blink OFF state Blink ON state

Blink Refreshment Interval

The control for which the blink is specified alternates between the blink OFF and blink ON states. Specify the intervals of both states at [panel/window properties] in the [Settings] menu. For the details on the setting method, refer to Section 5.5 "Panel Property Setting" and Section 5.6 "Window Property Setting."



7.1.10 Solid Frame

Specify the solid frame of the control.

Item	Description
Existence of a solid frame	Select the presence of the solid frame between "Yes" and "None."
Solid frame	Select the resource data name of the solid frame resource.

NOTE

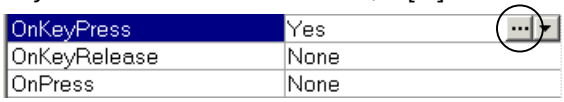
- For the solid frame resource, refer to Section 6 "Registration of Resources."

7.1.11 Callback Function

The callback function is an event-driven function for the user to add the original process in the C++ language. The callback function is generated in source codes after it is specified in the property window of NC Designer.

Item	Description
OnKeyPress	Select "Yes" to add a process to be executed after the key is pressed.
OnKeyRelease	Select "Yes" to add a process to be executed after the key is released.
OnPress	Select "Yes" to add a process to be executed after the mouse or another pointing device is pressed.
OnRelease	Select "Yes" to add a process to be executed after the mouse or another pointing device is released.
OnClick	Select "Yes" to add a process to be executed after the mouse or another pointing device is clicked. If the pointing device is released on the same control, an event occurs, following OnRelease.
OnDraw	Select "Yes" to add a process to be executed after the image is drawn.
OnTimer	Select "Yes" to add a process to be executed after the timer event is called.
OnSetFocus	Select "Yes" to add a process to be executed after the focus is located.
OnKillFocus	Select "Yes" to add a process to be executed after the focus moves apart from the control.
OnCreate	Select "Yes" to add a process to be executed after the page/control is generated.
OnDelete	Select "Yes" to add a process to be executed before the page/control is deleted.
OnUser	Select "Yes" to add an original process of the user.
OnScroll	Select "Yes" to add a process to be executed after the scroll bar is clicked on with the mouse or another pointing device.
OnScrollFinish	Select "Yes" to add a process to be executed after the caption character string has finished scrolling.
OnSelectChange	Select "Yes" to add a process to be executed when the selection line is changed in the list.

If "yes" is selected at each item, a [...] button is displayed in the setting area.

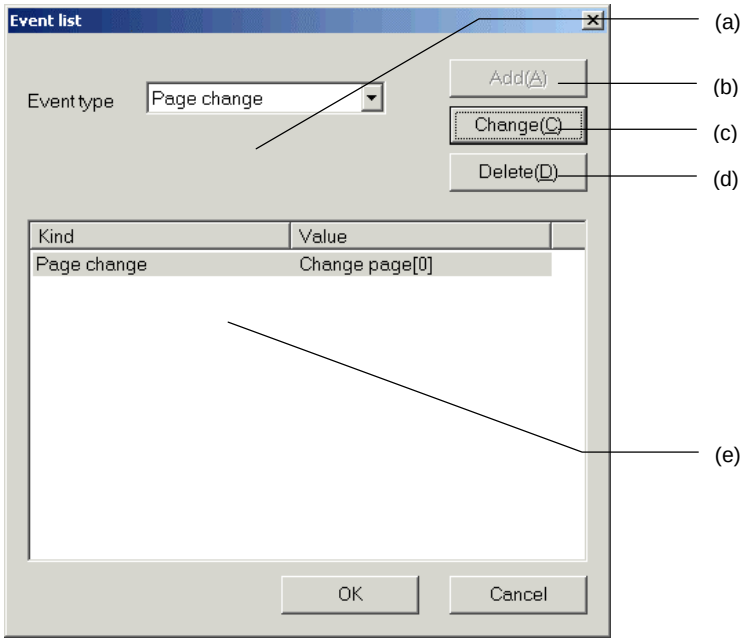


Click on the [...] button to display an [Event list] dialog box where details of the action can be specified.

NOTE

- If the character string does not overflow though "When a string overflows" is selected as a caption character string scrolling method, OnScrollFinish is called immediately without scrolling after the character string starts to scroll. For the caption character string scroll, refer to "Caption Character String Scroll."
- Whether each callback function can be used or not can be changed for each control by the user. The original callback function can be added for each control. For details, refer to Appendix 7 "Adding Original Event for User."

7.1.12 Event List Dialog Box

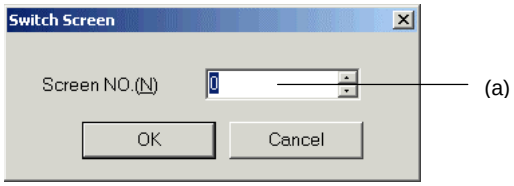


No.	Item	Description
(a)	Event type	Select the event to be added. Only "page change" can be specified.
(b)	Add	Add a selected event. Click on the "Add" button to display the [switch page] dialog box.
(c)	Change	Change the setting of an event selected from the registration list.
(d)	Delete	Delete an event selected from the registration list.
(e)	Entry List	A list of added events is displayed.

NOTE

- The page (screen) that can be switched with "page switching" is panel only. Switching to the window is not supported.

7.1.13 Switch Screen Dialog Box



No.	Item	Description
(a)	Screen NO.	Specify the destination page number. Specify the panel page number as the destination.

NOTE

- After the switch screen setting is given, the switch screen process is added automatically in the callback functions during source code generation.

7.1.14 Show/Hide

Specify whether the control is displayed or hidden.

Item	Description
Show/Hide	Select whether the control is displayed or hidden.

7.1.15 Input Permission

Specify whether entry is permitted or prohibited for the control.

Item	Description
Input permission	Select whether the entry is accepted (permission) or rejected (prohibition).

NOTE

- Select "Hide" for [Show/Hide] and "Permission" for [Input Permission] to create a transparent control object.

7.2 Standard Control

7.2.1 Basic Control Object (GCBasicControl)

The basic control object is a control for providing an area where original drawing for the user or process can be performed.
The basic control does not have a specific function and appearance, action and other properties are described by the user after the source is generated.

To create a basic control object, select [Basic Control] from the [Control] menu of NC Designer or select the following icon.

7.2.1.1 Property Settings

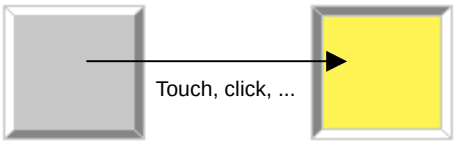
The property settings of the basic control object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Callback function	:	Specify the presence of the callback function.
Show/hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

For properties, refer to "7.1 Common Functions of Controls".

7.2.2 Button Object (GCButton)

The button object holds the ON/OFF status internally and the status changes each time it is pressed or released.
Two types of actions can be selected for the button action: momentary and alternate.



OFF ON

To generate a button object, select [Button] from the [Control] menu of NC Designer or select the icon shown below.

7.2.2.1 Property Settings

The property settings for the button object are divided into followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Button type	:	Select the button action.
Display type	:	Select the display type of the button.
Color/pattern	:	Specify the color and pattern of the control.
Image	:	Specify the image of the control.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Button Type

Item	Description
Button type	Select the button action among the following three types.
Momentary	The button turns on when it is pressed. It turns off when it is released.
Alternate	The button alternates ON and OFF each time it is pressed.
None	The button does not turn on or off even if it is pressed.

Display Type

Item	Description
Display type	Select the button type among the following three types.
Square	Rectangular button. The button is indicated in the designated color and pattern. 
Circle	Round button. The button is indicated in the designated color and pattern. 
Image	The button is indicated with the designated image resource. 

Color/Pattern

Item	Description
Pattern at the time of ON[*1]	Specify the pattern of the ON button.
Foreground color at the time of ON[*1]	Specify the foreground color of the ON button.
Background color at the time of ON[*1]	Specify the background color of the ON button.
Design at the time of ON[*2]	Specify the image of the ON button.
Pattern at the time of OFF[*1]	Specify the pattern of the OFF button.
Foreground color at the time of OFF[*1]	Specify the foreground color of the OFF button.
Background color at the time of OFF[*1]	Specify the background color of the OFF button.
Design at the time of OFF[*2]	Specify the image of the OFF button.

*1: The setting is valid if the [Display Type] is "Square" or "Circle."
*2: The setting is valid if the [Display Type] is "Image."

Image

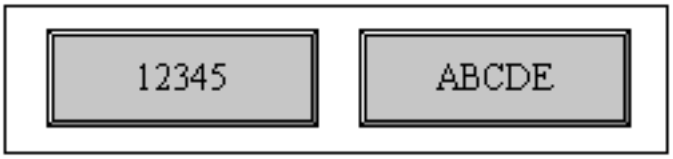
Item	Description
Effect at the time of focus	Specify whether the color of the button when the focus is located changes or not. Select between "change color" and "no change."
Pattern at the time of focus*1	Specify the pattern of the button when the focus is located.
Foreground color at the time of focus*1	Specify the foreground color of the button when the focus is located.
Background color at the time of focus*1	Specify the background color of the button when the focus is located.
Design at the time of focus*2	Specify the image of the button when the focus is located.
Pattern at the time of disable*1	Specify the pattern of the button when the entry is disabled.
Foreground color at the time of disable*1	Specify the foreground color of the button when the entry is disabled.
Background color at the time of disable*1	Specify the background color of the button when the entry is disabled.
Design at the time of disable*2	Specify the image of the button when the entry is disabled.

*1: The setting is valid if the [Display Type] is "Square" or "Circle."
*2: The setting is valid if the [Display Type] is "Image."

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.3 Text Box Object (GCTextBox)

The text box object is a control for the values and character strings to display or enter in the designated rectangle.
For values, character string/binary conversion is made.



To create a text box object, select [Text Box] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.3.1 Property Settings

The property settings for the text box object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color/pattern	:	Specify the color and pattern of the control.
Display type	:	Select the display type.
Password	:	Specify the password.
Caption	:	Specify the caption (character string) to be displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the blink of the caption character string.
Callback function	:	Specify the presence of callback functions.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Color/Pattern

Item	Description
Existence of a background color	Select if the background color is provided or not. If "None" is selected, the background is transparent.
Background color	Specify the background color.
Background color at the time of focus	Specify the background color when the focus is located.
Background color at the time of disable	Specify the background color when the entry is disabled.

Display Type/Display Format

Item	Description
Type	Select the displaying and entry type among the following six types.
Character sequence	A character string is displayed or entered.
signed short	A signed short value is displayed or entered.
unsigned short	An unsigned short value is displayed or entered.
signed long	A signed long value is displayed or entered.
unsigned long	An unsigned long value is displayed or entered.
float	A floating point value is displayed or entered.
Display format (Note)	Specify the value-to-character string conversion type.
Number of the maximum characters	Specify the maximum number of characters to be displayed (1 to 256).
Maximum check	For the value field, select whether to check for the maximum value limit or not.
Maximum	Specify the maximum value for the maximum value check. (-2147483648 to 4294967295).
Minimum check	For the value field, select whether to check for the minimum value limit or not.
Minimum	Specify the minimum value for the minimum value check. (-2147483648 to 4294967295).
Comma	For the value field, select whether to display commas or not.
Half-size number	For entry in the text box, select whether one-byte numbers are allowed or not.
Half-size English small letter	For entry in the text box, select whether one-byte lower case letters are allowed or not.
Half-size English capital letter	For entry in the text box, select whether one-byte upper case letters are allowed or not.
Half-size sign	For entry in the text box, select whether one-byte symbols are allowed or not.
Full size	For entry in the text box, select whether two-byte characters are allowed or not.

(Note)Specify the suitable display format for each type. If not, it will not be displayed correctly.

Password

Item	Description
Password setup	Select "Yes" to display entered characters with asterisks (*). 

Character Attribute

Item	Description
Effect at the time of focus	To display the cursor in the text box where the focus is located, select "With cursor." To display no cursor, select "No effect." To select all characters, select "Selected."

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.4 Label Object (GCLabel)

The label object is a control for displaying a character string inside the designated rectangle.

Screen title created with label

To create a label object, select [Label] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.4.1 Property Settings

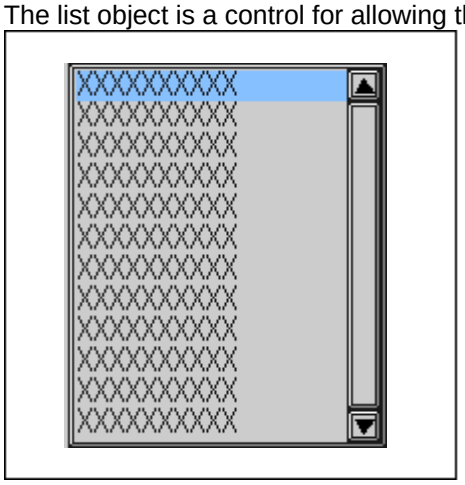
The property settings of the label object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.

For properties, refer to "7.1 Common Functions of Controls".

7.2.5 List Object (GCList)

The list object is a control for allowing the user to select from a list of several character strings.



To generate a list object, select [List] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.5.1 Property Settings

The property settings of the list object are divided into the followings.

Control name	:	Specify the name of the control.
Position/size	:	Specify the position and size of the control.
Color/Pattern	:	Specify the color and pattern of the control.
Scroll bar	:	Specify the color and width of the scroll bar and scroll bar button.
Selection bar	:	Specify the color of the selection bar.
Max. number of lines	:	Specify the maximum number of lines of character strings shown in the list.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Callback functions	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Color/Pattern

Item	Description
Background color	Specify the background color of the control.
Background color at the time of focus	Specify the background color of the list where the focus is located.
Background color at the time of disable	Specify the background color of the disabled list.

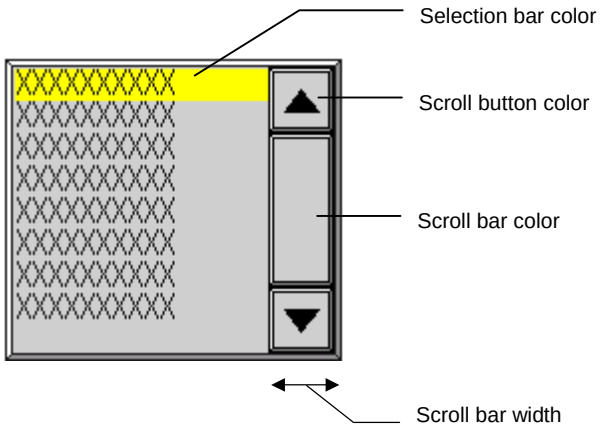
Scroll Bar

Item	Description
Scroll bar color	Specify the color of the scroll bar.
Scroll button color	Specify the color of the scroll button.
Scroll bar width	Specify the width of the scroll bar in dots (16 to 960).

Selection Bar

Item	Description
Select bar color	Specify the color of the selection bar.

The scroll bar and selection bar settings are reflected on the following parts.



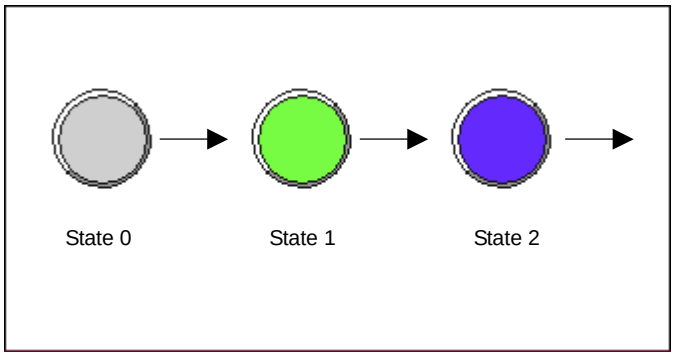
Max. Number of Lines

Item	Description
Number of the maximum lines	Specify the maximum number of lines of character strings displayed in the list (1 to 512).

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.6 Picture Object (GCPicture)

The picture object is a control where the image is switched according to the state of an external device or the internal state of software to notify the user of the state.
The picture supports up to 32 states. Each state has separate appearance.



To create a picture object, select [Picture] from the [Control] menu of NC Designer or select the icon shown below.

7.2.6.1 Property Settings

The property settings of the picture object are divided into the followings.

Control name	:	Specify the name of the control.
Position/size	:	Specify the position and size of the control.
Display type	:	Select the display type of the picture.
Number of states	:	Specify the number of states expressed with the picture.
Color/pattern	:	Specify the color and pattern of the control.
Image	:	Specify the image given to the control.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.

Display Type

Item	Description
Display type	Select the button type among the following three types.
Square	Rectangular button. The button is indicated in the designated color and pattern. 
Circle	Round button. The button is indicated in the designated color and pattern. 
Image	The button is indicated with the designated image resource. 

Number of States

Item	Description
Number of status	Specify the number of states expressed with the picture (1 to 32).

Color/Pattern/Image

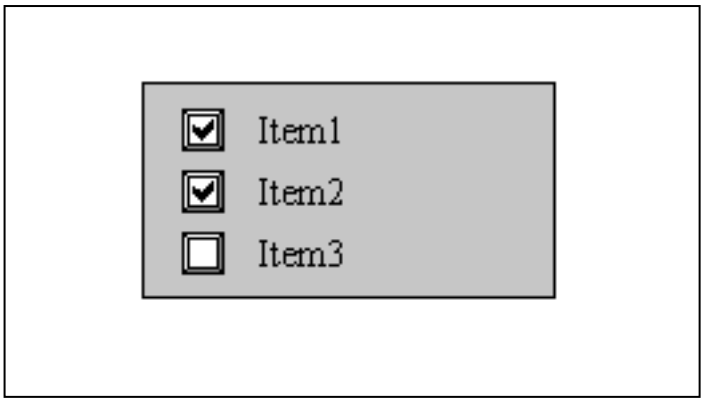
Item	Description
State0 to 31 pattern[*1]	Specify the filling pattern of the picture for each state.
State0 to 31 foreground color[*1]	Specify the foreground color (pattern color) of the picture for each state.
State0 to 31 background color[*1]	Specify the background color of the picture for each state.
State0 to 31 Design[*2]	Select the ID of the image resource displayed for the picture for each state.

*1: The setting is valid if the [Display type] is "Square" or "Circle."
*2: The setting is valid if the [Display type] is "Image."

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.7 Check Box Object (GCCheckBox)

The check box object is a control where the ON/OFF state is held and the ON/OFF state is graphically displayed upon a user-driven state change.
The check box holds the ON/OFF state internally, and the state changes according to events.



To create the check box object, select [Check Box] from the [Control] menu of NC Designer, or select the icon shown below.

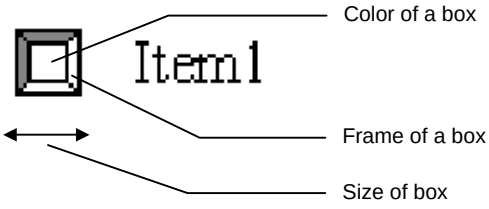
7.2.7.1 Property Settings

The property settings of the check box object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Box	:	Specify the color, size and solid frame of the box.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Focus	:	Specify the displaying method and color of the check box where the focus is located.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Box

Item	Description
Size of box	Specify the box size in dots (8 to 1920). The box is a square.
Existence of a box frame	Specify presence/absence of the box frame.
Frame of a box	Select the ID of the solid frame resource to be given to the box.
Color of a box	Specify the color of the box.



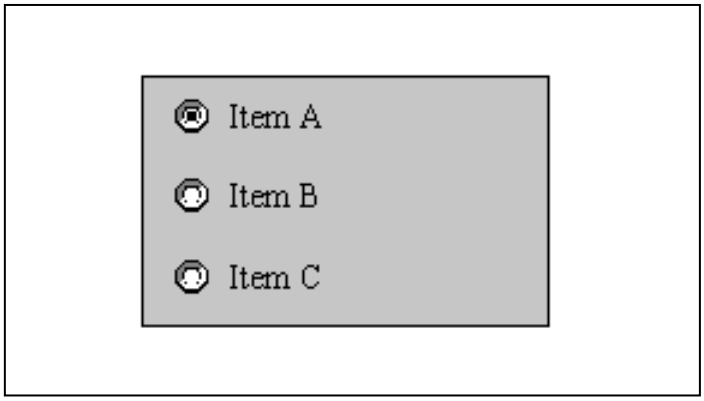
Focus

Item	Description
Effect at the time of focus	Select the displaying method of the check box where the focus is located. (Only "Change color" can be selected with this version.)
Background color at the time of focus	Specify the background color of the check box where the focus is located.
Color of a box at the time of disable	Designate the color of the disabled box.
Character color at the time of disable	Designate the character color of the disabled check box.

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.8 Radio Button Object (GCRadioButton)

The radio button object is a control for realizing exclusive selection among a group of multiple radio buttons.
The radio button holds the ON/OFF state internally, and the state changes according to events.



To create a radio button object, select [Radio Button] from the [Control] menu of NC Designer, or select the icon shown below.

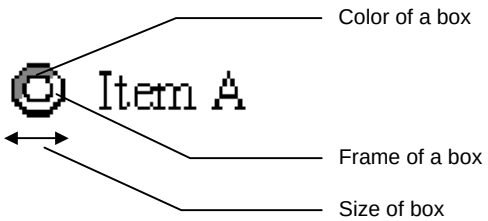
7.2.8.1 Property Settings

The property settings of the radio button object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Box	:	Specify the color, size and solid frame of the box.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of the caption.
Focus	:	Specify the displaying method and color of the radio button where the focus is located.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Specify the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Box

Item	Description
Group NO.	Specify the group number to which the radio button belongs (0 to 32767). Only one radio button among those belonging to the same group number is allowed to be active.
Size of box	Specify the box size in dots (8 to 1920).
Existence of a box frame	Specify the presence of the box frame.
Frame of a box	Select the ID of the solid frame resource to be given to the box.
Color of a box	Specify the color of the box.



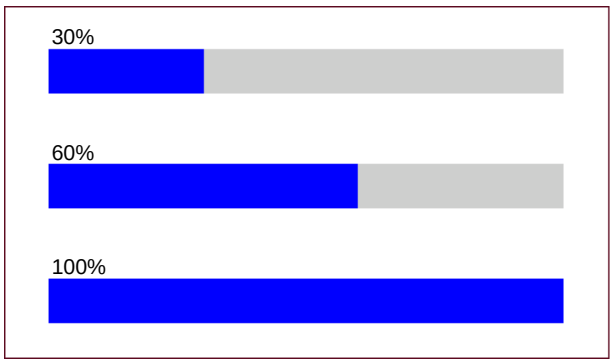
Focus

Item	Description
Effect at the time of focus	Select the displaying method of the radio button where the focus is located (Only "Change color" can be selected with this version.)
Background color at the time of focus	Designate the background color of the radio button where the focus is located.
Color of a box at the time of disable	Designate the color of the disabled box.
Character color at the time of disable	Designate the character color of the disabled radio button.

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.9 Progress Bar Object (GCProgressBar)

The progress bar object is a control expressing the progress of a process with the filled amount.



To create a progress bar object, select [Progress Bar] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.9.1 Property Settings

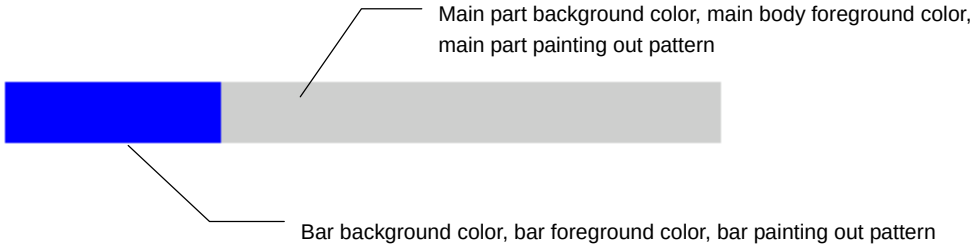
The property settings of the progress bar object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color/pattern	:	Specify the color and pattern.
Filling direction	:	Specify the filling direction of the progress bar.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.

Color/Pattern

Item	Description
Main part background color	Specify the background color of the main body of the progress bar.
Main part foreground color	Specify the foreground color of the main body of the progress bar.
Main part painting out pattern	Specify the filling pattern of the main body of the progress bar.
Bar background color	Specify the background color of the bar.
Bar foreground color	Specify the foreground color of the bar.
Bar painting out pattern	Specify the filling pattern of the bar.

The color settings are reflected in the following way.



Filling Direction

Item	Description
Direction	Select the direction of progress of the bar among the following options: "From left to right," "From right to left," "From top to bottom" and "From bottom to top."
Minimum	Designate the 0% bar length (-2147483648 to 2147483647).
Maximum	Designate the 100% bar length (-2147483648 to 2147483647).

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.10 HTML Browser Object (GCHtmlBrowser)

The HTML browser object is a control displayed on the screen upon interpretation of the data of an HTML file.

To create an HTML browser object, select [HTML Browser] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.10.1 Property Settings

The HTML browser property settings are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Default	:	Specify the default character color, default background color, and default character attribute applied when they are not specified in the HTML file.
Solid frame	:	Specify the solid frame of the control.
HTML file	:	Specify the HTML displayed at the control.
Scroll bar	:	Specify the color and width of the scroll bar and the color and image of the scroll bar button.
Show/Hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Default

Item	Description
Default character color	Specify the default character color used when it is not specified in the HTML file.
Default background color	Specify the default background color used when it is not specified in the HTML file.
Default link color	Specify the default link color used when it is not specified in the HTML file.
Default font	Specify the default font used when it is not specified in the HTML file.

HTML File

Item	Description
HTML Folder	Select the folder containing an HTML file
HTML File	Select the ID of the HTML file resource as an HTML file displayed first as a control.

NOTE

◆ To store HTML files in HTML folders, absolute paths have to be defined in the Config.ini file. The relationship between the description in the combo boxes and actual folders is shown below.

<Example of M700/M700VW(WindowsXPe)>
[HTML_BROWSER]
HTMLDATA0=D:/Custom/HTMLDATA0/
HTMLDATA1=D:/Custom/HTMLDATA1/
HTMLDATA2=D:/Custom/HTMLDATA2/
:
HTMLDATA7=D:/Custom/HTMLDATA7/

<Example of M700(WindowsCE)>
[HTML_BROWSER]
HTMLDATA0=/Memory Card/Custom/HTMLDATA0/
HTMLDATA1=/Memory Card/Custom/HTMLDATA1/
HTMLDATA2=/Memory Card/Custom/HTMLDATA2/
:
HTMLDATA7=/Memory Card/Custom/HTMLDATA7/

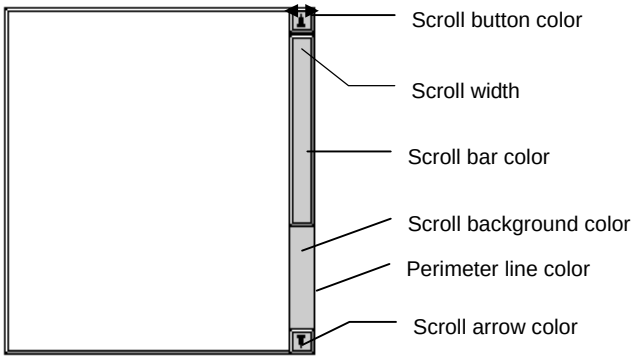
<Example of M70/M70V/M700VS/E70>
[HTML_BROWSER]
HTMLDATA0=/custom/HTML/
HTMLDATA1=
HTMLDATA2=
:
HTMLDATA7=

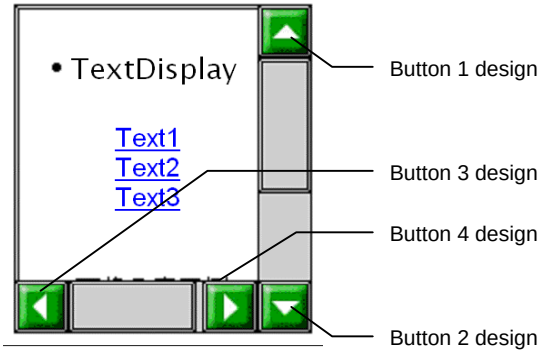
Scroll Bar

Item	Description
Display scroll bar	Select between "Painting out" and "Image."
Scroll bar color*1	Specify the color of the scroll bar.
Scroll button color*1	Specify the color of the scroll button.
Scroll arrow color*1	Specify the color of the arrow of the scroll bar.
Button1 to 4 design at the time of ON*2	Specify the image displayed at the ON scroll bar.
Button1 to 4 design at the time of OFF*2	Specify the image displayed at the OFF scroll bar.
Existence of a perimeter line	Select the perimeter line between "Yes" and "None."
Perimeter line color	Specify the color of the perimeter line.
Scroll background color	Specify the background color of the scroll bar.
Scroll width	Specify the width of the scroll bar in dots (16 to 96).

*1: The setting is valid if "display scroll bar" is "painting out."
*2: The setting is valid if "display scroll bar" is "image."

The scroll bar settings are reflected on the following parts.





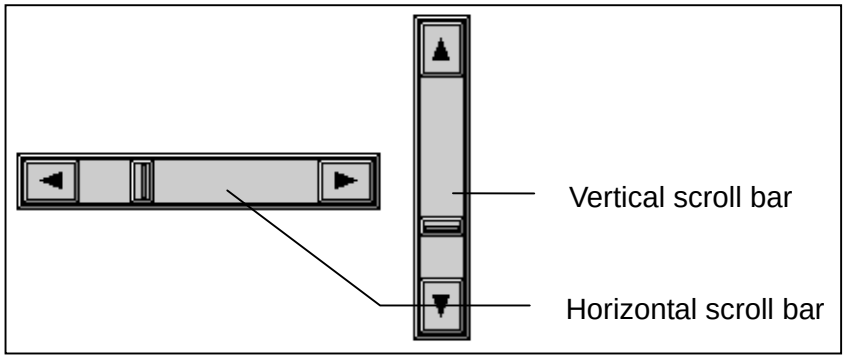
NOTE

- If the displaying area of the HTML file is larger than that of the HTML browser control, the vertical and/or horizontal scroll bar(s) is (are) displayed. If the displaying area of the HTML file is smaller than that of the HTML browser control, the remaining screen area is filled with the background color and no scroll bar is displayed.

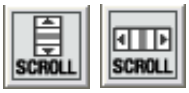
For the other properties, refer to "7.1 Common Functions of Controls".

7.2.11 Scroll Bar Object (GCScrollBarEx)

The scroll bar object is a sliding button used to scroll the screen image up/down or left/right. The scroll bar includes two types: vertical scroll bar for vertical movement and horizontal scroll bar for horizontal movement.



To create a scroll bar object, select [Vertical Scroll Bar] or [Horizontal Scroll Bar] from the [Control] menu of NC Designer, or select the icon shown below.



7.2.11.1 Property Settings

The Property settings of the scroll bar are divided into the followings. The same settings are used for both the vertical and horizontal scroll bars.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Display type	:	Select the displaying type of the button.
Color/image	:	Specify the color and width of the scroll bar and the color and image of the scroll button.
Scroll movement	:	Specify the minimum and maximum values of the scroll bar.
Knob	:	Specify the width and color of the knob.
Solid frame	:	Specify the solid frame of the control.
Perimeter line	:	Specify the perimeter line of the control.
Perimeter solid frame	:	Specify the perimeter solid frame of the control.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control or bar is displayed or hidden.
Input permission	:	Select whether entry is accepted (permission) or rejected (prohibition).

Display Type

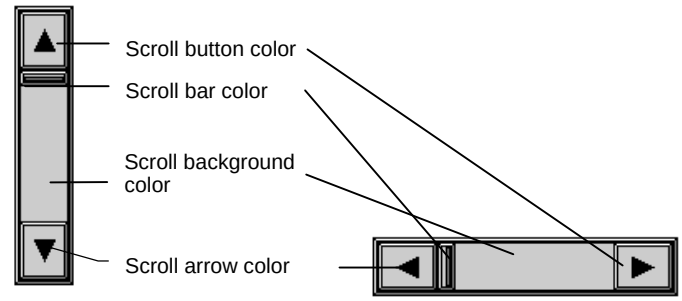
Item	Description
Display type	Select the button type from the following two options.
Painting out	The button is expressed in the designated scroll button and scroll arrow colors.
Image	The button is expressed with the designated image resource.

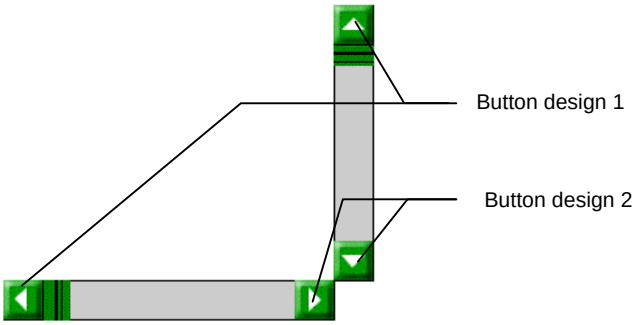
Color/Image

Item	Description
Scroll bar color*1	Specify the color of the scroll bar.
Scroll button color*1	Specify the color of the scroll button.
Scroll arrow color*1	Specify the color of the scroll button arrow.
Button 1 to 2 design at the time of ON*2	Specify the image of the ON button.
Button 1 to 2 design at the time of OFF*2	Specify the image of the OFF button.
Scroll background color	Specify the background color of the scroll bar.

*1: The setting is valid if the [Display type] is "Painting out."
*2: The setting is valid if the [Display type] is "Image."

The scroll bar settings are reflected on the following parts.





Scroll Movement

Item	Description
Scroll minimum	Specify the minimum value of the movement range of the scroll bar (0 to 32767).
Scroll maximum	Specify the maximum value of the movement range of the scroll bar (0 to 32767).
1 page size	Specify the amount scrolled upon a click of the background color area of the scroll bar (1 to 32767).

Knob

Item	Description
Knob width	Specify the width of the knob (0 to 2560).
Knob color	Specify the color of the knob.

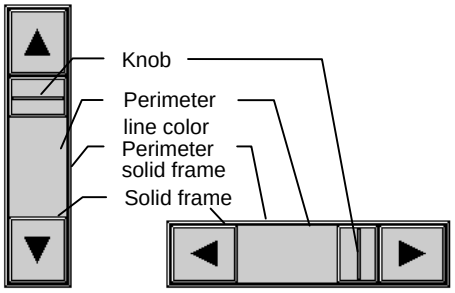
Perimeter Line

Item	Description
Existence of a perimeter line	Select the presence of the perimeter line of the scroll between "Yes" and "None."
Perimeter line color	Specify the color of the perimeter line.

Perimeter Solid Frame

Item	Description
Existence of a perimeter solid frame	Select the perimeter solid frame of the whole scroll bar between "Yes" and "None."
Perimeter solid frame	Select the ID of the solid frame resource.

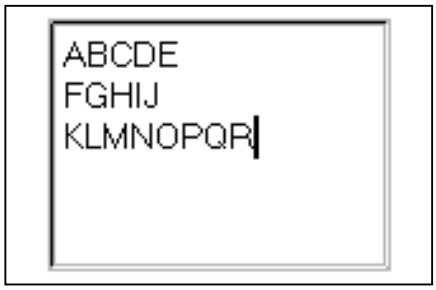
The knob, solid frame, perimeter line, and perimeter solid frame settings are reflected on the following parts.



For the other properties, refer to "7.1 Common Functions of Controls".

7.2.12 Edit Control Object (GCEdit)

The edit control object is a control for displaying, inserting or overwriting a character string in the designated rectangle. The cursor is displayed and carriage return can be entered.



To create an edit control object, select [Edit Control] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.12.1 Property Settings

The property settings of the edit control object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color/pattern	:	Specify the color and pattern of the control.
Solid frame	:	Specify the solid frame of the control.
Insert/overwrite	:	Select between the insertion and overwriting of the entered characters.
Buffer size	:	Specify the internal buffer size and line buffer size.
Scroll bar	:	Specify the color and width of the scroll bar and the color and image of the scroll bar button.
Character attribute	:	Specify the character attribute of the displayed character string.
Callback function	:	Specify whether or not the callback functions are provided.
Show/hide	:	Select whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).

Insert/Overwrite

Item	Description
Insert/Overwrite	Select the character entry mode between "Insert" and "Overwrite."

Buffer Size

Item	Description
Internal buffer size(KB)	Specify the total buffer size of the displayed character strings in kilo bytes (1 to 5123).
Single line buffer size(B)	Specify the line buffer size in bytes (2 to 2048).

NOTE

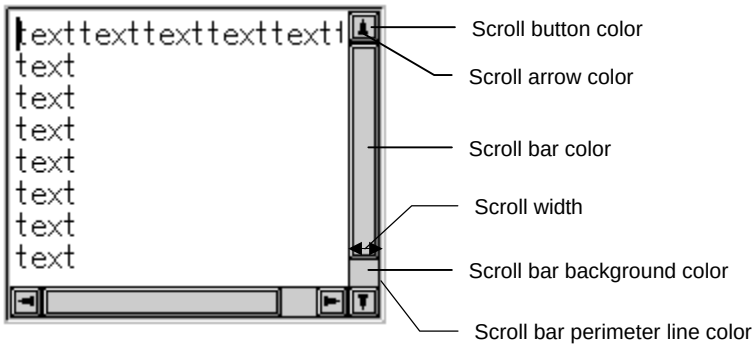
- The calculation method of the buffer size is shown below.
  - (Line buffer size) = (1 character (2B)) x (number of characters in line)
  - (Internal buffer size) = ((line buffer size) x (number of lines)) / 1000
- If characters exceeding the buffer size are entered, characters are stored up to the limit and overflowing characters are abandoned.

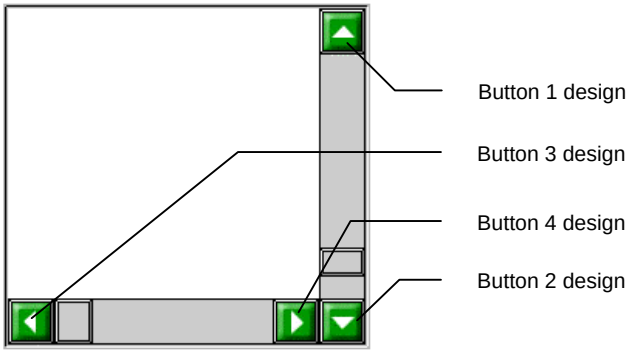
Scroll Bar

Item	Description
Display scroll bar	Select the scroll bar between "Painting out" and "Image."
Scroll bar color*1	Specify the color of the scroll bar.
Scroll button color*1	Specify the color of the scroll button.
Scroll arrow color*1	Specify the color of the arrow of the scroll button.
Existence of a scroll bar solid frame	Select the solid frame of the scroll bar between "Yes" and "None."
Scroll bar solid frame	Select the ID of the solid frame resource of the scroll bar.
Button1 to 4 design at the time of ON*2	Specify the image of the ON scroll bar.
Button1 to 4 design at the time of OFF*2	Specify the image of the OFF scroll bar.
Existence of a scroll bar perimeter line	Select the perimeter of the scroll bar between "Yes" and "None."
Scroll bar perimeter line color	Specify the color of the perimeter line of the scroll bar.
Scroll bar background color	Specify the background color of the scroll bar.
Scroll width	Specify the width of the scroll bar in dots (16 to 96).

*1: The setting is valid if [Display scroll bar] is "Painting out."
*2: The setting is valid if [Display scroll bar] is "Image."

The scroll bar settings are reflected on the following parts.





For the other properties, refer to "7.1 Common Functions of Controls".

7.2.13 Table Object (GNCTable)

Table control (GNCTable) is a control that uses cells with the number (n) of rows and columns to manage and display the character string data. Each row and column can have its own title.

To create a table control object, select [TABLE] from the [Control] menu or select the following icon in the NC Designer.

7.2.13.1 Property Settings

The property settings of the table control object are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entries are accepted (permission) or rejected (prohibition).
Table	:	Specify the number of rows/columns and the column width in the control.
Solid frame	:	Specify the solid frame of the control.
Character attribute	:	Specify the character attribute of captions.
Sub cursor	:	Specify whether to show/hide the Sub cursor, as well as its color attribute and default display position.
Row title	:	Specify whether to show/hide the row title, as well as its displayed character, color attribute and the space between the rows.
Column title	:	Specify whether to show/hide the column title, as well as its displayed character, color attribute and the space between the columns.
Data area	:	Specify the solid frame, color attribute and the space between the cells.
Callback function	:	Specify whether or not the callback functions are provided.

Table

Item	Description
Number of the rows	Specify the number of rows in the data area. (1 to 20).
Number of the columns	Specify the number of columns in the data area. (1 to 32)
Kind of the columns ratio	Select whether to use percentage or pixel values to specify the width of columns.
Columns ratio	Specify the width of columns according to the type selected in "Kind of the columns ratio". Use "\t" to delimit each value of width.

Sub Cursor

Item	Description
Existence of the sub cursor	Select the existence of a sub cursor between "Yes" and "None".
Initial row position of the Sub cursor	Specify the Sub cursor's initial position in rows.
Initial column position of the Sub cursor	Specify the Sub cursor's initial position in columns.
Foreground color of the Sub cursor	Select the character color of the Sub cursor.
Background color of the Sub cursor	Select the background color of the Sub cursor.

Row Title

Item	Description
Show/Hide of the row title	Select whether the row titles are displayed or hidden.
Character sequence of the row title	Input character strings for each row title.Use "\t" to delimit each character string.
Row spacing of the row title	Specify the space between the row title cells in pixels.
Column spacing of the row title	Specify the space between the row title and the data area in pixels.
Pattern of the row title	Select the fill pattern of the row title.
Foreground color of the row title	Select the character color of the row title.
Background color of the row title	Select the background color of the row title.
Boundary color of the row title	Select the boundary color of the row title cells.
Horizontal character position of the row title	Select "Align left"/"Center"/"Align right" for the horizontal character position in the row title.
Vertical character position of the row title	Select "Align top"/"Center"/"Align bottom" for the vertical character position in the row title.
Existence of the row title solid frame	Select the existence of the solid frame between "Yes" and "None".
Row title solid frame	Select the ID of the solid frame resource.

Column Title

Item	Description
Show/Hide of the column title	Select whether the column titles are displayed or hidden.
Character sequence of the column title	Input character strings for each column title. Use "\t" to delimit each character string.
Row spacing of the column title	Specify the space between the column title cells in pixels.
Column spacing of the column title	Specify the space between the column title and the data area in pixels.
Pattern of the column title	Select the fill pattern of the row title.
Foreground color of the column title	Select the character color of the column title.
Background color of the column title	Select the background color of the column title.
Boundary color of the column title	Select the boundary color of the column title cells.
Horizontal character position of column title	Select "Align left"/"Center"/"Align right" for the horizontal display position of characters in the column title.
Vertical character position of the column title	Select "Align top"/"Center"/"Align bottom" for the vertical display position of characters in the column title.
Existence of the column title solid frame	Select the existence of the solid frame between "Yes" and "None".
Column title solid frame	Select the ID of the solid frame resource.

Data Area

Item	Description
Existence of the whole data area solid frame	Select the existence of the solid frame for the whole data area between "Yes" and "None".
Whole data area solid frame	Select the resource ID of the solid frame for the whole area.
Row spacing of the data area	Specify the space between the rows in the data area in pixels.
Column spacing of the data area	Specify the space between the columns in the data area in pixels.
Pattern of the data area	Select the fill pattern of the data area.
Foreground color of the data area	Select the character color of the data area.
Background color of the data area	Select the background color of the data area.
Boundary color of the data area	Select the boundary color of the data area.
Horizontal character position of the data area	Select "Align left"/"Center"/"Align right" for the horizontal character position in the data area.
Vertical character position of the data area	Select "Align top"/"Center"/"Align bottom" for the vertical character position in the data area.
Existence of the data area solid frame	Select the existence of the solid frame for each cell between "Yes" and "None".
Data area solid frame	Select the resource ID of the solid frame for each cell.

Callback Function

Item	Description
OnChangeString	Select "Yes" to add a process to be executed after the contents of data area has changed.
OnSubCursorMove	Select "Yes" to add a process to be executed after the sub-cursor position has changed.

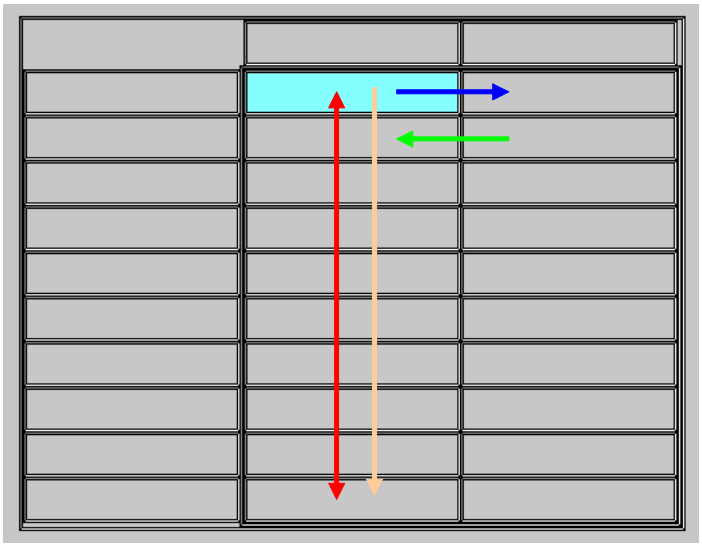
For the other properties, refer to "7.1 Common Functions of Controls".

7.2.13.2 Compliments

Movement of the Sub Cursor

The following table shows how the Sub cursor moves in the table control object.

↑ key	↓ key	Tab key	Shift+Tab key	Enter key
Sub cursor moves up in the same column.	Sub cursor moves down in the same column.	Sub cursor moves to the right.	Sub cursor moves to the left.	Sub cursor moves down in the same column.



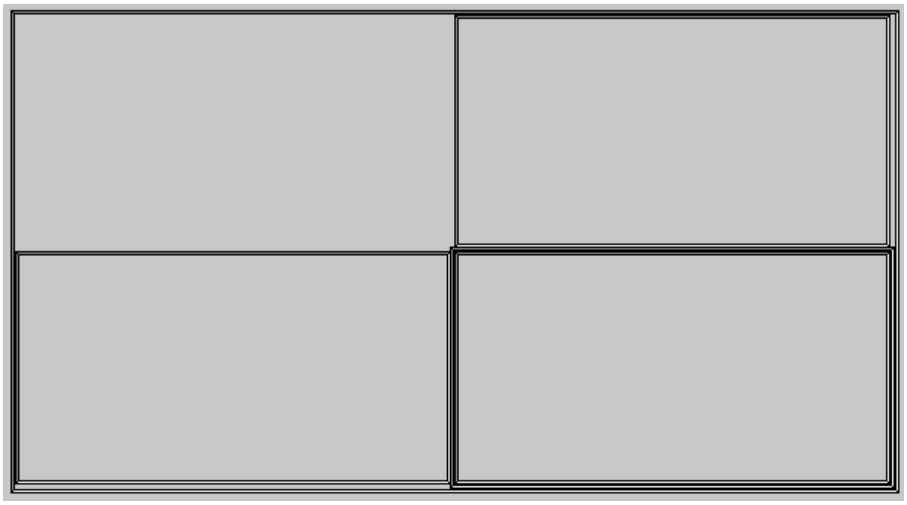
Usage Examples

The followings show the initial display of the control object and the screen images when the properties were changed.

[Screen1 (initial display)]

Property settings for initial display

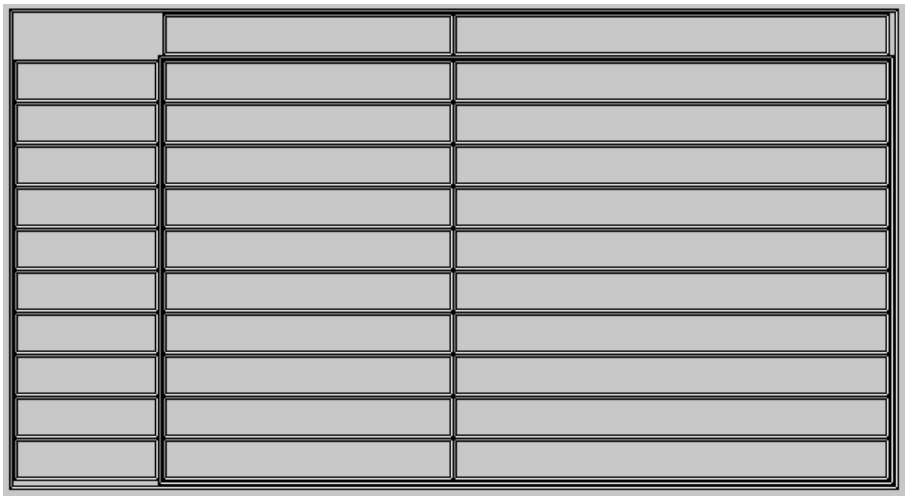
Number of rows	1
Number of columns	1
Columns ratio	Blank



[Screen 2]

Properties to change

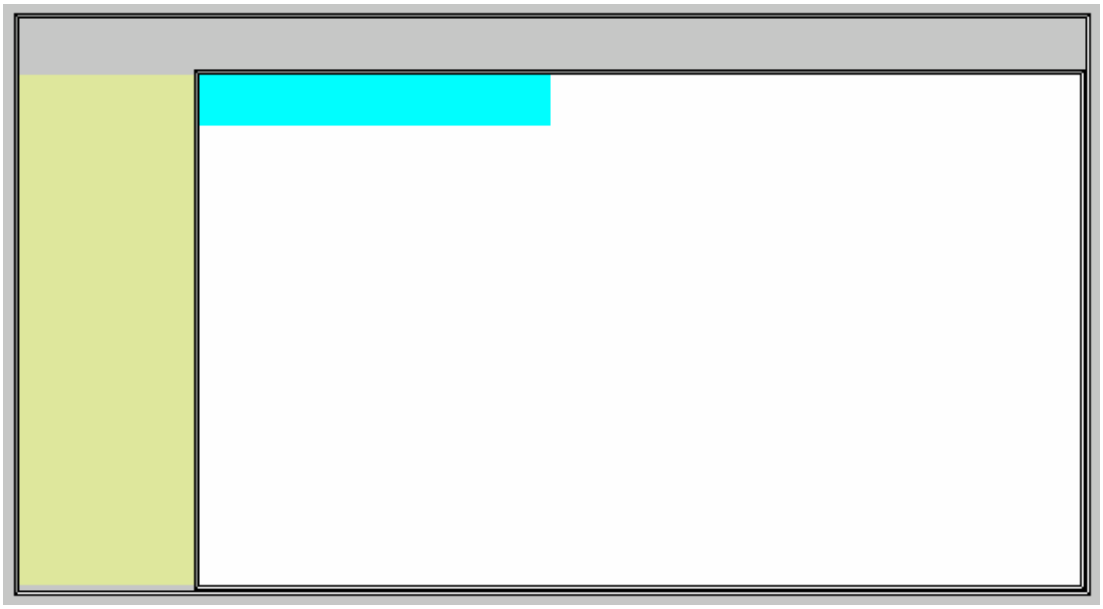
Number of rows	10
Number of columns	2
Columns ratio	1\t2\t3



[Screen 3]

Properties to change

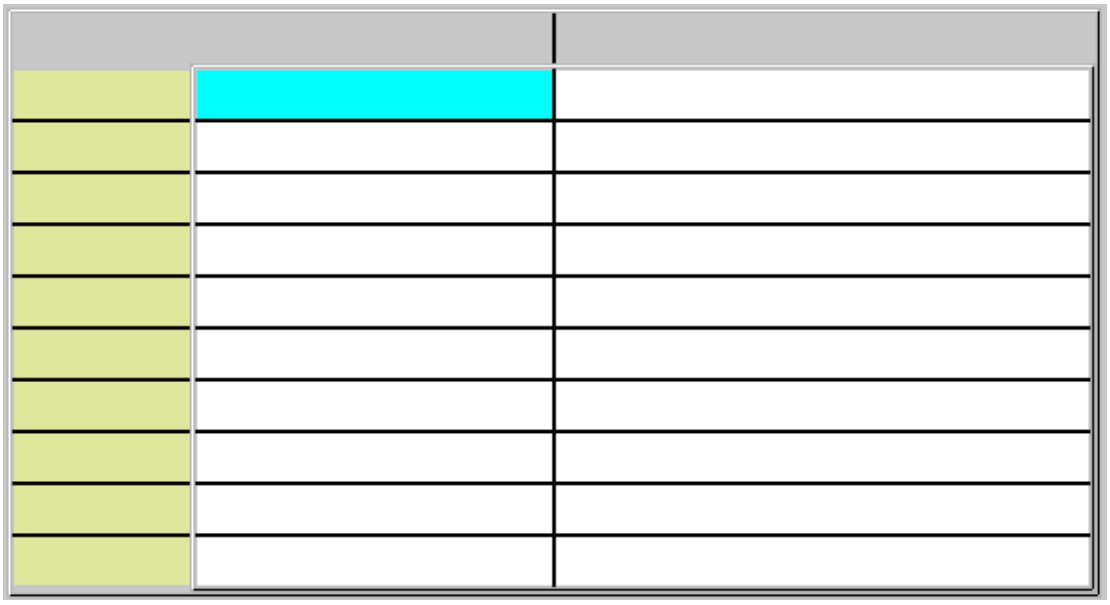
Existence of the row title solid frame	None
Existence of the column title solid frame	None
Existence of the data area solid frame	None
Background color of the row title	RGB(220,230,155)
Background color of the data area	RGB(255,255,255)
Sub Cursor	RGB(0,255,255)



[Screen 4]

Properties to change

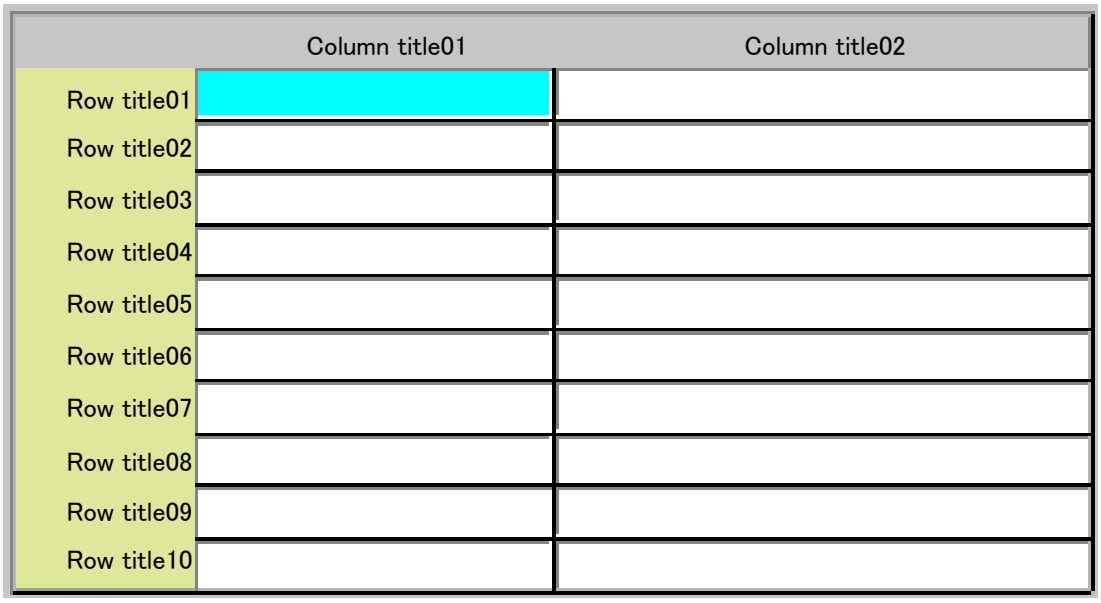
Perimeter solid frame	ID_BORDER00003
Whole data area solid frame	ID_BORDER00003
Row spacing of the data area	1
Column spacing of the data area	1



[Screen 5]

Properties to change

Character sequence of the row title	"Row title01\t Row title02\t Row title03\t Row title04\t Row title05\t Row title06\t Row title07\t Row title08\t Row title09\t Row title10"
Boundary color of the row title	RGB(220,230,155)
Horizontal character position of the row title	Align Right
Character sequence of the column title	"Column title 01\t Column title 02"
Boundary color of the column title	RGB(192,192,192)
Horizontal color of the column title	Center



Remarks

Restrictions

Restrictions for creating a control object are shown below.

(1) Properties for pattern and foreground color

The pattern and the foreground are not displayed as specified in the properties of column title, row title and data area, although the specification is possible.

(2) Properties of the control on the on-memory panel or window

The setting values in the properties of the control on the on-memory panel or window, after having been changed with public functions, are retained when redisplayed with a screen change.

(3) Availability of GCSTableGetCellNumFromPoint( ) when the control object has never been displayed

If the table control object, allocated on the on-memory panel or window, has never been displayed, an attempt to get a cell No. with GCSTableGetCellNumFromPoint( ) leads an error (GERR_NCTABLE_RANGEOVER).

(4) Setting range of a character string in GCSTableSetCellString( )

GCSTableSetCellString( ) can contain only 128 characters to specify a character string for a cell in the data area.

(5) Data after the number of rows and columns were changed with public functions

If the smaller number of rows than displayed is specified in GCSNCTableSetLineCount( ) to change the number of rows, the data (displayed character string, foreground color and background color) in the rows to be hidden will be cleared.

If a control object with 20 rows, for example, is changed to be displayed with 17 rows with GCSNCTableSetLineCount( ), the data that has been set in 18th to 20th rows is all lost after the change. The data will not be restored if the number of rows is set to 20 again with GCSNCTableSetLineCount( ).

The same happens when the number of columns is changed with GCSNCTableSetRowCount( )

(6) Setting range of the font resource ID with public functions

When changing the font with GCSNCTableSetFontID( ), specify a font resource ID for the 2nd argument "usID" from the ones registered as NC Designer resource in designing.

If the unregistered font resource ID is specified, Windows2000/XP does not change the font.(Then an error code "GERR_NCTABLE_RANGEOVER" appears.)

WindowsCE applies the font randomly selected by the system, which leads to a display failure such as unreadable characters. (Then an error code "GERR_NOERR" appears.)

Timing of the Display with the Changed Font and the Number of Rows/Columns

The following table shows the time to display the updates for the control object allocated on the panel or window, when the configurations (font, number of rows/columns) are changed with public functions.

Function name	When the control object is displayed	When the control object is hidden (on-memory)
GCSNCTableSetFontID	○	Δ
GCSNCTableSetLineCount	○	Δ
GCSNCTableSetRowCount	○	Δ

○: Updates instantly
Δ: Updates when the control object is displayed again
×: Execution is not available

7.2.14 Input Box Object(GInputBox)

The input box object is a control that displays numerical values and character strings as well as entered keys. It has the same functions as the text box object but differs in the following points.
- Operation function was added.
- The input values can be reflected to another specified control (Specification of a control to reflect the INPUT).

For the specification of a control to reflect the INPUT, refer to GCSInputBoxSetRefrectControl in "16.5.26 InputBox".
In combination with the sub cursor setting, the input value can be set in the control at which the sub cursor is pointed and the control can be moved by the arrow key/TAB key.
For the details of sub cursor settings, refer to "9.7 Sub Cursor Setting".

To create an input box object, select [Input Box] from the [Control] menu of NC Designer, or select the icon shown below.

7.2.14.1 Property Settings

The property settings for the input box object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Show/hide	:	Select whether the control is displayed or hidden.
Color/pattern	:	Specify the color and pattern of the control.
Display type/ Display format	:	Specify the format of character strings to be displayed in the control.
Password	:	Specify the password.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Operation function	:	Specify whether or not the operation function is provided.
Echo back	:	Specify whether or not the echo back is provided.
Input method	:	Specify the input method (Absolute/Incremental)
Sub cursor	:	Specify the display color of the sub cursor.
Callback function	:	Specify the presence of callback functions.

Color/pattern

Item	Description
Existence of a background color	Select whether to provide the background color. If "None" is selected, the background will be transparent.
Background color	Specify the background color.
Character color	Specify the character color.
Background color at the time of focus	Specify the background color of an input box where the focus is located.
Character color at the time of focus	Specify the character color of an input box when the focus is located.
Background color at the time of disable	Specify the background color of an input box when the entry is disabled. When the entry is disabled, entered contents will be cleared.

Display type/Display format

Item	Description
Number of the maximum characters	Specify the maximum number of characters to display. (1 to 256)

Password

Item	Description
Password setup	Select "Yes" to display entered characters with asterisks (*).

Character attribute

Item	Description
Effect at the time of focus	To display the cursor in the input box where the focus is located, select "With cursor." Not to display a cursor, select "No effect." To select all characters, select "Selected."
Cursor type	Select the type of cursor from "Vertical line" or "Block", when "Effect at the time of focus" is set as "With cursor" or "Selected".
Cursor color	Specify the color of cursor.

Operation function

Item	Description
Operation function	Select "Yes" or "None" to provide the operation function.

Echo back

Item	Description
Echo back	Select whether to echo back the positional value of the control to which INPUT will be reflected or the sub cursor, from "Yes"/"No". * The control to which INPUT will be reflected can be designated by the function "GCSInputBoxSetRefrectControl".

Input method

Item	Description
Abs/Inc	Specify whether to directly set the input value to the control when entering the INPUT key (absolute), or to add the positional value of the sub cursor or the control to which INPUT will be reflected, to the input value when entering the INPUT key (incremental).

Sub cursor

Item	Description
Sub cursor Background color	Specify the background color of the control where the sub cursor is displayed.
Sub cursor Character color	Specify the character color of the control where the sub cursor is displayed.

Callback function

Item	Description
OnSubCursorMove	Execute this function after the sub cursor position changed.
OnError	Execute this function if setting the contents to the control to which the INPUT will be reflected failed.
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

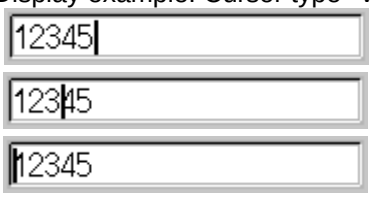
For the other properties, refer to "7.1 Common Functions of Controls".

7.2.14.2 Compliments

Cursor type

The cursor type can be changed between "Vertical line" and "Block" as shown below.

Display example: Cursor type "Vertical line"



Display example: Cursor type "Block"

Operations with Keys

Types of Keys

The following keys are available for the input box.

Key type	Key	Operation
Data setting keys	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 + - / * = . , ; ( ) etc.	Press these keys to set alphabetic characters, numerals and operation symbols, etc.
Data correction keys	INSERT(Data insert key)	When cursor type is "block", the data insertion mode is entered. Press a data setting key to insert a character before the current cursor position. (The overwrite mode is entered when the DELETE, C.B, INPUT, cursor or TAB, etc., key is pressed, or when the screen is changed.)
	DELETE(Data delete key)	Press DELETE to delete a character in the data setting area. - When the cursor is "Vertical line", the character after the cursor position will be deleted. - When the cursor is "Block", the character before the cursor position will be deleted.
	C.B(Cancel key)	Press C.B to cancel the setting in the data setting area.
Cursor keys	↑ ↓	When the sub cursor settings are made, press these keys to move the target control.
	|← →|	When the sub cursor settings are made, press these keys to move the target control.
	← →	Press these keys to move the cursor one character to the left or right in the data setting area. If a control is set as the destination of these arrow keys (←, →) by the sub cursor settings, the cursor will not move within the data setting area and, instead, will move to the target control.
INPUT key	INPUT	Press INPUT to fix the data in the data setting area and reflect the input value to the control which is specified to reflect the INPUT. When the sub cursor settings are made, the input value will be reflected before the sub cursor moves.
Operation key	SP(Space key)	Inserts a blank.

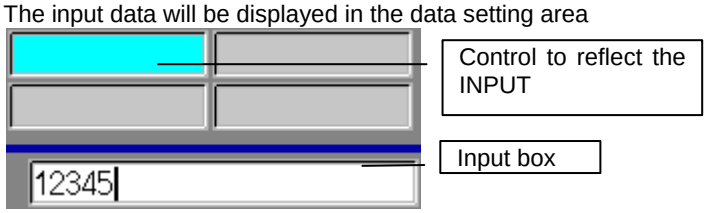
(Note) If any key other than those listed above is entered, it will be ignored. If [Key transfer control] is set by the sub cursor setting, the focus will move to the target control and transfers the key.
For the sub cursor setting, refer to "9.7 Sub Cursor Setting".

Set the input numerals and alphabetic characters

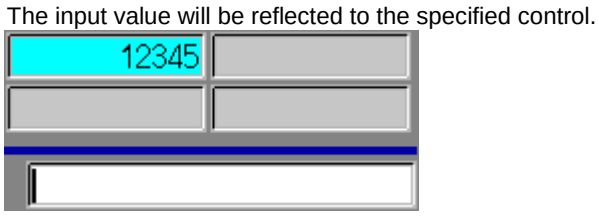
Input numeric, alphabetical and other keys while the focus is placed on the control to display the character strings.
By specifying a control to reflect the INPUT beforehand, the input data can be reflected to the specified control when pressing the INPUT key.

Operation example:

(1) Input data keys 1 2 3 4 5



(2) Press INPUT key.



(Note) When reflecting the input data to the specified control is conducted successfully, the data in the data setting area will be cleared. And if it fails, the data will remain displayed in the data setting area.

Operations in the data setting area

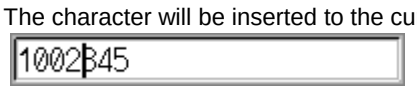
Keys will be input to where the cursor is displayed.
When a key is input, the data will be displayed at the current cursor position and the cursor will shift a character to the right.

Numeric keys/Alphabetical keys

Operation example: Cursor type "Vertical line"

(1) The cursor is placed as shown in the right.

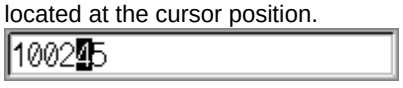
(2) Press 2 key.



Operation example: Cursor type "Block"

(1) The cursor is placed as shown in the right.

(2) Press 2 key.

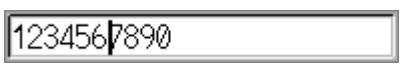


← / → Key

These keys move the cursor one character to the left or right.

Operation example: Cursor type "Vertical line"

(1) The cursor is placed as shown in the right.



(2) Press → key.

The cursor moves one character to the right.

Operation example: Cursor type "Block"

(1) The cursor is placed as shown in the right.

(2) Press → key.

The cursor moves one character to the right.

INSERT Key

When cursor type is "block", the data insertion mode is entered. The characters input after the mode is entered will be inserted before the cursor position. When cursor type is "Vertical line", it is always in the data insertion mode, so this key will be ignored.

Operation example:

(1) Move the cursor to the position where the data is to be inserted.

The cursor will move in the data setting area.

(2) Press the INSERT key, and then the data keys.
INSERT 7 7 7

The data will be inserted and the cursor will move to the right.

(Note) The overwrite mode is entered when DELETE, C.B, INPUT, cursor or TAB, etc., keys are pressed, or when the screen is changed.

DELETE Key

(1) When the cursor type is "Vertical line"

This key deletes the character to the right of the cursor.

Operation example:

(a) Move the cursor to the position where the data is to be deleted.

The cursor will move in the data setting area.

(b) Press DELETE key.

The character after the cursor will be deleted and the cursor will not move.

(2) When the cursor type is "Block"

This key deletes the character to the left of the cursor.

Operation example:

(a) Move the cursor to the data to be deleted.

The cursor will move in the data setting area.

(b) Press DELETE key.

The character before the cursor will be deleted and the cursor will not move.

C.B Key

This key deletes all the characters in the data setting area.

Operation example:

(1) Press C.B key.

All the characters in the data setting area will be deleted and the cursor will move to the far left.

(Verticalline)

(Block)

Inputting Operations

In addition to the method of directly inputting numeric data for specific data settings, the method of inputting the operation results using four rules operators and function symbols can be used.

Input method

Input numerical values, function symbols, operators and parentheses () in the data setting area. Press INPUT key to display the operation results.

By specifying a control to reflect the INPUT beforehand, the input data can be reflected to the specified control when pressing INPUT key again.

Setting examples of operators and functional symbols and the results are as follows.

Examples of operator settings, and results		
Operation	**Setting example**	**Result**
Addition	=100+50	150
Subtraction	=100-50	50
Multiplication	=12.3*4	49.2
Division	=100/3	33.3333333
Function	=1.2*(2.5+SQRT(4))	5.4

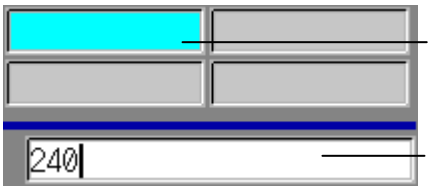
Function symbols, setting examples and results			
Function	**Function symbol**	**Setting example**	**Result**
Absolute value	ABS	=ABS (50-60)	10
Square root	SQRT	=SQRT (3)	1.7320508
Sine	SIN	=SIN (30)	0.5
Cosine	COS	=COS (15)	0.9659258
Tangent	TAN	=TAN (45)	1
Arc tangent	ATAN	=ATAN (1.3)	52.431408

Operation examples

(1)

Set as below and press INPUT key.
=12*20 INPUT

The operation results will appear in the data setting area.



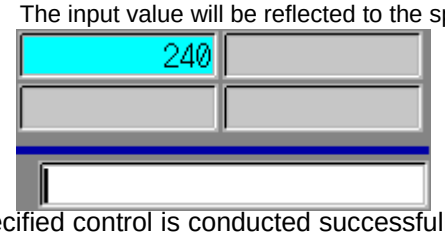
Control to reflect the INPUT

Input box

(2)

Press INPUT key again.

The input value will be reflected to the specified control.



(Note) When reflecting the input data to the specified control is conducted successfully, the data in the data setting area will be cleared. And if it fails, the data will remain displayed in the data setting area.

Precautions for using arithmetic operators and functions

- Division	: An error will occur if the denominator of a division is zero.
- Square root	: An error will occur if the value in parentheses () is negative.
- Trigonometric function	: The unit of angle θ is degree (°).
- Arc tangent	: -90＜calculation result＜90

Restrictions

Followings are the restrictions for operation function.

(1) Always input "="before any characters.
(2) Do not use the following characters as the second or the last character.
- Invalid as second character: *, /,)
- Invalid as last character: *, /, (, +, -
(3) An error will occur when the number of opening and closing parentheses is not equal.
(4) The 360° limit does not apply on the angle. SIN (500) is interpreted as SIN (140).
(5) The exponential setting, like "1.23E-4", cannot be used. The operation result is not displayed with exponential.
(6) It is not possible to set characters exceeding the number of characters which can be input to the data setting area.
(7) It is not possible to omit "0" before a decimal point, like ".5", when inputting operations.
(8) The accuracy is guaranteed for the calculation with 15 digits or less. An unintended rounding will occur to calculations with over 15 digits.
Ex.1) When the 18th to 20th digits are rounded down.
(=12345678901234567890*1 → 12345678901234567000)
Ex.2) When the 18th digit is rounded to the positive direction.
(=123456789012345678*1 → 123456789012345680)
(9) Operators and functions which are not mentioned above, such as "ASIN", cannot be used.
(10) Regardless of the input setting unit and metric system/inch system, the maximum digit number below the decimal point of the operation result is seven.

7.2.14.3 Restrictions

(1) When the control to reflect the INPUT is a text box (when "float" is set in the "Type" property), a value different from the one displayed in the data setting area may be set.
(2) When a value is input after "0", "0" will be attached at the top of the value.
The "0" will be cleared only when an arithmetic processing is executed.

7.2.15 Ten-key object (GSoftKey)

The ten-key is a control which displays numerical values and character strings as well as input keys. The key buttons within the ten-key control and the keys on the operation board can be used as input keys.
The following operations are also available with the ten-key control.
Operation function
The input values can be reflected to another specified control (Specification of a control to reflect the INPUT).
For the specification of a control to reflect INPUT, refer to GCSSoftKeySetRefrectControl in "16.5.27 SoftKey".
In combination with the sub cursor function, the input value can be set in the control at which the sub cursor is pointed and the control can be moved by the arrow key/TAB key.
For the details of sub cursor setting, refer to "9.7 Sub Cursor Setting".

To create an input box object, select [Input Box] from the [Control] menu of NC Designer, or select the icon shown below.

The ten-key is a control which facilitates input processing to the text box and others by saving the need for using macro description. The SW keyboard is a keyboard window which substitutes the NC keyboard. The following table shows the difference in functions.

		Ten-key	SW keyboard
Purpose		SW keyboard control which can interface with text box, PLC text box, NC data text box.	Simple NC keyboard window
		No macro creation needed for data input processing	Macro creation needed for the data input processing
		Layout can be custmized	Layout fixed (Left, center, right)
Function			
	Key type	Three （Decimal/Hexadecimal/ALL key）	Two (Ten-key/ALL key)
	Abs/Inc	Yes	No
	Calculation	Yes	No

7.2.15.1 Property Settings

The property settings for the ten-key object are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Show/hide	:	Select whether the control is displayed or hidden.
Color/pattern	:	Specify the color and pattern of the control.
Display type/ Display format	:	Specify the format of character strings to be displayed in the control.
Password	:	Specify the password.
Character attribute	:	Specify the character attribute of the caption.
Solid frame	:	Specify the solid frame of the control.
Operation function	:	Specify whether or not the operation function is provided.
Input method	:	Specify the input method (Absolute/Incremental) and the switching over between uppercase and lowercase characters.
Sub cursor	:	Specify the display color of the sub cursor.
Input type	:	Specify the input type.
Display type	:	Specify the display type of buttons.
Control display position movement amount	:	Specify the movement amount of the display position.
Callback function	:	Specify the presence of callback functions.

Position/Size

Item	Description
X	Specify the horizontal position from the upper left of the page/view frame of the control (X coordinate) in dots (0 to 2559).
Y	Specify the vertical position from the upper left of the page/view frame of the control (Y coordinate) in dots (0 to 1919).
WIDTH (Note)	Specify the width of the control in dots (8 to 2560).
HEIGHT (Note)	Specify the height of the control in dots (8 to 1920).

(Note) If an area smaller than the entire ten-key is specified, the drawings will not be updated correctly while key input can be handled. Make sure that the specified area is as large as the ten-key.

Show/Hide

Item	Description
Show/Hide	Select whether to display the control.
Title bar Show/Hide	Select "Show" to display and "Hide" not to display the title bar.
Title bar Displayed character string (Note)	Specify the character string to be displayed on the title bar. There are two specification methods for character string. - Select from the registered character string resources. - Newly input character strings.
Original value Show/Hide	Select "Show" to display and "Hide" not to display the original value.

(Note) The maximum number of characters for the title name differs depending on the input type and display type. When the title name exceeds the limit, it will overlap with the movement mark △. Make sure that the title name does not exceed the maximum number of characters.

	Type	Button (Small)		Button (middle)		Horizontal		ALL key type
		Dec.	Hex.	Dec.	Hex.	Dec.	Hex.	
Max. number of characters	M700VW	11	11	18	18	34	43	18
	M700VS/M70V/E70	12	12	20	20	38	49	20

* When characters are input as one-byte characters.

Color/pattern

Item	Description
Original value Background color	Specify the background color of the original value part.
Original value Character color	Specify the character color of the original value part.
Input data display Background color	Specify the background color of input data display part.
Input data display Character color	Specify the character color of input data display part.
Background color at the time of disable	Specify the background color of an input data display part when the entry is allowed but the focus is OFF or the entry is disabled. When the entry is disabled, entered contents will be cleared.

Display type/Display format

Item	Description
Number of the maximum characters	Specify the maximum number of characters to display/set. (1 to 256) Characters exceeding the display range will be displayed by scrolling.

Password

Item	Description
Password setup	Select "Yes" to display entered characters with asterisks (*). Characters will also be displayed with asterisks (*) at original value part.

Character attribute

Item	Description
Cursor Display (Note)	When the entry is allowed and the focus is ON, select "With cursor" to display the cursor and "No effect" not to display the cursor.
Cursor type (Note)	Select the cursor type which will be displayed when the entry is allowed and the focus is ON, from "Vertical line" or "Block".
Cursor background color (Note)	Specify the background color of the sub cursor.
Cursor character color (Note)	Specify the character color of the sub cursor.

(Note) The settings will be valid only when "ALL key type" is selected for the property item "Input type".

Solid Frame

Item	Description
Control Existence of a solid frame	Select the presence of the solid frame between "Yes" and "None."
Control Solid frame	Select the ID of the solid frame resource.

Operation function

Item	Description
Operation function (Note)	When the "Input type" is set to "ALL key type", select whether to provide the operation function between "Yes" and "None." When 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.

(Note) The settings will be valid only when "ALL key type" is selected for the property item "Input type".

Input method

Item	Description
Abs/Inc	Specify whether to directly set the input value to the control when entering the INPUT key (absolute), or to add the positional value of the sub cursor or the control to which INPUT will be reflected, to the input value when entering the INPUT key (incremental). When "Inc" is set, the Inc key button on the ten-key control will be highlighted.
ABC.../abc... (Note)	Select the input method from uppercase/lowercase character. When "abc..." is set, the ABC../abc.. key button on the ten-key control will be highlighted and lowercase character will be selected.

(Note) The settings will be valid only when "ALL key type" is selected for the property item "Input type".

Sub cursor

Item	Description
Sub cursor Background color (Note)	Specify the background color of the control where the sub cursor is displayed.
Sub cursor Character color (Note)	Specify the character color of the control where the sub cursor is displayed.

(Note) For the details of sub cursor setting, refer to "9.7 Sub Cursor Setting".

Input type

Item	Description
Input type	Select the input type for the ten-key from Decimal number/ Hexadecimal/ALL key type.

Display type

Item	Description
Display type	Select the display type of the key buttons from "Small"/"middle"/"Horizontal". The selected display type will become valid when the input type is set to "Decimal number"/"Hexadecimal".

Control display position movement amount

Item	Description
Control display position movement amount	Specify how much the control moves with the position movement button. If the size of the display after the movement with the position movement button exceeds the size of the panel or the window, the display position of the ten-key control will not change. Also, if the ten-key control is placed inside a frame, it will not move beyond the frame. (1 to 2560)

Callback function

Item	Description
OnSubCursorMove	Execute this function after the sub cursor position changed.
OnError	Execute this function if setting the contents to the control to which the INPUT will be reflected failed.
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select " Yes" to add a process to be executed before the panel/window is hidden.

For the other properties, refer to "7.1 Common Functions of Controls".

7.2.15.2 Compliments

Screen configuration

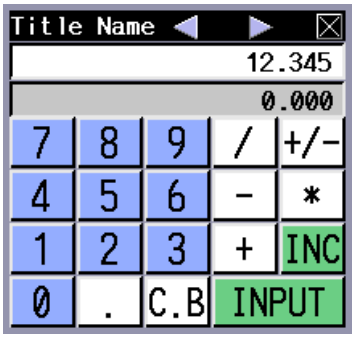
Screen image

The input types of the ten-key control are divided into decimal input (normal/horizontal), hexadecimal input (normal/horizontal), and all key type input (normal). Also, two button sizes are available for decimal input/hexadecimal input (normal).

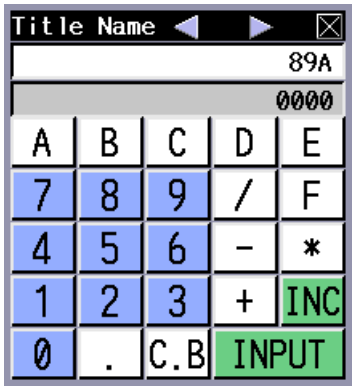
Display image for each input type

The decimal/hexadecimal/all key type input can be switched over by changing the property item "Input type". Images of each input type are shown below.

<Decimal input type>



<Hexadecimal input type>

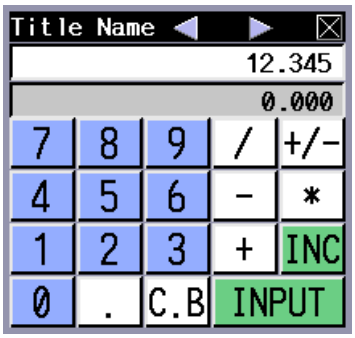


<All key type input>

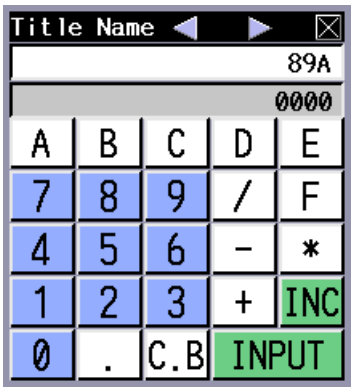
Display image for each display type

When the input type is either decimal or hexadecimal, the layout and size of the buttons can be switched between three patterns; normal button (Small)/normal button (middle)/Horizontal button (Small) by changing the property item "Display type". The images of each display type are shown below.

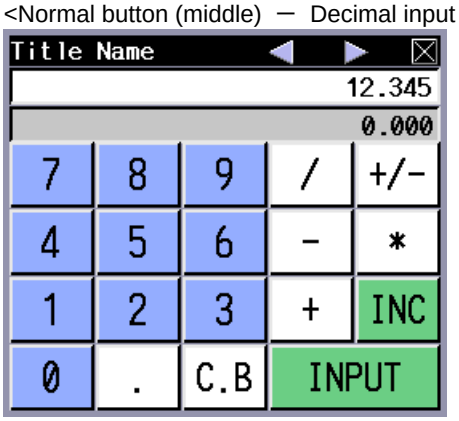
<Normal button (Small) — Decimal input type>



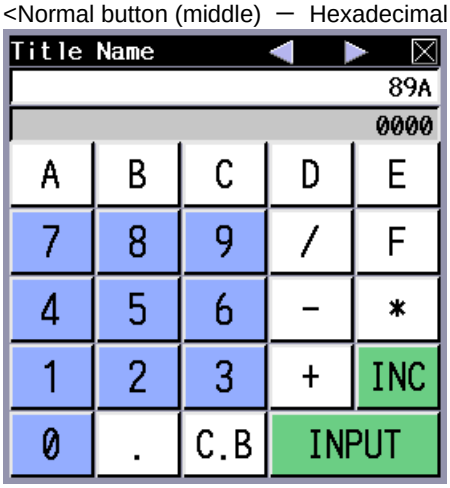
<Normal button (Small) — Hexadecimal input type>



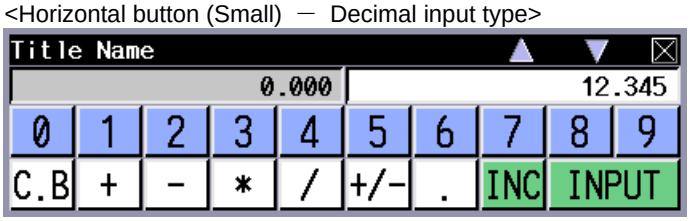
<Normal button (middle) — Decimal input type>



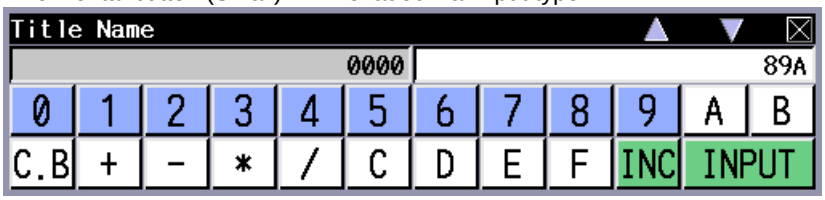
<Normal button (middle) — Hexadecimal input type>



<Horizontal button (Small) — Decimal input type>

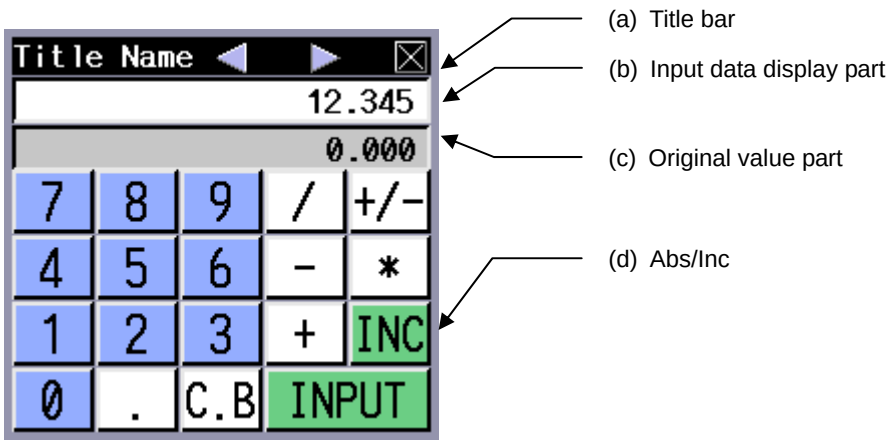


<Horizontal button (Small) — Hexadecimal input type>



Content

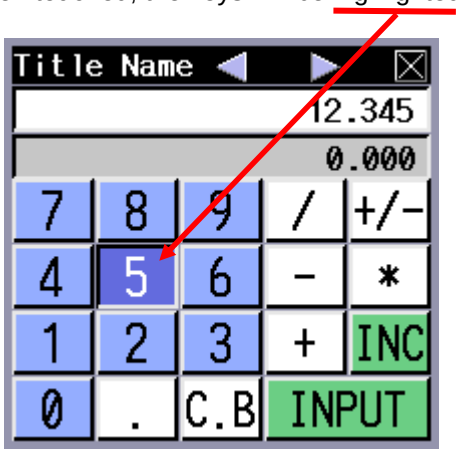
The configuration of a numerical control keypad is shown below. The title bar and the original value part can be hidden by changing the setting in the property item "Title bar Show/Hide" and "Original value Show/Hide".



Name	Details
(a) Title bar	Displays the title name, the position movement button, and the close button. Specify the movement amount handled by the position movement button in the property. The title bar can be hidden by the property setting.
(b) Input data display part	Displays the value to be set. Press the [INPUT] key button to set this value to the control specified to reflect the INPUT.
(c) Original value part	Displays the value in the control specified to reflect the INPUT. Original value part can be hidden by the property setting.
(d) Abs/Inc	Change the setting method between absolute and incremental. When the incremental method is selected ([INC] key is highlighted), the value in the input data display part will be added to the value in the control to reflect the INPUT. But when the character string type of the control is "character string", inputting a value to the control will be disabled.

Highlighting the key button

When touched, the keys will be highlighted as below.

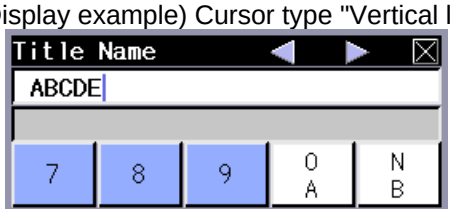


(Note) The position movement button and the close button on the title bar will not be highlighted.

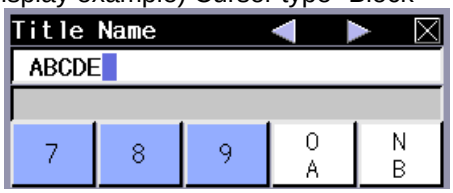
Type of sub cursor

When "With cursor" is selected for the property item "Cursor Display", the type of the cursor will be as follows depending on the property setting "Cursor type".

Display example) Cursor type "Vertical line"



Display example) Cursor type "Block"



Operations with Keys

Buttons on the title bar

The "position movement button" and the "close button" are located on the title bar. The operations of controls with these buttons are explained below. Be aware that, when the title bar is hidden, these buttons cannot be used.

<Position movement button>
Click these buttons to move the ten-key to the direction indicated by the button.
The ten-key will move up and down when the "Input type" is "Decimal number" or "Hexadecimal" and also the "Display type" is "Horizontal". In other cases, it will move to right and left. Set the movement amount in the property item "Position movement".

<Close button>
Click this button to hide the ten-key. When the ten-key is specified as the "input control" in the sub cursor setting, the sub cursor will remain displayed.
The ten-key will reappear if the sub cursor movement is made when the ten-key is hidden. Also, touch the control to which the sub cursor setting is made to display the ten-key at the coordinate set as the "display start position" in the sub cursor setting.
For the details of sub cursor setting, refer to "9.7 Cub Cursor Setting".

Types of key buttons

The operations with key buttons displayed in the ten-key control and with those on the NC keyboard are received.

Keys	Details
[0] to [9], [A] to [Z]	These reflect the input numerals and alphabets to the input data display part.
[.]	This reflects the decimal point to the input data display part.
[+/-]	This highlights the sign of the value in the input data display part.
[+], [—], [*], [/]	These set the four rules operators to the input data display part.
[SP]	This puts a character space to the input data display part.
[C.B]	This clears (blank) the value in the input data display part.
[INPUT]	When an operation is not being performed (when the four rules operators are displayed), this sets the value in the input data display part to the control to reflect the INPUT. When an operation is being performed (when the four rules operators are displayed), this sets the operation result to the input data display part.
[INC]	Change the setting method between absolute and incremental. When the incremental method is selected ([INC] key is highlighted), the value in the input data display part will be added to the value in the control to reflect the INPUT.
[SHIFT]	Press this key to select the characters written in the lower part of each button. Example） Press [SHIFT] and then [G] to input "C".
[ABC…/abc…]	Press this key to switch between uppercase and lowercase alphabets.
Others	The input values will be reflected to the input data display part.

Key type	Key	Operation
Keys related to data input (Keys available for the ten-key control)		
Data setting keys	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h I j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 + - = * / . ; ( ) etc.	Press these keys to set alphabetic characters, numerals and operation symbols, etc.
Data correction keys	INSERT (Data insert key)	When the type of the cursor is "block", the data insertion mode is entered. Press a data setting key to insert a character before the current cursor position. (The overwrite mode is entered when the DELETE, C.B, INPUT, cursor or TAB, etc., keys are pressed, or when the screen is changed.)
	DELETE (Data delete key)	Press DELETE to delete a character in the data setting area. - When the cursor is "Vertical line", the character after the cursor position will be deleted. - When the cursor is "Block", the character before the cursor position will be deleted.
	C･B (Cancel key)	Press C･B to cancel the setting in the data setting area.
Cursor keys	↑ ↓	When the sub cursor settings are made, press these keys to move the target control. (*)The values in the data setting area will not be cleared even if the sub cursor moves.
	|← →|	When the sub cursor settings are made, press these keys to move the target control. (*)The values in the data setting area will not be cleared even if the sub cursor moves.
	← →	Press these keys to move the cursor one character to the left or right in the data setting area. If a control is set as the destination of these arrow keys (←, →) by the sub cursor settings, the cursor will not move within the data setting area and, instead, will move between the target controls. (*)The values in the data setting area will not be cleared even if the sub cursor moves.
INPUT key	INPUT	Press INPUT to fix the data in the data setting area and reflect the input value to the control which is specified to reflect the INPUT. When the sub cursor settings are made, the input value will be reflect before the cursor moves to the target control. When operators are displayed in the ten-key display part, the operation result will be displayed in the input part but the input value will not be reflected to the specified control.
Operation key	SP (Space key)	Inserts a blank. (*) If INPUT is entered when only a space is input, the display characters in the control to which the sub cursor setting is made will be cleared.

(Note) If any key other than those listed above is entered, it will be ignored. If [Key transfer control] is set by the sub cursor setting, the focus will move to the set control and transfers the key. For the sub cursor setting, refer to "9.7 Sub Cursor Setting".

Set the input numerals and alphabetic characters

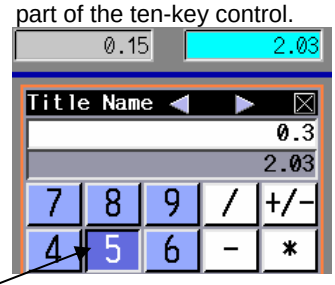
Input numeric, alphabetical and other keys while the focus is placed on the control to display the character strings.
By specifying a control to reflect the INPUT beforehand, the input data can be reflected to the specified control when pressing the INPUT key.

Decimal／Hexadecimal input

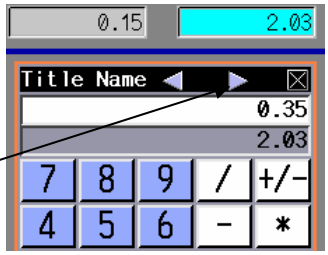
(1)

Input data keys.
Press 0 . 3 5 key buttons.

The input data will be displayed in the input data display part of the ten-key control.



Pressed key button will be highlighted.

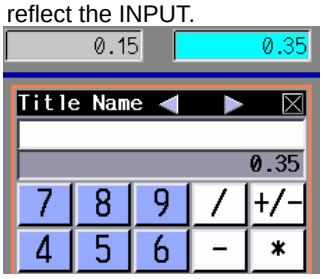


Release the key button to clear the highlighting and to display the value to the input data display part.

(2)

Press the INPUT key button.

The input value will be reflected to the control specified to reflect the INPUT.



(Note 1) Data will be right-aligned in the input data display part and the original value part.
(Note 2) When the setting succeeds, the content in the input data display part will be cleared and the display of the original value part will be updated.
(Note 3) When the setting fails, neither displayed content in the input data display part nor the original value part will be changed.
(Note 4) When "0" before a decimal point is omitted, like . 3 5, ".35" will be displayed in the input data display part. And if the INPUT key is pressed, the character string ".35" will be set in the specified control. The setting will fail if the specified control to reflect the INPUT accepts numerals only.

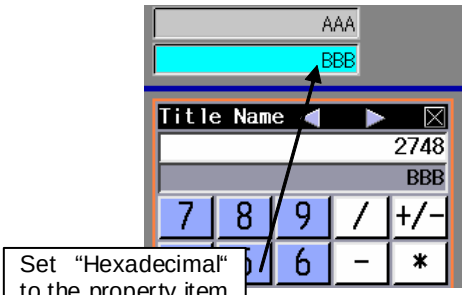
All key type input

(Note 1) Data will be left-aligned in the input data display part and the original value part.
(Note 2) When the setting succeeds, the content in the input data display part will be cleared and the display of the original value part will be updated.
(Note 3) When the setting fails, neither displayed content in the input data display part nor the original value part will be changed.
(Note 4) When the character string type of the control is "character string", press SP key to insert a blank and then press INPUT key button to clear the content of the control. But if more than one blank are inserted, the blanks will be set in the control.

When data is set to the hexadecimal display control from a decimal input ten-key

The value in the input data display part will be converted to hexadecimal number before being reflected to the control to reflect the INPUT. Likewise, when a value is set to a decimal display control from a hexadecimal input ten-key, the value in the input data display part will be converted to decimal number before being reflected. All key type input can be operated as same as when using a decimal input ten-key.

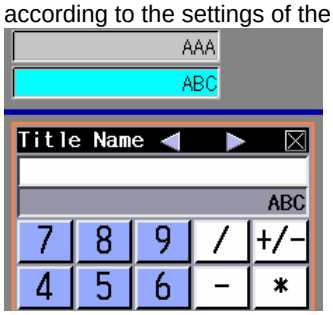
The input data will be displayed in the input data display part of the ten-key.



Set "Hexadecimal" to the property item "Input type".

INPUT

The value in the input data display part will be reflected to the control after being converted according to the settings of the control.



(Note) The content of the control to which the INPUT is to be reflected will directly be displayed in the original value part regardless of the ten-key input type.
When the input type is "Decimal number" and the control to reflect the INPUT is a hexadecimal display control, a hexadecimal number will be displayed in the original value part.

When data setting fails

Setting an input data may fail depending on the property setting of the control to reflect the INPUT.
If the data setting fails, the process written in the callback function OnError will be executed.

Operation in the input data display part

For the all key type input, set the property item "Cursor Display" to "With cursor" to display the cursor in the input data display part.
Key input will be performed to where the cursor is currently displayed.
When a key is input, data will be displayed at the cursor position and the cursor will move a character to the right.
When the cursor is hidden, the data will be overwritten as well as when the cursor type is "Block".
The data will be inserted to the far right when the input type is decimal or hexadecimal.

Numeric keys/Alphabetical keys

For the details of operation, refer to "7.2.14 Input box(GInputBox)".

← / → key

For the details of operation, refer to "7.2.14 Input box(GInputBox)".

(Note 1) The cursor will not moved during decimal and hexadecimal input. Characters will always be inserted to the far right even using ← / → key.
(Note 2) When a control is specified as the destination of sub cursor movement in the sub cursor setting, ← / → key operation gives priority to the sub cursor movement between controls. So the ← / → key operation will become unable to move the sub cursor within the input data display part.

INSERT key

For the details of operation, refer to "7.2.14 Input box(GInputBox)".
(Note) For the decimal and hexadecimal input, it is always in the data insertion mode, so this key will be ignored.

DELETE key

(1) All key type input
When the cursor type is "Vertical line", the character after the cursor position will be deleted. When the cursor is "Block", the character before the cursor position will be deleted.
For the details of operation, refer to "7.2.14 Input box(GInputBox)".
(Note) When a control is specified as the destination of sub cursor movement in the sub cursor setting, the sub cursor will not move within the input data display part. So when the cursor type is "Vertical line", DELETE key will become unable to delete data. Use C.B key to delete characters in the input data display part.
(2) For the decimal and hexadecimal input, the character at the far right will be deleted.

C.B key

This key deletes all the characters in the input data display part. When an operator is being displayed, the operator will be cleared, too.
For the details of operation, refer to "7.2.14 Input box(GInputBox)".

Input operation

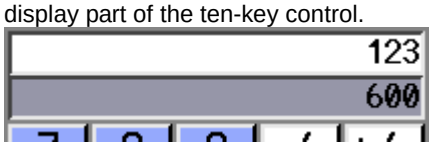
The ten-key control provides a four rules operation function. The following operation procedure applies also when the input type is set to "ALL key type" and the operation function is set to "Yes".

Operation example

(1)

Input the first item.
Press 1 2 3 key buttons.

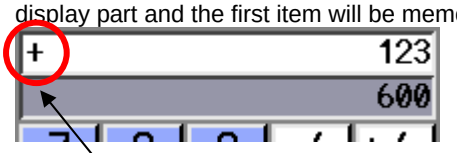
The input data will be displayed in the input data display part of the ten-key control.



(2)

Press the operation key button +.

The operator will be displayed in the input data display part and the first item will be memorized.

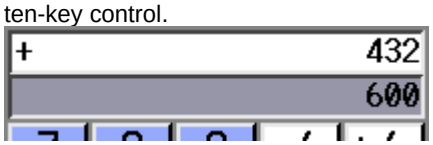


The operator will be displayed.

(3)

Input the second item.
Press 4 3 2 key buttons.

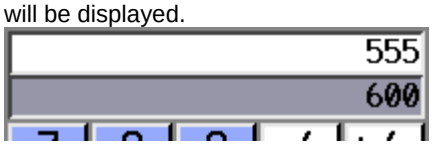
The first item will be cleared and the second item will be displayed in the input data display part of the ten-key control.



(4)

Press the INPUT key button.

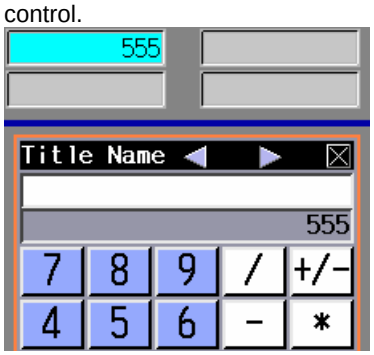
The operator will disappear and the operation result will be displayed.



(5)

Press the INPUT key button again.

The operation result will be reflected to the specified control.



(Note) Press an operator key button after inputting the second item, instead of the INPUT key button, to display the operation result and the operator in the input data display part and to continue with another operation.

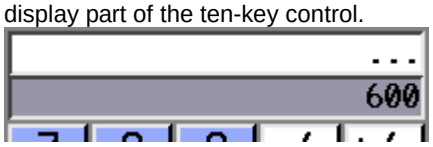
Example of operation failure

If the operation fails, neither displayed content in the input data display part or the original value part will be updated. But the operator will be cleared and the operator input will be canceled.

(1)

Input the first item.
Press [.][.][.] key buttons.

The input data will be displayed in the input data display part of the ten-key control.



(2)

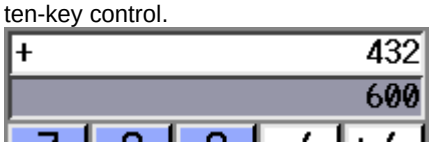
Press the operation key button [+].

The operator will be displayed in the input data display part.

(3)

Input the second item.
Press [4][3][2] key buttons.

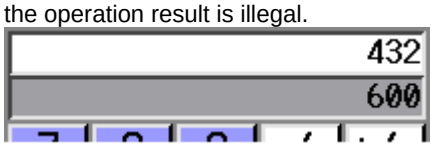
The first item will be cleared and the second item will be displayed in the input data display part of the ten-key control.



(4)

Press the [INPUT] key button.

The operator will be cleared but the content in the input data display part will not be changed because the operation result is illegal.



Canceling the operation input

Press the INPUT key after inputting the operator to cancel the operator input.

(1)

Input the first item.
Press 1 2 3 key buttons.

(2)

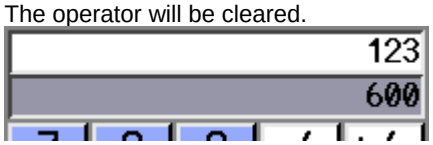
Press the operation key button +.

The operator will be displayed in the input data display part.

(3)

Press the INPUT key button.

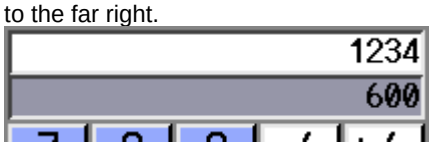
The operator will be cleared.



(4)

Press 4 key button.

The first item will not be cleared and "4" will be added to the far right.



The operator display when character string is displayed by scrolling

If a character string exceeding the display range of the input data display part is input, a scroll display will be applied. The characters to be scrolled are those set with the data setting keys and the +/- key. When an operator is being displayed, its position is fixed and will not be scrolled.

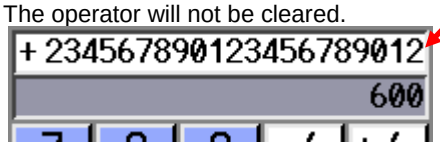
(1)

Press the operator key button + and input as many characters as possible to display.

(2)

Press 2 key button.

The "1" at the far left will be hidden and "2" will appear at the far right.
The operator will not be cleared.



Precautions for using operators and functions

(1) An error will occur if the denominator of a division is zero.
(2) When more than one operator is pressed one after another, the last one will become valid.
(3) Several decimal points can be displayed at a time, but an operation will be illegal with several decimal points displayed.
(4) The operation will be illegal when a "0" before a decimal point is omitted, like [.] [3] [5].
(5) In the all key type input mode, the operation will be handled in decimal number. Operations in hexadecimal number are illegal.

Restrictions of operation function

Followings are the limitations of operation function.

(1) The accuracy is guaranteed for the calculation with 15 digits or less in decimal number. An unintended rounding will occur to calculations with over 15 digits.
Ex.1) When the 18th to 20th digits are rounded down.
(=12345678901234567890*1 → 12345678901234567000)
Ex.2) When the 18th digit is rounded to the positive direction.
(=12345678901234678*1 → 12345678901234680)
The calculation accuracy is the same for switching the sign of values using the [+/-] key button.
(2) Regardless of the input setting unit and metric system/inch system, the maximum digit number below the decimal point of the operation result is 7.
(3) Operation using () cannot be performed.
(4) Operation using a function symbol not provided in the key buttons cannot be performed.
(5) Operation will not be performed by pressing "=" key.

7.2.15.3 Restrictions

(1) When the property item "Original value Show/Hide" is set to "Show" and the character string set in the control to reflect the INPUT exceeds the "maximum number of characters" set in the property, the original value part will be blank.
(2) When a text box is set as the control to reflect the INPUT (when the "type" property is set to "float"), a value different from the one displayed in the input data display part may be set in the text box.
(3) If the ten-key is moved over a control with a display updating cycle (such as counter and F command), it will not be displayed correctly. Make sure that the layout does not cause any overlapping. Also, hide the title bar to invalidate the movement button operation.
(4) When a value is input after "0", "0" will be attached at the top of the value. The "0" will be cleared only when an arithmetic processing is executed.
(5) The hexadecimal input type ten-key cannot handle decimal numbers. A decimal point will not be displayed even by pressing the "." key button.

7.3 NC Control Object

7.3.1 Counter (GNXCounter); Counter Display Part

The counter display part can display the current position, workpiece coordinate position, etc.

To create a counter display part, select [Counter] from the [Control] menu or select the following icon in NC Designer.

7.3.1.1 Property Settings

The property settings of the Counter are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Character attribute	:	Specify the character attribute of captions.
Counter kind	:	Specify the kind of counter to display.
Display	:	Specify the number of axes to display and the presence of space between the rows.
Axis status	:	Specify the display of the axis status.
Title	:	Specify the display of the title.
Axis name	:	Specify the display of the axis name.
Coordinate	:	Specify the display of the coordinate.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal: Normal font Normal Bold: Normal bold font Middle: Middle-sized font Big: Font with double height and width Huge: Font with triple height and width

Counter Kind

Item	Description
CounterKind	Specify what kinds of coordinate values to display. Current Position: Current position Work Coordinate Position Workpiece coordinate position Machine Position: Machine's position Program Position: Program position Remain Command: Remaining command Manual Interruption Amount: Manual interruption amount Next Command: Next command Restart Position: Restart position Restart Remain Distance: Remaining distance for the restart Tip Wk Coord Position: Tip workpiece coordinate position Pulse(Tool Axis Movement): Tool axis movement Tip Mach Position: Tip machining position Relative Position: Relative position All the other settings are invalid.

Display

Item	Description
LineNumber	Specify the number of axes to display. (1 to 8)
LineGapVisibleLineGapVisible	Specify whether the space is provided or not between the rows.

Axis Status

Item	Description
AxisStatusVisible	Specify the presence of the axis status.
AxisStatusType	Select “Normal Type”.
AxisStatusForeColor	Specify the character color for the axis status display.
AxisStatusBackColor	Specify the background color for the axis status display.

Title

Item	Description
TitleForeColor	Specify the character color of the title.
TitleBackColor	Specify the background color of the title.

Axis name

Item	Description
AxisNameForeColor	Specify the character color for the axis name display.
AxisNameBackColor	Specify the background color for the axis name display.

Coordinate

Item	Description
ValueForeColor	Specify the character color for the coordinate value display.
ValueBackColor	Specify the background color for the coordinate value display.
CharacterNumber	Specify the digit number of the coordinate value. (1 to 16)
AxisCross	Specify whether to enable to switch over the display of axes names during mixed synchronization control. 1: Enable 0: Disable

For the other properties, refer to "7.1 Common Functions of Controls".

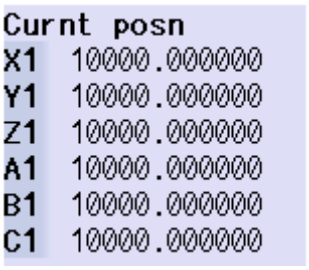
7.3.1.2 Complements

Screen Specifications

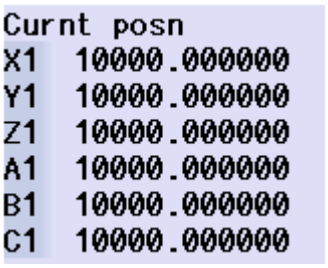
Screen Images

The followings are the images of the counter display part for each font type.

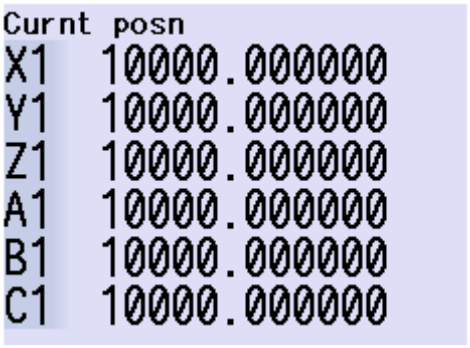
Example 1: Display in normal font (Normal)



Example 2: Display in normal bold font (Normal Bold)

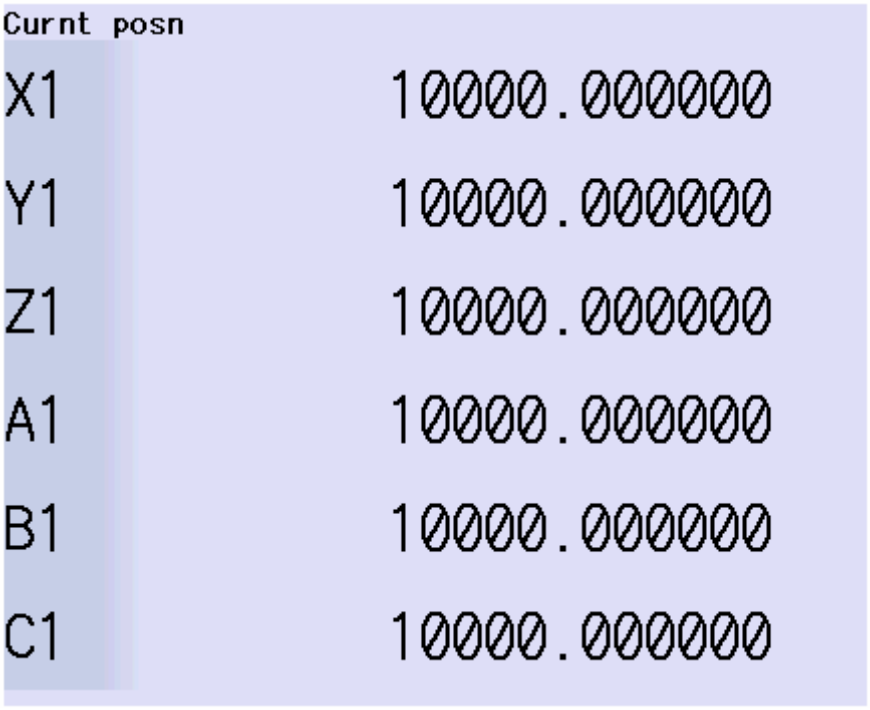


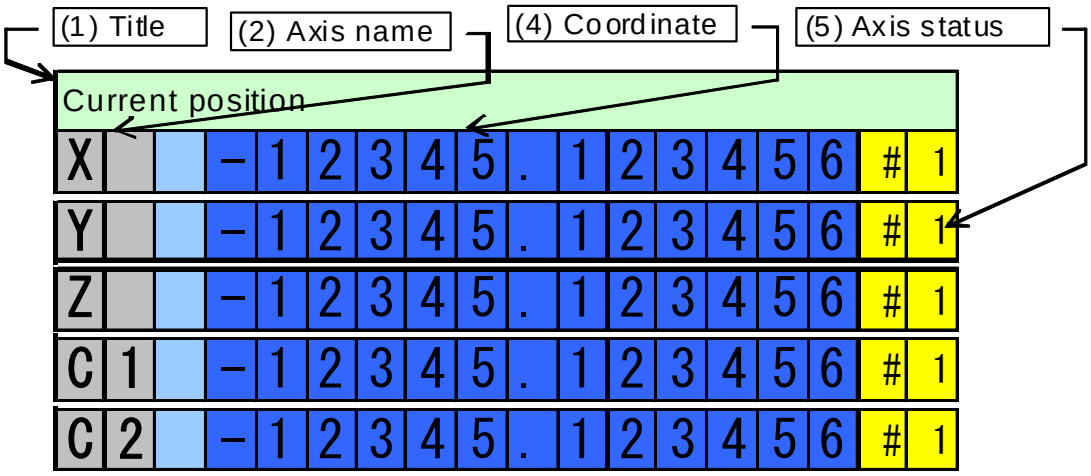
Example 3: Display in middle-sized font (Middle)



Example 4: Display in the font with double height and width (Big)

Example 5: Display in the font with triple height and width (Huge)





Details of the Axis Status Display

1st digit	2nd digit	Description
#	Reference point No. 1 or 2 or 3 or 4	The reference point return is executed to the reference point that corresponds to the reference point No.
]	[	In emergency stop status.
M	R	The mirror image is executed.
>	<	The axis is being removed.
C	T	The auxiliary axis is selected by NC axis/auxiliary axis switchover function.

Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
1002	axisno Number of axes	Specify the number of axes in each part system, as well as the number of PLC axes.	Part system 1: 1 to 8 Part system 2: 0 to 8 PLC axis: 0 to 2
1022	axname2 Displayed axis name	Specify the displayed axis name for each axis.	1st letter: A to Z 2nd letter: A to Z or numerals 1 or 2 digits
1041	I_inch Initial state (inch)	Specify the unit system for the program movement amount when the power is turned ON or reset, as well as the unit system for the position display.	0: Metric system 1: Inch system
1069	no_dsp Axis with no display	Specify an axis that is not displayed.	0: Displayed 1: Not displayed
1123	origin Origin set prohibit	Prohibits the origin set and origin cancel operations.	0: Not prohibit 1: Prohibit
1494 (PR)	dsp_ax_change Axis order of counter display	Set this in order to change the axis order of counter display. The axes will be displayed in ascending order of the setting values. However, axis whose setting is "0" will be given the lowest priority. (Note 1) When the same value is set for more than one axis, axis that is displayed on the left side on the parameter screen will be first displayed. (Note 2) When both of mixed synchronization control (option) and interchange coordinate position display (1280 ext16/bit2" OFF) are valid, and when there are two or more valid part systems, this parameter will be ignored.	0: This axis is displayed after the display of the axes with setting value "1" to "8". 1 to 8: These axes are displayed in ascending order. Others: The setting value will be treated as "0".
12800 (PR)	chgauxno Auxiliary axis number	Set the axis No. to be controlled as auxiliary axis using auxiliary axis interface.	0 to 16

7.3.2 CycleTime (GNXCycleTime); Cycle Time Display Part

The cycle time display part is used to display the automatic start-up time and the cycle time.

Automatic start-up time (STL)	Total accumulated time during the automatic operation, from when the automatic start-up button is pressed in the memory (tape) mode or MDI to when the feed hold stop, block stop or reset button is pressed.
Cycle time (CYC)	The automatic operation time from when the automatic start-up button is pressed in the memory (tape) mode or MDI to when the feed hold stop, block stop or reset button is pressed. This is preset to "0" by turning the power OFF.
Date (DAT) ……	The current date set in the NC is displayed. Year: 4 digits, Month: 2 digit, Date: 2 digit (YYYY.MM.DD)
Time (TIM) ……....	The current time set in the NC is displayed with the 24-hour system. (HH:MM:SS)
Power-ON time (PON) ……..	This displays the total integrated time of the time from NC power ON to OFF. (HH:MM:SS)
Automatic operation time (OP) ……..	The total integrated time from NC power ON to OFF. (HH:MM:SS)
External accumulated time1 (EX1) …	This content differs depending on the PLC sequence.(HH:MM:SS)
External accumulated time2 (EX2) …	This content differs depending on the PLC sequence.(HH:MM:SS)

To create a cycle time display part, select [CycleTime] from the [Control] menu or click the following icon in NC Designer.

7.3.2.1 Property Settings

The property settings of the CycleTime are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Character attribute	:	Specify the character attribute of captions.
Time type	:	Specify the time type
Display	:	Specify the existence of frame, as well as space between the rows.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Character attribute

Item	Description
FontType	Specify the font size and thickness. Normal: Normal font Normal Bold: Normal bold font Big: Font with double height and width

Time type

Item	Description
UpperTimeType LowerTimeType	Specify the time type to be displayed at first time. DATE …….. Date TIME …….. Time POWERON ……. Power ON AUTORUN ……. Automatic operation AUTOEXEC ……... Automatic start-up OUTSIDE1 ……. External accumulated time1 OUTSIDE2 …….. External accumulated time2 CYCTIME ……. Cycle time DEFAULT ……. Default (upper: automatic start / lower: cycle time)

Display

Item	Description
LineGapVisible	Specify whether the space is provided or not between the rows.
FrameVisible	Specify the frame is provided or not.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

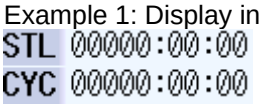
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.2.2 Complements

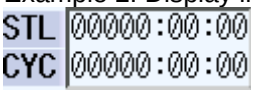
Screen Specifications

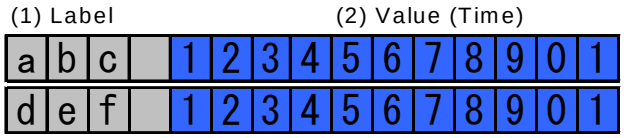
Screen Images

Example 1: Display in normal font (Normal) without frame



Example 2: Display in normal font (Normal) with frame





7.3.3 Feedrate (GNXFeedrate); F Display Part

F display part shows the vector direction speed currently being moved in during interporation feed, the speed of the axis with highest speed during each axis independent feed. This part also shows dwell (code: G04).
Setting the property (speed display type) enables the display of tool tip speed.

To create a F display part, select [Feedrate] from the [Control] menu or select the following icon in NC Designer.

7.3.3.1 Property Settings

The property settings of the Feedrate are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Character attribute	:	Specify the character attribute of captions.
Speed display type	:	Specify the type of the speed display.
Display	:	Specify the frame is provided or not.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal: Normal font Big: Font with double height and width

Speed Display Type

Item	Description
SpeedType	Specify the type of the speed display. Normal: Normal speed display Tool Tip Speed: Tool tip speed display

Display

Item	Description
FrameVisible	Specify the frame is provided or not.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

For the other properties, refer to "7.1 Common Functions of Controls".

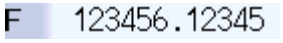
7.3.3.2 Complements

Screen Specifications

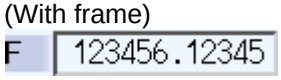
Screen Images

<Speed display type (normal speed display)>

(Without frame)

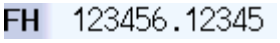


(With frame)

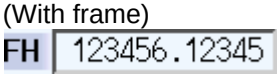


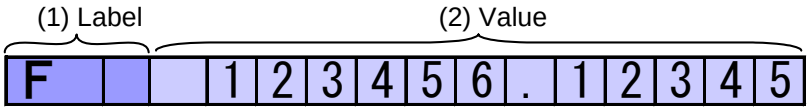
<Speed display type (tool tip speed display)>

(Without frame)



(With frame)





Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
#1125	real_f Actual feedrate display	Specify the feedrate display on the monitor screen.	0: Command speed 1: Real movement feedrate

(Note) The tool tip speed display shows the command speed regardless of the setting of the parameter above.

7.3.4 GModal M (GNXGModal); M System Modal Display Part

The G modal display part is used to show each modal state.

To create a M system modal display part, select [GModal M] from the [Control] menu or select the following icon in NC Designer.

7.3.4.1 Property Settings

The property settings of the GModal M are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Display	:	Specify whether the space is provided or not between the rows and whether to display MSTB.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

Display

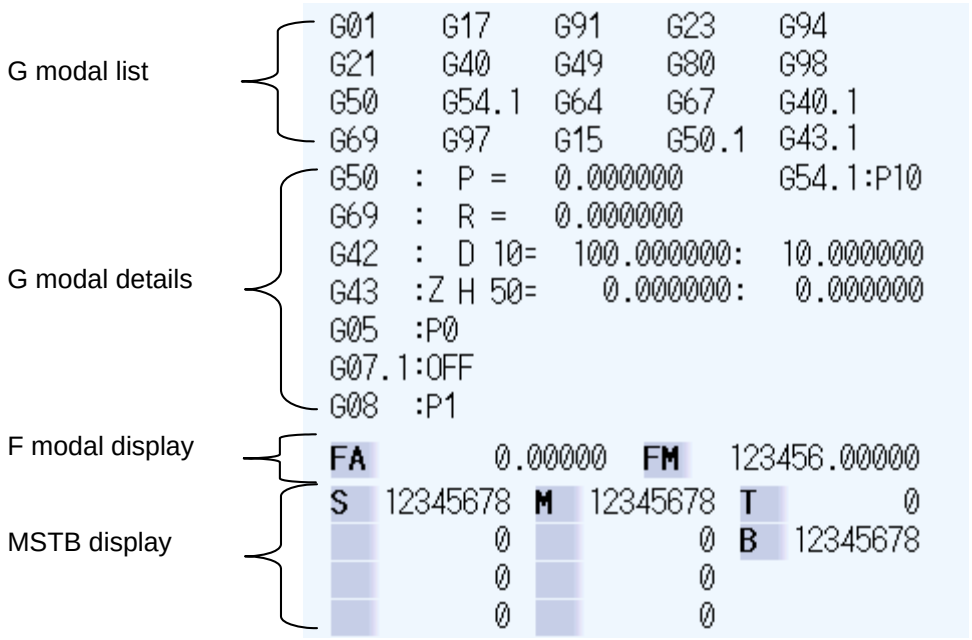
Item	Description
LineGapVisible	Specify whether the space is provided or not between the rows.
MSTBVisible	Specify whether to display MSTB.

For the other properties, refer to "7.1 Common Functions of Controls".

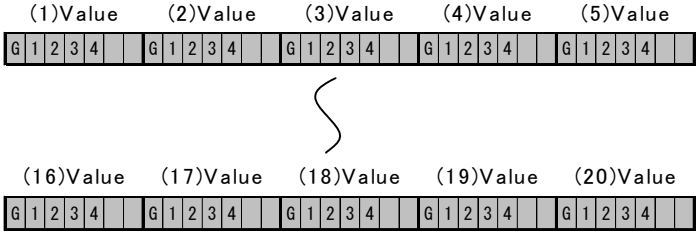
7.3.4.2 Complements

Screen Specifications

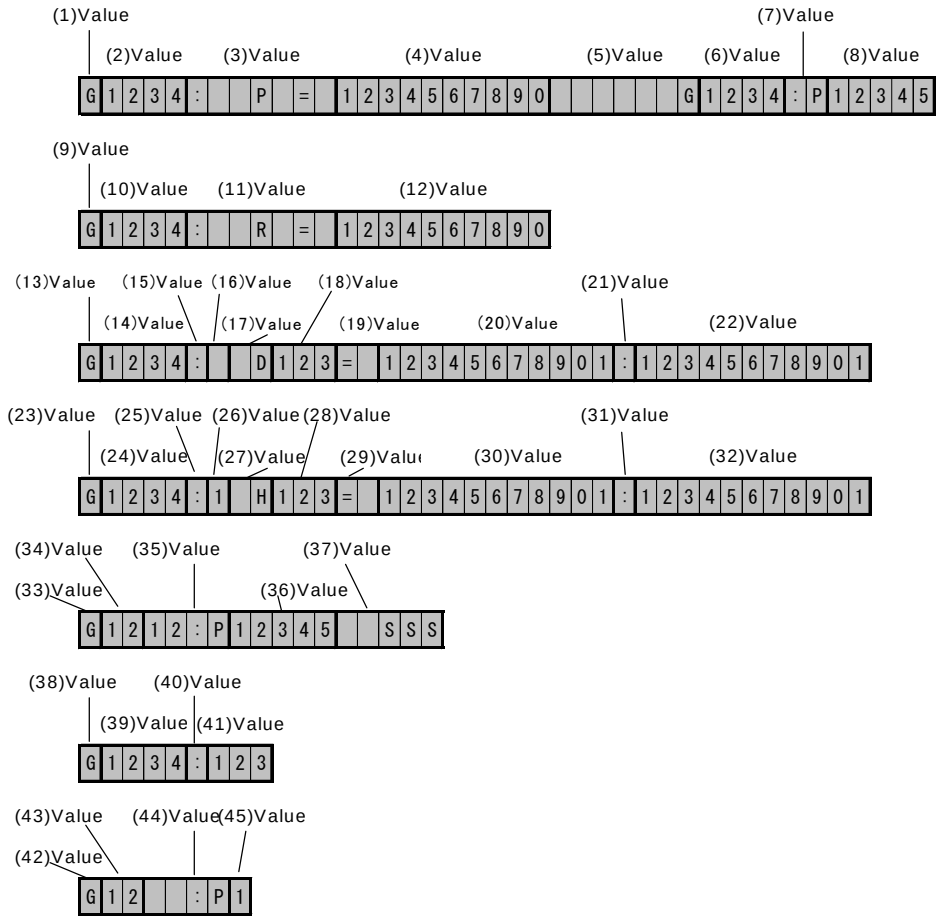
Screen Images



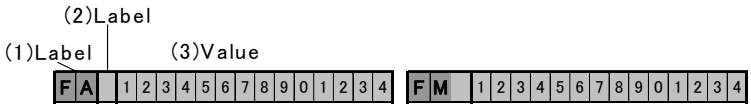
<G modal list>



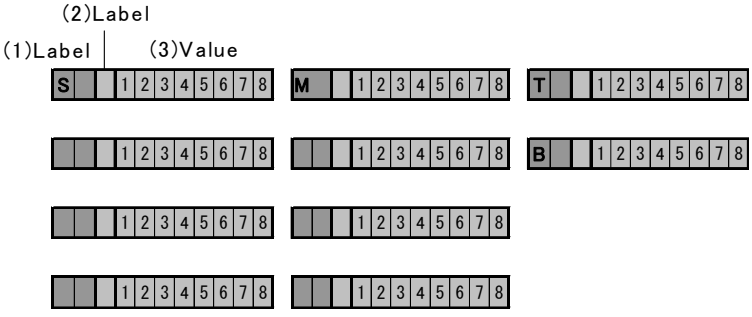
<G modal details>



<F modal display>

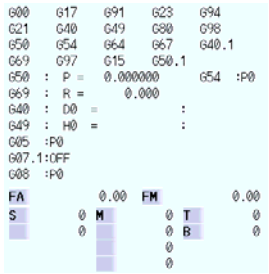
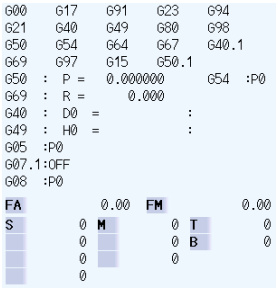
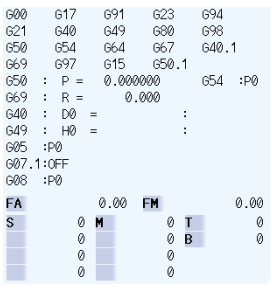
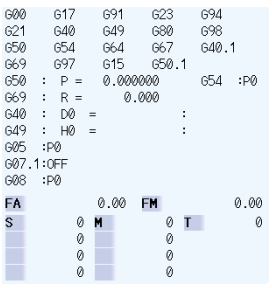


<MSTB display>



Displayed Item

Displayed part	Displayed item	Details
G modal list	G01 to G94 | G69 to G43.1	G command modal currently executed Group 1 to 20
G modal details	G50 : P = 0.000000	Details of G group 11 P = Scaling factor
	G54.1 : P10	Details of G group 12 P No.
	G69 : R = 0.000000	Details of G group 16 R = Coordinate rotation angle
	G42 : D 10= 100.000000: 10.000000	Details of G group 7 D offset No. = Offset amount : Wear amount
	G43 : Z H 50= 0.000000: 0.000000	Details of G group 8 Offset axis name H Offset No. = Offset amount : Wear amount
	G05.1 : P10000 SSS	High-speed machining mode (fixed to G05) P No. (or Q No.) SSS control ON
	G07.1 : OFF	Details of G group 21 (fixed to G07.1) Cylindrical interporation mode (ON/OFF)
	G08 : P1 *1	High-accuracy control mode (fixed to G08) P0: High-accuracy control mode OFF P1: High-accuracy control mode ON **(Note)** The high-accuracy control mode is not displayed in the G modal details when: • All of the options of the high-accuracy control (G61.1), the high-speed and high-accuracy control 1 and 2, and the spline interporation are OFF. • The base common parameter “#1267 ext03 BIT0” (G code type) is set to “0”.
F modal display	FA 0.00000	F modal value of the program command currently executed
	FM 123456.00000	Manual feedrate

Displayed part	Displayed item	Details
MSTB display	S 12345678 0 0 0	S command (S1 to S4) The number of values displayed is decided by the parameter “#1039 spinno (Number of spindles)”. [Display examples] (Example 1) Parameter: "#1039 spinno" is set to "3"  (Example 2) Parameter: "#1039 spinno" is set to "2" 
	M 12345678	M command (M1 to M4) The number of values displayed is decided by the parameter “#12005 Mfig (Number of M)”. [Display examples] (Example 1) Parameter: "#12005 Mfig" is set to "3" 
	T 0	T command
	B 12345678	2nd miscellaneous function (B) command Displays the 2nd miscellaneous function modal value of the program command being executed. Whether the value is displayed or not depends on the setting of the parameter “#1170 M2name (second miscellaneous code)”. [Display examples] (Example 1) Parameter: "#1170 M2name" is set to "B"  (Example 2) Parameter: No value is set for "#1170 M2name" 

Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
1003 (PR)	iunit Input setup unit	Specify the input setting value for each part system and the PLC axis. The unit of the parameters is decided by this setting.	B,C,D,E
1013 (PR)	axname Axis name	Specify each axis' name address with an alphabetic character.	Axis addresses
1037 (PR)	cmdtyp Command type	Specify the program G code series.	1 to 8
1039 (PR)	spinno Number of spindles	Specify the existence of a spindle	0 to 4
1041 (PR)	I_inch Initial state (inch)	Specify the unit system for the program movement amount when the power is turned ON or reset and for position display. Specify the internal unit system.	0: Metric system 1: Inch system
1130	set_t Display selected tool number	Specify the tool command value display. 0: T-modal value of program command is displayed. 1: Tool number sent from PLC is displayed.	0/1
1170	M2name Second miscellaneous code	Set the address code when using the 2nd miscellaneous command.	Address with A, B and C that is not used with "#1013 axname" or "#1014 incax"
1267 (PR)	ext03 (bit0) G code type	Select the high-speed high-accuracy G code type. 0: Conventional format 1: MITSUBISHI CNC special format	0/1
1520 (PR)	Tchg34 Additional axis tool compensation operation	Select whether to carry out the additional axis' tool compensation function with the 3rd axis or 4th axis. 0: Select 3rd axis 1: Select 4th axis	0/1
3001 to 3004	slimt 1 to 4 Limit rotation speed	Set spindle rotation speed for maximum motor rotation speed with gears 00, 01, 10, 11. (Set the spindle speed for the S analog output 10V.)	0 to 99999 [r/min]
3005 to 3008	smax 1 to 4 Maximum rotation speed	Set maximum spindle rotation speed with gears 00, 01, 10, 11. Set to slimt ≥ smax. By comparing the S command value and the value of gear 1 to 4, a spindle gear shift command will be output automatically.	0 to 99999 [r/min]

<table>
<tr><th>No.</th><th>Name</th><th>Details</th><th>Setting range</th></tr>
<tr><td>12005
(PR)</td><td>Mfig
Number of M</td><td>Number of M: Set the number of M that can be specified within the same block.
M binary: Refer to the following table.
<table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>1 to 4

-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12006
(PR)</td><td>Mbin
M binary</td><td></td><td></td></tr>
<tr><td>12008
(PR)</td><td>Sbin
S binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".

(Note 1) Sbin can be set with -1, 0 and 1, but the S command cannot be BCD output.
If BCD (0) is selected for Sbin, it will be handled as a singed binary (-1).</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12010
(PR)</td><td>Tbin
T binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12012
(PR)</td><td>Bbin
B binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
</table>

Restrictions

When no M command has been executed, the value in the M display shows “0”.

7.3.5 GModal L (GNXGModal_L); L System Modal Display Part

The G modal display part (L system) is used to show each modal state.

To create a L system modal display part, select [GModal L] from the [Control] menu or select the following icon in NC Designer.

7.3.5.1 Property Settings

The property settings of the GModal L are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Display	:	Specify whether the space is provided or not between the rows.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

Display

Item	Description
LineGapVisible	Specify whether the space is provided or not between the rows.

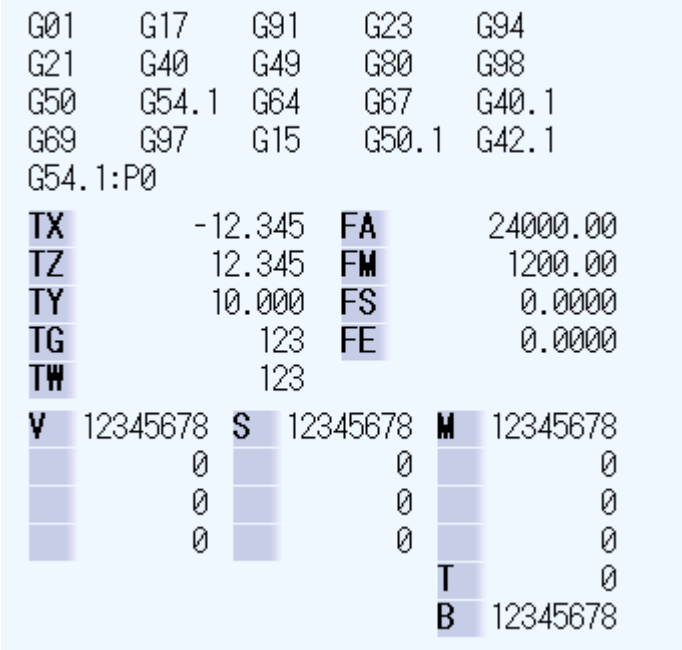
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.5.2 Complements

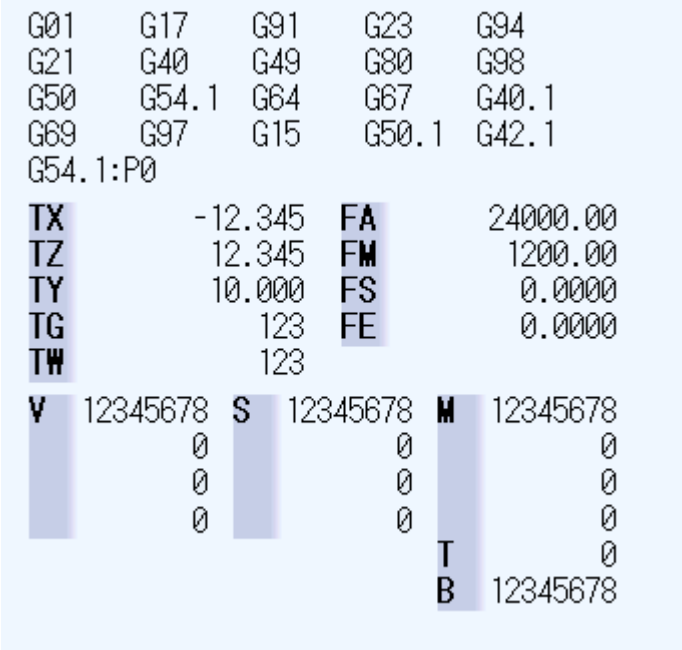
Screen Specifications

Screen Images

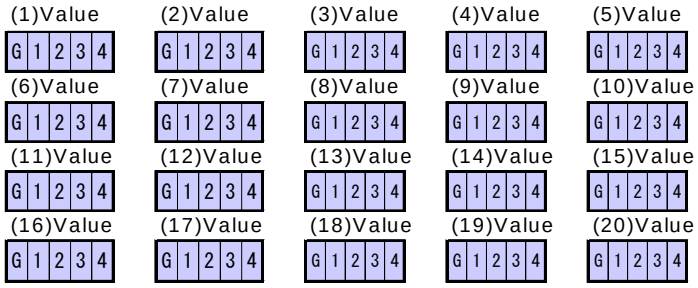
<Displayed with space between the rows (default)>



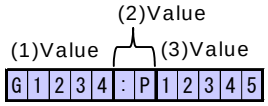
<Displayed without space between the rows>



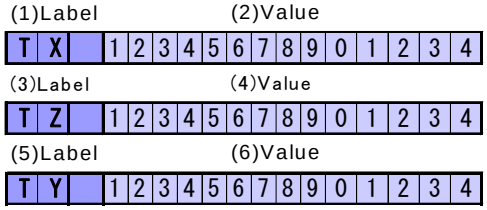
<G modal list>



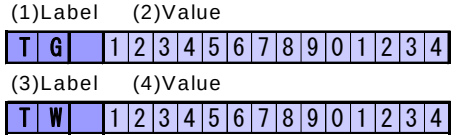
<G modal details>



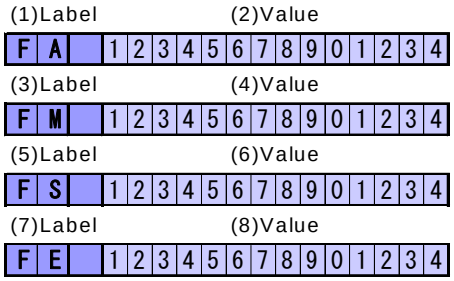
<Tool length and wear compensation amount display>



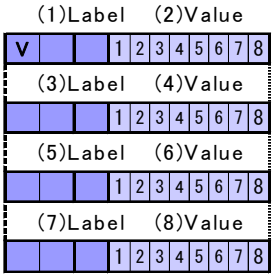
<Compensation No. display>



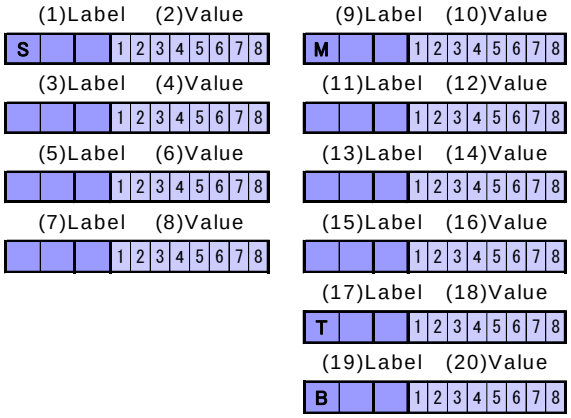
<F modal display>



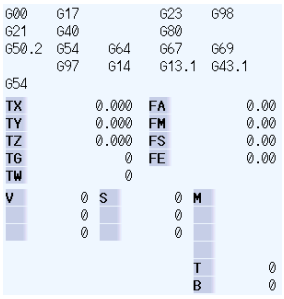
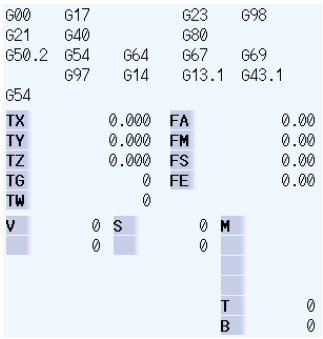
<Constant surface speed control display (for surface speed S)>

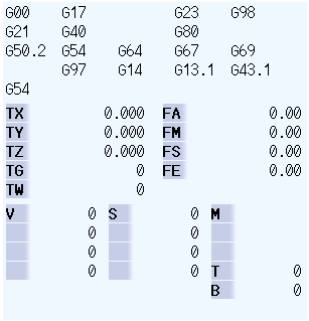
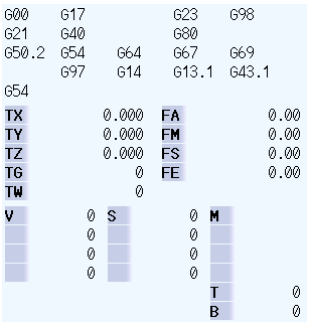
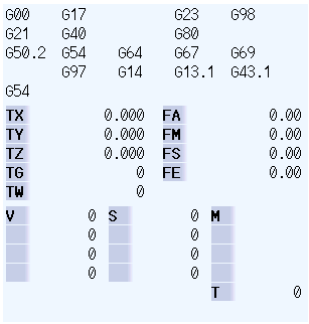


<MSTB display>



Displayed Item

Displayed part	Displayed item	Details
G modal list	G01 to G94 | G69 to G42.1	G command modal currently executed Group 1 to 20
G modal details	G54.1:P10	Details of G group 12 (Note) P No.
Tool length and wear compensation amount display	TX:-12.345	Total amount of the tool length and the wear compensation amount of the 1st axis of the tool currently used
	TZ: 12.345	Total amount of the tool length and the wear compensation amount of the 2nd axis of the tool currently used
	TY: 10.000	Total amount of the tool length and the wear compensation amount of the additional axis of the tool currently used
Compensation No. display	TG: 123	Tool length compensation No.
	TW: 123	Wear compensation No.
F modal display	FA 24000.00	Unsynchronized feedrate modal value of the program command currently executed (mm/min)
	FM 1200.00	Manual feedrate (mm/min)
	FS 0.0000	Synchronized feedrate modal value of the program command currently executed (mm/rev)
	FE 0.0000	Synchronized feedrate modal value of the thread lead command currently executed (mm/rev)
Constant surface speed control display (for surface speed S)	V 12345678 0 0 0	Spindle rotation speed modal (V1 to V4) value when the surface speed is constant. The number of values displayed is decided by the parameter “#1039 spinno (Number of spindles)”. [Display examples] (Example 1) Parameter: "#1039 spinno" is set to "4"  (Example 2) Parameter: "#1039 spinno" is set to "2" 

Displayed part	Displayed item	Details
MSTB display	S 12345678 0 0 0	S modal (S1 to S4) value of the program command currently executed. The number of values displayed is decided by the parameter “#1039 spinno (Number of spindles)”, which is the same as in the case of the constant surface speed control display (for surface speed S). See the display example of the constant surface speed control display (for surface speed S).
	M 12345678	M modal (M1 to M4) value of the program command being executed. (Up to 4 pairs)The number of values displayed is decided by the parameter “#12005 Mfig (Number of M)”. [Display examples] (Example 1) Parameter: "#12005 Mfig" is set to "3” 
	T 0	T modal value of the program command being executed.
	B 12345678	2nd miscellaneous function modal value of the program command being executed. Whether the value is displayed or not depends on the setting of the parameter “#1170 M2name (second miscellaneous code)”. [Display examples] (Example 1) Parameter: "#1170 M2name" is set to "B”  (Example 2) Parameter: No value is set for "#1170 M2name” 

(Note) Compliment: the data format of the details of G group No.12 is as follows.

G code	Function	Format of the display
G54	Workpiece coordinate system selection 1	G5x :P (x = 4 to 9)
G55	Workpiece coordinate system selection 2	
G56	Workpiece coordinate system selection 3	
G57	Workpiece coordinate system selection 4	
G58	Workpiece coordinate system selection 5	
G59	Workpiece coordinate system selection 6	
G54.1	Workpiece coordinate system selection The number of pairs can be extended to 0, 48 or 96 at the option settings.	G54.1:Pyy (yy = 1 to 96)

Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
1003 (PR)	iunit Input setup unit	Specify the input setting value for each part system and the PLC axis. The unit of the parameters is decided by this setting.	B,C,D,E
1013 (PR)	axname Axis name	Specify each axis' name address with an alphabetic character.	Axis name
1037 (PR)	cmdtyp Command type	Specify the program G code series.	1 to 8
1039 (PR)	spinno Number of spindles	Specify the existence of a spindle	0 to 4
1041 (PR)	I_inch Initial state (inch)	Specify the unit system for the program movement amount when the power is turned ON or reset, as well as the unit system for the position display. Specify the internal unit system.	0:Metric system 1:Inch system
1130	set_t Display selected tool number	Specify the tool command value display. 0 : T-modal value of program command is displayed. 1 : Tool number sent from PLC is displayed.	0/1
1170	M2name Second miscellaneous code	Set the address code when using the 2nd miscellaneous command.	Address with A, B and C that is not used with "#1013 axname" or "#1014 incax"
1520 (PR)	Tchg34 Additional axis tool compensation operation	Select whether to carry out the additional axis' tool compensation function with the 3rd axis or 4th axis. 0: Select 3rd axis 1:Select 4th axis	0/1
3001 to 3004	slimt 1 to 4 Limit rotation speed	Set spindle rotation speed for maximum motor rotation speed with gears 00, 01, 10, 11. (Set the spindle speed for the S analog output 10V.)	0 to 99999 [r/min]
3005 to 3008	smax 1 to 4 Maximum rotation speed	Set maximum spindle rotation speed with gears 00, 01, 10, 11. Set to slimt ≥ smax. By comparing the S command value and the value of gear 1 to 4, a spindle gear shift command will be output automatically.	0 to 99999 [r/min]

<table>
<tr><th>No.</th><th>Name</th><th>Details</th><th>Setting range</th></tr>
<tr><td>12005
(PR)</td><td>Mfig
Number of M</td><td>Number of M: Set the number of M that can be specified within the same block.</td><td>1 to 4</td></tr>
<tr><td>12006
(PR)</td><td>Mbin
M binary</td><td>M binary: Refer to the following table.
<table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12008
(PR)</td><td>Sbin
S binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".

(Note 1) Sbin can be set with -1, 0 and 1, but the S command cannot be BCD output.
If BCD (0) is selected for Sbin, it will be handled as a singed binary (-1).</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12010
(PR)</td><td>Tbin
T binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12012
(PR)</td><td>Bbin
B binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
</table>

7.3.6 GModal Simple (GNXGModalSimple); Simple Modal Display Part

The G modal display part (simple) is used to show each modal state.

To create a simple modal display part, select [GModal Simple] from the [Control] menu or select the following icon in NC Designer.

7.3.6.1 Property Settings

The property settings of the GModal Simple are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Character attribute	:	Specify the character attribute of captions.
Display	:	Specify whether the space is provided or not between the rows.
Value	:	Specify the display of the value.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal Normal font Big Font with double height and width

Display

Item	Description
LineGapVisible	Specify whether the space is provided or not between the rows.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

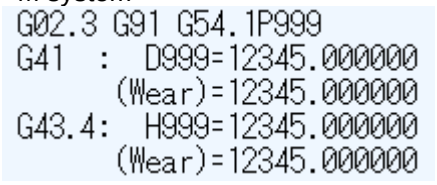
For the other properties, refer to "7.1 Common Functions of Controls".

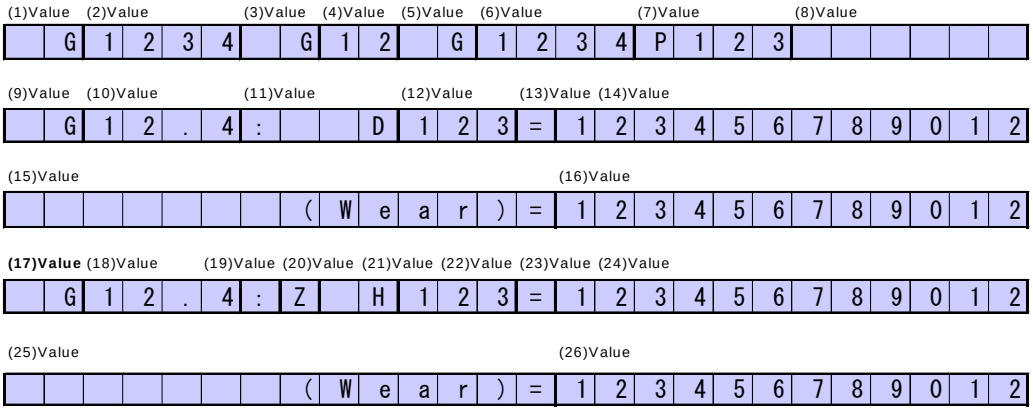
7.3.6.2 Complements

Screen Specifications

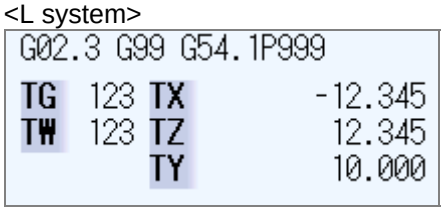
Screen Images

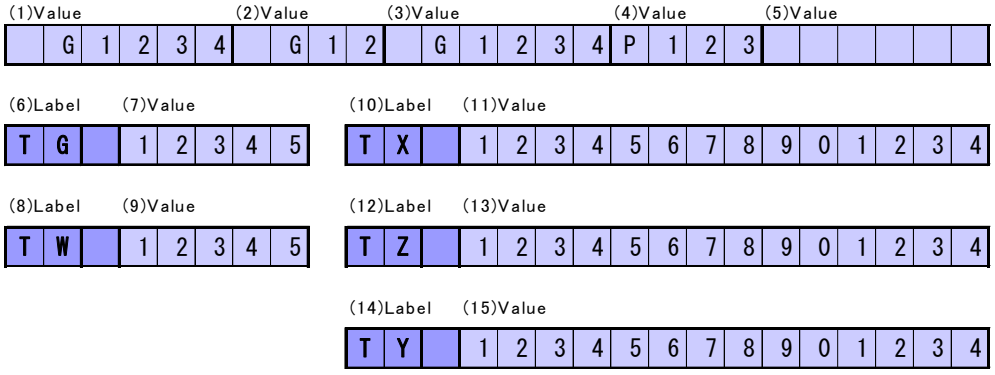
<M system>





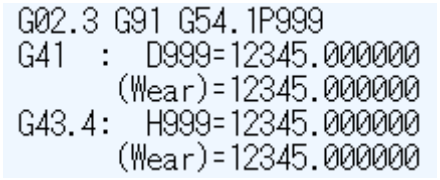
<L system>



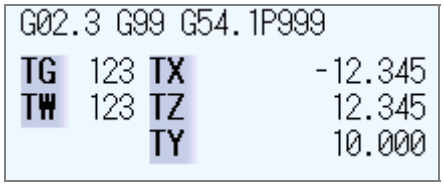


Displayed Item

<M system>

Displayed part	Details	
	Displays the modal states.	
	G02.3	The G command modal state of group No.1 that is being executed
	G91	The G command modal state of group No.3 that is being executed
	G54.1P999	Selected coordinate system
	G41 : D999 =12345.000000 (Wear)=12345.00000	Tool radius offset modal Offset No. Tool radius offset amount Tool radius wear amount
	G43.4: Z H999 =12345.000000 (Wear)=12345.00000	Tool length offset modal Offset axis name Offset No. Tool length offset amount Tool length wear amount

<L system>

Displayed part	Details	
	Displays the modal states.	
	G02.3	G command modal state of group No.1 that is currently executed
	G99	G command modal state of group No.5 that is being executed
	G54.1Ｐ999	Selected coordinate system
	TX:-12.345	Total amount of the tool length and the wear compensation amount of the 1st axis of the tool being used. (Note 1)
	TZ: 12.345	Total amount of the tool length and the wear compensation amount of the 2nd axis of the tool being used. (Note 2)
	TY: 10.000	Total amount of the tool length and the wear compensation amount of the additional axis of the tool being used. (Note 3)
	TG: 123	Tool length compensation No.
	TW: 123	Wear compensation No.

(Note 1) Displays the axis name that is set for the 1st axis in the part system with “#1013 axname”, the base axis specifications parameter.

(Note 2) Displays the axis name that is set for the 2nd axis in the part system with “#1013 axname”, the base axis specifications parameter.

(Note 3) If the base specifications parameter “#1520 Tchg34 (additional axis tool compensation operation)” is set to “0”, this displays the axis name that is set for the 3rd axis in the part system with “#1013 axname”, the base axis specifications parameter. When "#1520 Tchg34” is set to “1”, displays the one for the 4th axis.

Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
1003 (PR)	iunit Input setup unit	Specify the input setting value for each part system and the PLC axis. The unit of the parameters is decided by this setting.	B,C,D,E
1013 (PR)	axname Axis name	Specify each axis' name address with an alphabetic character.	Axis name
1037 (PR)	cmdtyp Command type	Specify the program G code series. 1: System 1 (for M) 2: System 1 (for M) 3: System 2 (for L)	1 to 8
1041 (PR)	I_inch Initial state (inch)	Specify the unit system for the program movement amount when the power is turned ON or reset, as well as the unit system for the position display. Specify the internal unit system.	0: Metric system 1: Inch system
1130	set_t Display selected tool number	Specify the tool command value display. 0: T-modal value of program command is displayed. 1: Tool number sent from PLC is displayed.	0/1
1520 (PR)	Tchg34 Additional axis tool compensation operation	Select whether to carry out the additional axis' tool compensation function with the 3rd axis or 4th axis. 0: Select 3rd axis 1: Select 4th axis	0/1

7.3.7 LoadMeter (GNXLoadMeter); Load Meter Display Part

The Load meter display part can display the spindle load and Z axis load in the bar graphs by using user PLC.
(When the load meter is not set by user PLC, these are not displayed on the screen.))
Two load meters are displayed by using four lines (the area of the spindle standby and the load meter) when the spindle standby is not displayed.
One load meter is displayed by using two lines (the area of the spindle standby and the load meter) when the spindle standby is displayed.

To create a LoadMeter display part, select [LoadMeter] from the [Control] menu or select the following icon in NC Designer.

7.3.7.1 Property Settings

The property settings of the LoadMeter are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Display type	:	Specify the display of the spindle load and the Z axis load.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.
CharacterNumber	Specify the size of the Load meter display part. (4 to 50)

Display Type

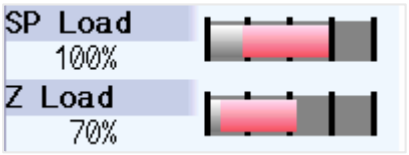
Item	Description
DispType	Specify the display of the spindle load and the Z axis load. 0: Displays the spindle load only 1: Displays the Z axis load only 2: Displays the both spindle and Z axis loads

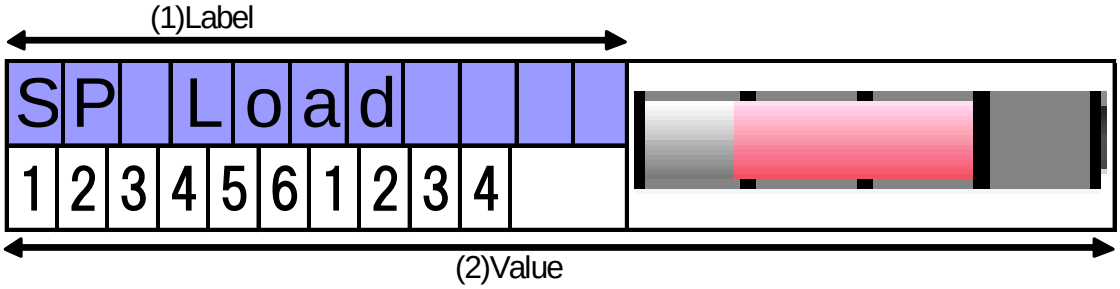
For the other properties, refer to "7.1 Common Functions of Controls".

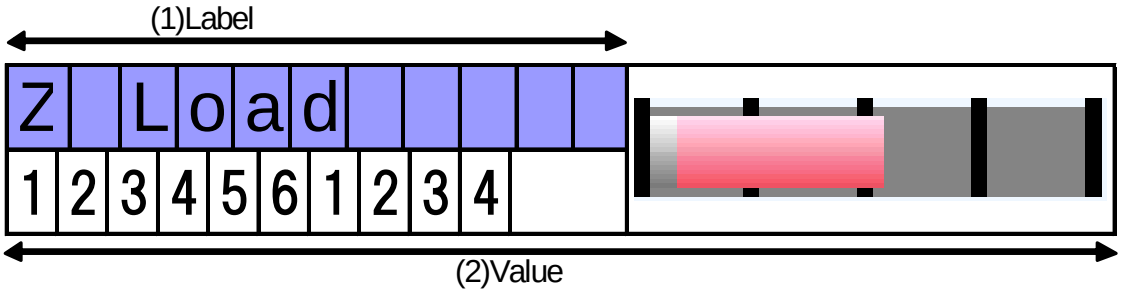
7.3.7.2 Complements

Screen Specifications

Screen Images





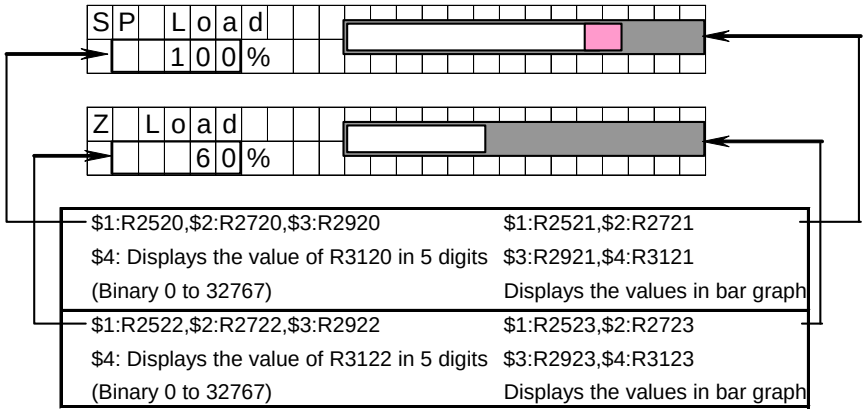


File Registers (R) for the Load Meter Display

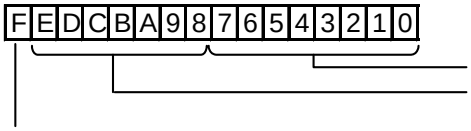
		$1	$2	$3	$4
Load meter 1	For numeral display	R2520	R2720	R2920	R3120
	For bar graph display	R2521	R2721	R2921	R3121
Load meter 2	For numeral display	R2522	R2722	R2922	R3122
	For bar graph display	R2523	R2723	R2923	R3123

(Note) Machines without part system use the display for $1.

Screen Image of the Load Meter Display Part and the Correspondence of the File Register (R)



Contents of R2521,R2721, R2921,R3121, R2523,R2723, R2923,R3123



Load meter display is valid

(a) Whole length of the bar graph
(b) Length of the bar in white
Difference of (a) and (b) is shown in pink.

Restrictions

The Load meter display control does not show the scale and its markings, even if they have been set in the ladder program.

7.3.8 MSTB (GNXMSTB); MSTB display part

The MSTB display part can display each command of spindle function (S), miscellaneous function (M), tool function (T) and 2nd miscellaneous function (B).

To create a MSTB display part, select [MSTB] from the [Control] menu or select the following icon in NC Designer

7.3.8.1 Property Settings

The property settings of the MSTB are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Number of displays	:	Specify the number of the MSTB commands displayed.
Character attribute	:	Specify the character attribute of captions.
Display type	:	Select the property setting (in “Number of displays”) or the NC parameter setting to give priority when specifying the number of the MSTB commands to display.
Display	:	Specify whether to make frame, the space between lines and scroll bar visible and whether to activate the three columns display as well as the number of digits to be displayed.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Number of Displays

Item	Description
S_Number	Specify the number of the S commands displayed (0 to 4).
M_Number	Specify the number of the M commands displayed (0 to 4).
T_Number	Specify the number of the T command displayed (0 or 1).
B_Number	Specify the number of the B command displayed (0 or 1).

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal: Normal font Big: Font with double height and width

Display Type

Item	Description
DisplayType	Select the property setting (in “Number of displays”) or the NC parameter setting to give priority when specifying the number of the MSTB commands to display. Type0: The number of the MSTB commands displayed is specified by the property setting (in “Number of displays”). Type1: The number of the M commands displayed is specified by the parameter “#12005 Mfig (Number of M)”. (When the setting value of the parameter “#12005 Mfig (Number of M)” is larger than that of the property settings (in “Number of displays”), the property setting value is applied to the number of the M commands displayed. The number of the other STB commands displayed depends on the property settings (in “Number of displays”). Type2: The number of the S commands displayed is specified by the parameter “#1300 ext36 (bit 0) (Multiple spindle control II)”. (When the parameter “#1300 ext36 (bit 0) (Multiple spindle control II)” is set to “1”, the number of the S commands displayed is also set to “1”, regardless of the setting of the parameter “#1039 spinno (Number of spindles)”. The number of the other MTB commands displayed depends on the property settings (in “Number of displays”). When the parameter “#1300 ext36 (bit0) (Multiple spindle control II)” is set to “0”, the number of the MSTB commands displayed depends on the property settings (in “Number of displays”).

Display

Item	Description
LineGapVisible	Specify whether the space is provided or not between the lines.
FrameVisible	Specify the frame is provided or not.
ThreeSequencesType	Specify whether to display the S commands, M commands, and T, B commands in the three columns.
ScrollBarVisible	Specify whether to enable the scroll bar. (Note 1) The scroll bar will not be displayed when the three columns display is valid. (Note 2) When the display digit is 0, the scroll bar will not be displayed. (Note 3) When the display width is narrow, the scroll bar will not be displayed.
LineNumber	Specify the display digit. 0: All the lines 1 to 12: Only the specified lines (Note) When the three columns display is valid, all the data will be displayed regardless of the display digit.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

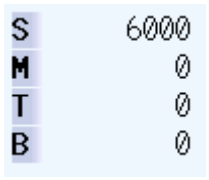
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.8.2 Complements

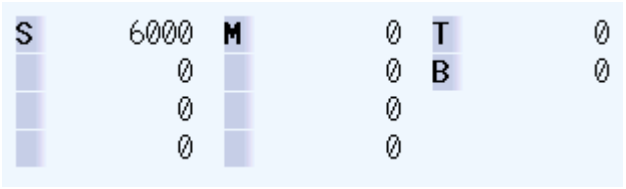
Screen Specifications

Screen Images

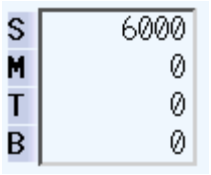
<Standard display>
(No frame, with space between every two lines)



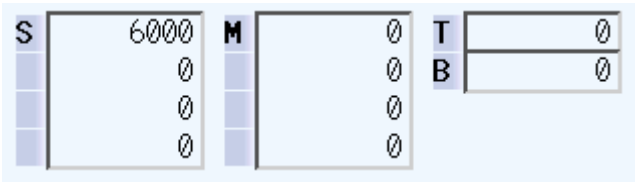
<Three columns display>
(No frame, with space between every two lines)



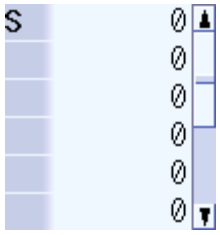
(With frame, with space between every two lines)

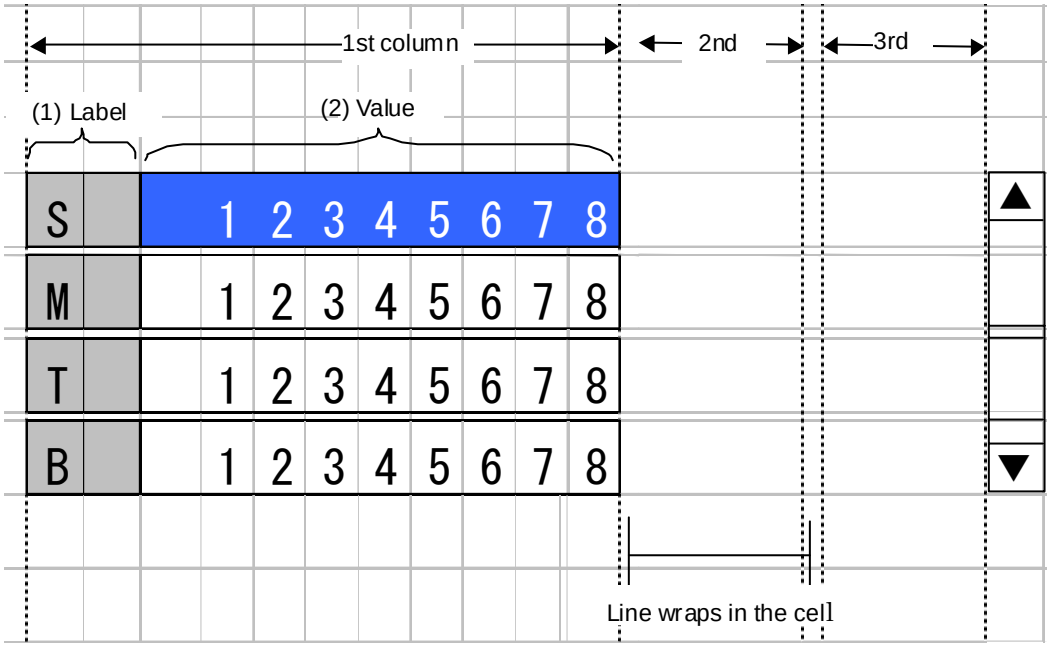


(With frame, with space between every two lines)



(No frame, with space between every rows, scroll bar, number of lines 6)





Related NC Parameters

The following table shows the related parameters.

<table>
<tr><th>No.</th><th>Name</th><th>Details</th><th>Setting range</th></tr>
<tr><td>1039
(PR)</td><td>spinno
Number of spindles</td><td>Specify the existence of a spindle</td><td>0 to 4</td></tr>
<tr><td>1130</td><td>set_t
Display selected tool number</td><td>Specify the tool command value display.
0: T-modal value of program command is displayed.
1: Tool number sent from PLC is displayed.</td><td>0/1</td></tr>
<tr><td>1170</td><td>M2name
2nd miscellaneous code</td><td>Set the address code when using the 2nd miscellaneous command.</td><td>Address with A, B and C that is not used with "#1013 axname" or "#1014 incax"</td></tr>
<tr><td>1228</td><td>aux12 (bit7)
Protection with manual value command</td><td>Set up this option to protect a manual value command.
0: Does not protect the manual value command.
1: Protects the manual value command.</td><td>0/1</td></tr>
<tr><td>1300</td><td>ext36 (bit0)
Multiple spindle control II</td><td>Select multiple spindle control I or II.
0: Multiple spindle control I (L system only)
1: Multiple spindle control II (select from ladder)</td><td>0/1</td></tr>
<tr><td>3001 to 3004</td><td>slimt 1 to 4
Limit rotation speed</td><td>Set spindle rotation speed for maximum motor rotation speed with gears 00, 01, 10, 11. (Set the spindle speed for the S analog output 10V.)</td><td>0 to 99999 [r/min]</td></tr>
<tr><td>3005 to 3008</td><td>smax 1 to 4
Maximum rotation speed</td><td>Set maximum spindle rotation speed with gears 00, 01, 10, 11. Set to slimt ≥ smax. By comparing the S command value and the value of gear 1 to 4, a spindle gear shift command will be output automatically.</td><td>0 to 99999 [r/min]</td></tr>
<tr><td>12005
(PR)</td><td>Mfig
Number of M</td><td rowspan="2">Number of M: Set the number of M that can be specified within the same block.
M binary: Refer to the following table.
<table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>1 to 4</td></tr>
<tr><td>12006
(PR)</td><td>Mbin
M binary</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Details</th><th>Setting range</th></tr>
<tr><td>12008
(PR)</td><td>Sbin
S binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".

(Note 1) Sbin can be set with -1, 0 and 1, but the S command cannot be BCD output.
If BCD (0) is selected for Sbin, it will be handled as a singed binary (-1).</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12010
(PR)</td><td>Tbin
T binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
<tr><td>12012
(PR)</td><td>Bbin
B binary</td><td><table><tr><th>Data type</th><th>Output data</th></tr><tr><td>0</td><td>BCD</td></tr><tr><td>1</td><td>Unsigned binary</td></tr><tr><td>-1</td><td>Signed binary</td></tr></table>For unsigned binary:
The absolute value 1 is output for -1.
For signed binary:
-1 is output as "0xFFFFFFFF".</td><td>-1: Signed binary
0: BCD
1: Unsigned binary</td></tr>
</table>

Restrictions

(1) The macro interruption codes (M96, M97) and subprogram call codes (M98, M99) will not be processed if they are issued.

(2) When no M command has been executed, the M command value area shows “0”.

7.3.9 ONB (GNONB); ONB Display Part

The ONB display part displays the program No., sequence No. and block No. currently being executed.
When a subprogram is being executed, the subprogram's program No., sequence No., block No. and percentage display are displayed.

To create a ONB display part, select [ONB] from the [Control] menu or select the following icon in NC Designer.

7.3.9.1 Property Settings

The property settings of the ONB are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Character attribute	:	Specify the character attribute of captions.
Display	:	Specify the presence of frame, as well as space between the lines and columns.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Device display	:	Specify whether or not to display the device.
Nest level display	:	Specify whether or not to display the nest level.
Callback function	:	Specify whether or not the callback functions are provided.
DisplayPercent	:	Specify whether or not to display the percentage.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal: Normal font Big: Font with double height and width

Display

Item	Description
LineGapVisible	Specify whether the space is provided or not between the lines.
SequenceGapS	Specify whether the space is provided or not between the columns.
FrameVisible	Specify the frame is provided or not.

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

Device Display

Item	Description
DisplayDeviceName	Specify whether or not to display the device.

Nest Level Display

Item	Description
DisplayNestLevel	Specify whether or not to display the nest level.

DisplayPercent

Item	Description
DisplayPercent	Specify whether or not to display the percentage.

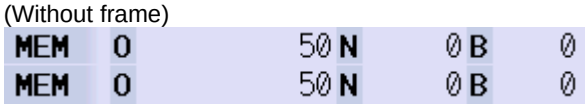
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.9.2 Complements

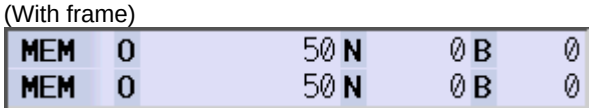
Screen Specifications

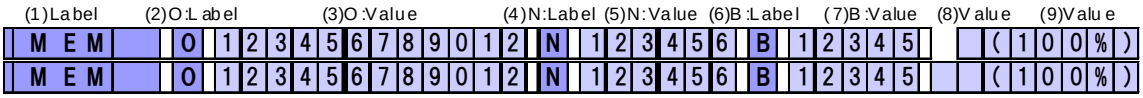
Screen Images

(Without frame)



(With frame)





Details of the Device Name Display

Device name	Details
MEM	Memory is selected.
DS	DS is selected.
232C	Serial is selected.
HD	HD (hard disk) is selected.
MemC	Memory card is selected.
FD	FD is selected.

*MDI mode displays "MDI" in the "0" value.

7.3.10 ProgramBuffer (GNXPrgBuff); Program Buffer Display Part

The program buffer display part displays the contents of the machining program currently being executed.
The block being executed in the program currently is highlighted.

To create a Program buffer display part, select [ProgramBuffer] from the [Control] menu or select the following icon in NC Designer.

7.3.10.1 Property Settings

The property settings of the ProgramBuffer are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Part system designation	:	Specify the part system.
Display	:	Specify the number of lines to display the machining programs, the number of characters in each line, and the presence of space between the lines.
Value	:	Specify the display of the value.
Character attribute	:	Specify the character attribute of captions.
Callback function	:	Specify whether or not the callback functions are provided.

Part system designation

Item	Description
NumberOfSystems	Specify the part system.

Display

Item	Description
LineNumber	Specify the number of lines to display the programs. (1 to 30)
CharacterNumber	Specify the number of characters in each line to display the programs. (2 to 80)
LineGapVisible	Specify whether the space is provided or not between the lines.

Value

Item	Description
ValueForeColor	Specify the character color of the usual value area.
ValueBackColor	Specify the background color of the usual value area.
ValueReverseForeColor	Specify the character color of the value area when selected and reversed.
ValueReverseBackColor	Specify the background color of the value area when selected and reversed.

Character attribute

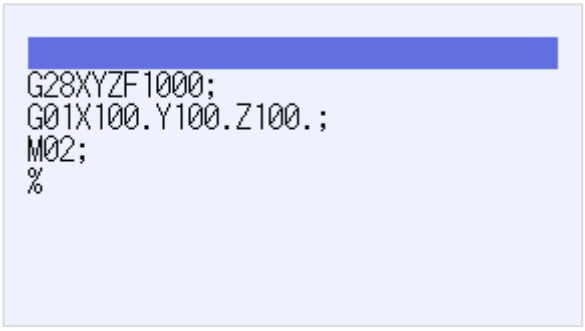
Item	Description
FontType	Specify the font size and thickness. Normal …….. Normal font Normal Bold ……. Normal bold font Middle …… Middle-sized font

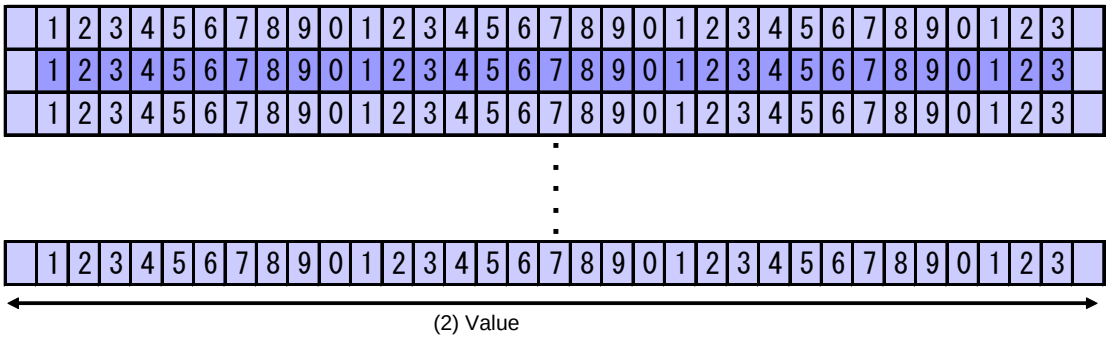
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.10.2 Complements

Screen Specifications

Screen Images





7.3.11 SPCommand (GNXSPCommand); S Display Part

S display part can display the spindle modal (S) and the value of actual spindle rotation speed.

To create a S display part, select [SPCommand] from the [Control] menu or select the following icon in NC Designer.

7.3.11.1 Property Settings

The property settings of the SPCommand are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Axis designation	:	Specify the spindle No. to be displayed.
Number of displays	:	Specify the number of the S commands displayed.
Character attribute	:	Specify the character attribute of captions.
Label	:	Specify the display of the label.
Value	:	Specify the display of the value.
Display	:	Specify whether to activate the three columns display and the number of digits to be displayed.
Callback function	:	Specify whether or not the callback functions are provided.

Axis designation

Item	Description
AxisFlag	Specify the spindle No. to be displayed. - When "0" is set, axes will be displayed in ascending order. - When the number of S display is 1, set one of 1/2/4/8/16/32, then one of 1st spindle(1) to the 6th spindle(32) will be displayed in the spindle No to be displayed. - When the number of S display is more than 1, the spindle No. to be displayed will be decided by a combination of setting values. Set "12" to display the 3rd spindle (4) and the 4th spindle (8). The smaller spindle No. will be displayed first when several setting values are combined. - Even when the spindle No. becomes larger than the number of S display due to combining setting values, the number of spindles to be displayed will not exceed the value designated to the number of S display. When a value out of the setting range is set, it is regarded as the default value (0). - When the No. of a spindle which is not mounted is set, the 1st spindle will be displayed.

Number of Displays

Item	Description
S_Number	Specify the number of the S commands displayed. (1 to 4)

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal Normal font Middle Middle-sized font Big Font with double height and width

Label

Item	Description
LabelForeColor	Specify the character color of the label.
LabelBackColor	Specify the background color of the label.

Value

Item	Description
ValueForeColor	Specify the character color of the value.
ValueBackColor	Specify the background color of the value.

Display

Item	Description
Line_Number	Specify the number of digits to be displayed.
ThreeSequencesType	Specify whether to activate the three columns display.

For the other properties, refer to "7.1 Common Functions of Controls".

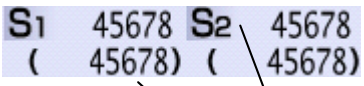
7.3.11.2 Complements

Screen Specifications

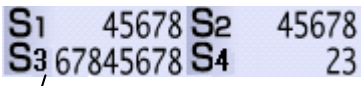
If two or less spindles are set, both the spindle modal and the actual spindle rotation speed (in parentheses) are displayed.
If three or more spindles are set, only the spindle modal is displayed.

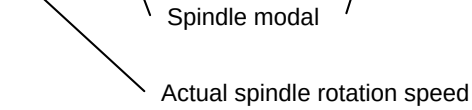
Screen Images

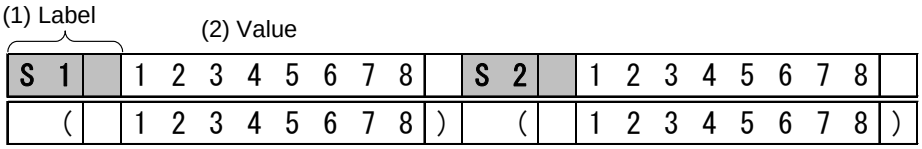
When two spindles are set



When four spindles are set







Related NC Parameters

The following table shows the related parameters.

No.	Name	Details	Setting range
1039	spinno Number of spindles	Specify the existence of a spindle	0 to 4

7.3.12 PLC Button Object (GNCPLCButton)

The PLC button control enables to read and write data from/to the PLC bit device of NC. It also enables to change the ON/OFF state of the button according to the state of the bit device.

To create a PLC button, select [PLC button] from the [Control] menu or select the following icon in NC Designer.

7.3.12.1 Property Settings

The property settings of the PLC button are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entries are accepted (permission) or rejected (prohibition).
NC Number	:	Specify the No. of NC to connect.
Ground	:	Specify the foreground and background.
Button type	:	Select the button action.
Display type	:	Specify the display type of the button.
PLC device	:	Specify the target PLC device.
Color/Pattern	:	Specify the color and pattern of the control.
Image	:	Specify the image of the control.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of captions.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Set the blink of the caption character string.
Update condition		Specify the update condition for drawing.
Callback function	:	Specify whether or not the callback functions are provided.

NC Number

Item	Description
NC_Number	Specify the No. of NC to connect. (Usually set to “1”.)

Ground

Item	Description
Ground	Specify the foreground and background. (Usually set to “0”.)

Button Type

Item		Description
Button type		Select the button action among the following three types.
	Momentary	The button turns ON when pressed, OFF when released.
	Alternate	The button alternates ON and OFF each time it is pressed.
	None	The button does not turn ON nor OFF when pressed.

Display type

Item	Description
Display type	Select the button type among the following three types.
Square	Rectangular button. The button is indicated in the designated color and pattern. 
Circle	Round button. The button is indicated in the designated color and pattern. 
Image	The button is indicated with the designated image resource. 

PLC Device

Item	Description
PLC device 1	Specify the address of the PLC bit device for the read or write operation.
Action of PLC device 1	Specify the operation to the PLC bit device specified in “PLC device 1“. (Read or write)
PLC device 2	Specify the address of the PLC bit device for reading operation.
Action of PLC device 2	Specify the operation to the PLC bit device specified in “PLC device 2“. (Read)

Color/Pattern

Item	Description
Pattern at the time of ON*1	Specify the pattern of the ON button.
Foreground color at the time of ON[*1]	Specify the foreground color of the ON button.
Background color at the time of ON[*1]	Specify the background color of the ON button.
Design at the time of ON[*2]	Specify the image of the ON button.
Character sequence at the time of ON	Specify the character string of the ON button.
Pattern at the time of OFF[*1]	Specify the pattern of the OFF button.
Foreground color at the time of OFF[*1]	Specify the foreground color of the OFF button.
Background color at the time of OFF[*1]	Specify the background color of the OFF button.
Design at the time of OFF[*2]	Specify the image of the OFF button.
Character sequence at the time of OFF	Specify the image of the OFF button.
Patten at the time of ON ON	Specify the pattern of the ONON button.
Foreground color at the time of ON ON	Specify the foreground color of the ONON button.
Background color at the time of ON ON	Specify the background color of the ONON button.
Design at the time of ON ON	Specify the image of the ONON button.
Character sequence at the time of ON ON	Specify the character string of the ONON button.
Pattern at the time of ON OFF	Specify the pattern of the ONOFF button.
Foreground color at the time of ON OFF	Specify the foreground color of the ONOFF button.
Background color at the time of ON OFF	Specify the background color of the ONOFF button.
Design at the time of ON OFF	Specify the image of the ONOFF button.
Character sequence at the time of ON OFF	Specify the character string of the ONOFF button.

*1: This setting is valid when the “Display type” is set to “Square or “Circle”.

*2: This setting is valid when the “Display type” is set to “Image”.

Image

Item	Description
Effect at the time of focus	Specify whether the color of the button changes or not when the focus is located. Select between "change color" and "no change."
Pattern at the time of focus[*1]	Specify the pattern of the button when the focus is located.
Foreground color at the time of focus[*1]	Specify the foreground color of the button when the focus is located.
Background color at the time of focus[*1]	Specify the background color of the button when the focus is located.
Design at the time of focus[*2]	Specify the image of the button when the focus is located.
Pattern at the time of disable[*1]	Specify the pattern of the button when the entry is disabled.
Foreground color at the time of disable[*1]	Specify the foreground color of the button when the entry is disabled.
Background color at the time of disable[*1]	Specify the background color of the button when the entry is disabled.
Design at the time of disable[*2]	Specify the image of the button when the entry is disabled.

*1: This setting is valid when the “Display type” is set to “Square or “Circle”.

*2: This setting is valid when the “Display type” is set to “Image”.

Update condition

Item	Description
Update condition	Select the update condition between "Always" and "At change". "Always" updates the drawing in each cycle. "At change" updates the drawing only when the display is changed.

Callback function

Item	Description
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

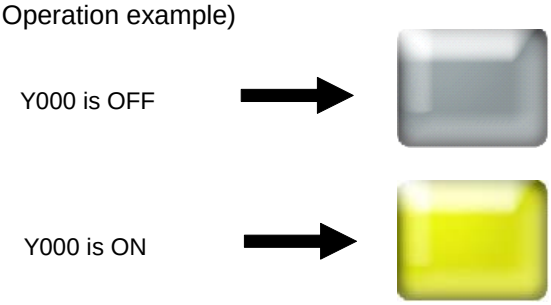
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.12.2 Complements

PLC Device Read Function

The button can correspond to the state of PLC bit device in the NC and can display the ON/OFF state according to the property settings.

Operation example)

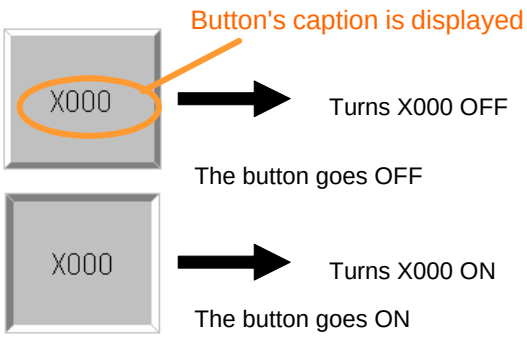


*4 types of design can be selected for the display according to the two addresses specified.

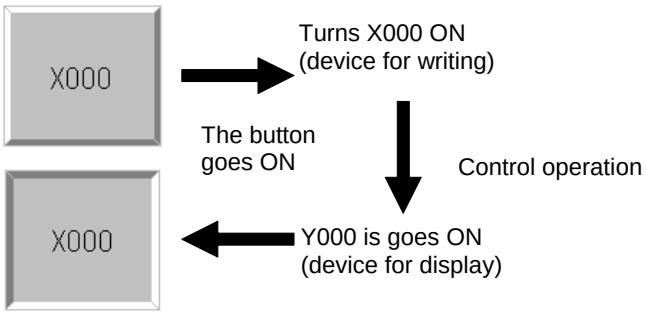
PLC Device Write Function

The button can write the ON/OFF state into the bit device in the NC specified in the property settings, at the time the button is pressed and its state is changed.
The button has “Momentary”(the button is ON as long as pressed) and “Alternate”(the button alternates ON/OFF when pressed) operation types.

Example of a single operation)



Example of a compounded operation)



Functional Specifications

Settings of PLC Bit Devices

Up to two PLC devices can be set for the read and write operations.
The PLC device 1 can be set to “None/Read/Write”, while PLC device 2 can be set to “None/Read”.
The PLC device to set should be a bit device.

Item	Specifications	Read	Write
PLC device 1	Reads or writes data from/into the PLC bit device	○	○
PLC device 2	Reads data from the PLC bit device	○	×

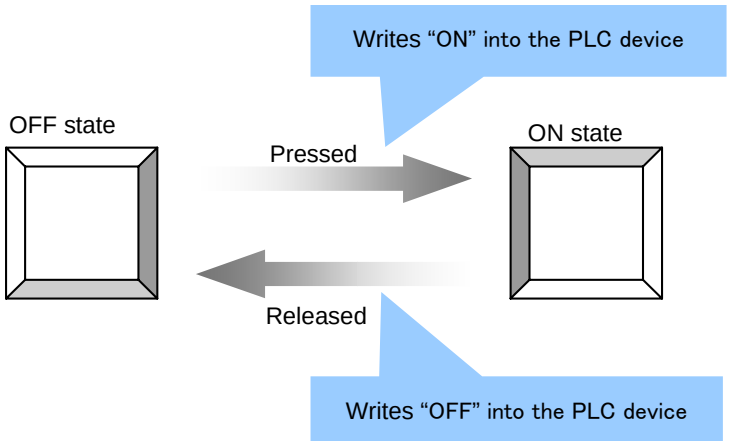
Button Action Type

There are the following three types of the button’s action when pressed.

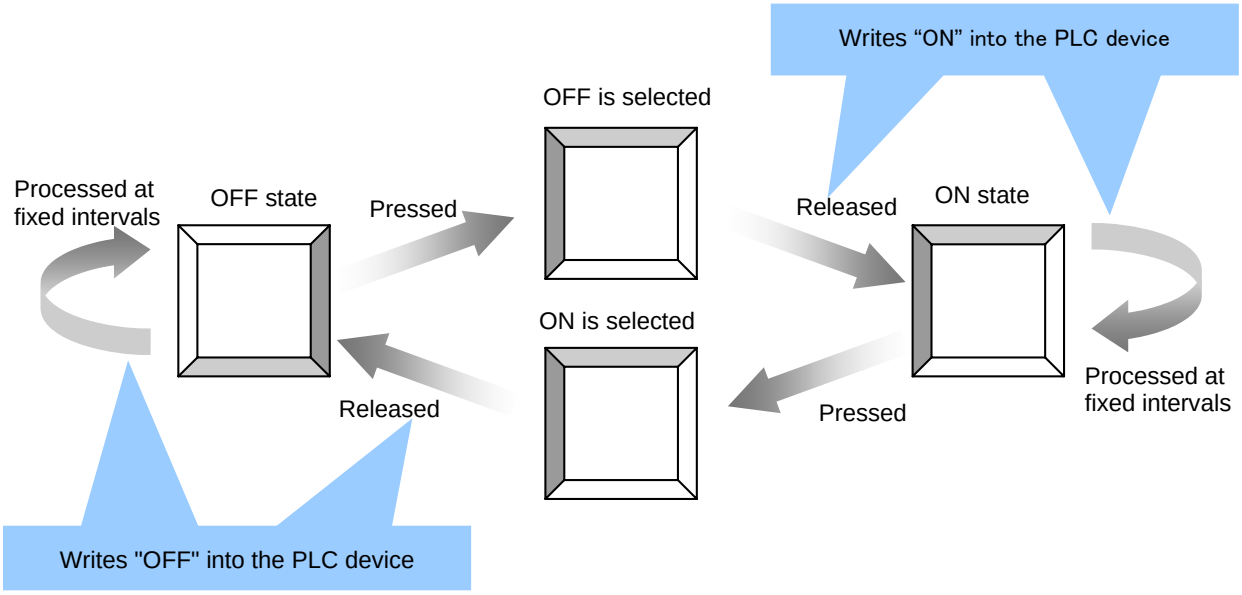
Button action	Writing “ON” into the PLC device	Writing “OFF” into the PLC device
Momentary (The button is ON as long as pressed)	Executed when the button is pressed	Executed when the button is released
	Executed when the ENTER key is pressed while the focus is located at the button.	Executed when the ENTER key is released while the focus is located at the button.
Alternate (The button alternates ON/OFF when pressed)	Executed when the OFF-state button is pressed and released.	Executed when the ON-state button is pressed and released.
	Executed when the ENTER key is pressed and released while the focus is located at the OFF-state button.	Executed when the ENTER key is pressed and released while the focus is located at the ON-state button.
	Executed at the fixed interval while the button is ON.	Executed at the fixed interval while the button is OFF.
None (The button does not turn ON/OFF when pressed)	Not executed	Not executed

“Momentary” and “Alternate” have each different time to write data into the PLC device.

(1) Timing of the “Momentary” write



(2) Timing of the “Alternate” write



Display Design

The display design of the button can be changed according to its ON/OFF state. The conditions of the display depend on how many PLC devices are set to “read”. The following table shows the display design of the button according to the number of devices for “read”.

Item	Number of PLC devices for “read”	Specifications
Pattern at the time of ON Foreground color at the time of ON Background color at the time of ON Design at the time of ON Character sequence at the time of ON	0	Displayed while the button is ON
	1	Displayed while the PLC device 1 (or 2) is ON
	2	Displayed when the PLC device 1 is ON and the PLC device 2 is OFF
Pattern at the time of OFF Foreground color at the time of OFF Background color at the time of OFF Design at the time of OFF Character sequence at the time of OFF	0	Displayed while the button is OFF
	1	Displayed while the PLC device 1 (or 2) is OFF
	2	Displayed when the PLC device 1 and 2 are both OFF
Patten at the time of focus Foreground color at the time of focus Background color at the time of focus Design at the time of focus	Independent from the number	Displayed when the focus is located at the button and “change color” is set for the "Effect at the time of focus”
Pattern at the time of disable Foreground color at the time of disable Background color at the time of disable Design at the time of disable Character sequence <A> at the time of OFF	Independent from the number	Displayed when the button is disabled
Pattern at the time of ON OFF Foreground at the time of ON OFF Background at the time of ON OFF Design at the time of ON OFF Character string at the time of ON OFF	2	Displayed when the PLC device 1 is OFF and the PLC device 2 is ON
Patten at the time of ON ON Foreground at the time of ONON Background at the time of ON ON Design at the time of t ON ON Character string at the time of ON ON	2	Displayed when the PLC device 1 and 2 are both ON

Ex.)

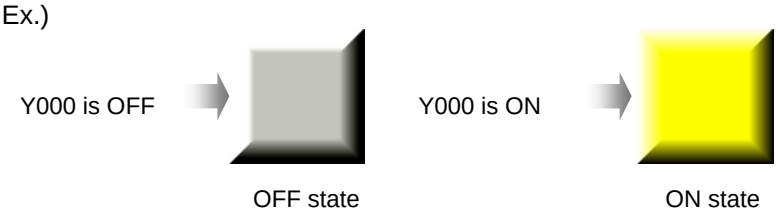


Figure 1. Display of the button when one PLC device is set to “read”

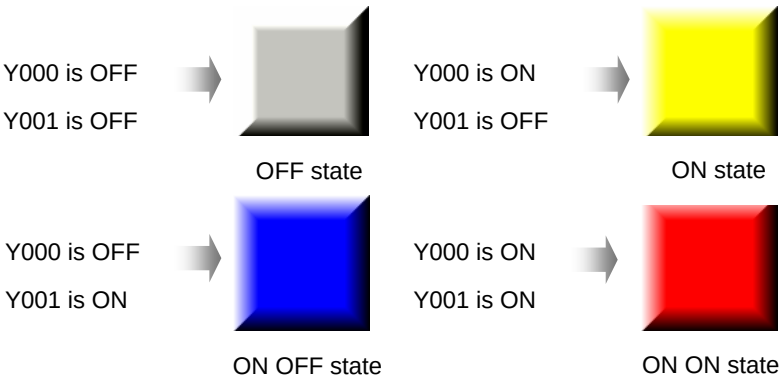


Figure 2. Display of the button when two PLC devices are set to “read”

7.3.13 PLC Text Box Object (GNCPLCTextBox)

The PLC text box control enables to read and write data from/to the PLC device of NC.

To create a PLC textbox, select [PLC textbox object] from the [Control] menu or select the following icon in NC Designer.

7.3.13.1 Property Settings

The property settings of the PLC text box are divided into the followings.

Control name	:	Specify the control name.
Position/Size	:	Specify the position and the size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entries are accepted (permission) or rejected (prohibition).
NC Number	:	Specify the No. of NC to connect.
Ground	:	Specify the foreground and background.
Color/Pattern	:	Specify the color and pattern of the control.
PLC device	:	Specify the target PLC device.
Display type/ Display format	:	Specify the format of the character string displayed on the control.
Password	:	Specify the password.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of captions.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Set the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.

NC Number

Item	Description
NC_Number	Specify the No. of NC to connect. (Usually set to “1”.)

Ground

Item	Description
Ground	Specify the foreground and background. (Usually set to “0”.)

Color/Pattern

Item	Description
Existence of a background color	Select if the background color is provided or not. If "None" is selected, the background is transparent.
Background color	Specify the background color.
Background color at the time of focus	Specify the background color when the focus is located.
Background color at the time of disable	Specify the background color when the entry is disabled.

PLC Device

Item	Description
PLC device	Specify the address of the PLC word device for the read or write operation.
Size	Specify the PLC device size used for reading or writing. (2 or 4 byte)
Type	Specify the format to display the PLC device value. (Signed decimal integer/ Unsigned decimal integer/ Hexadecimal integer/ Real number)
Magnification	Specify the magnification to the PLC device value to display.
Offset	Specify the offset value added to the PLC device value to display.

Display Type/Display Format

Item	Description
Number of the maximum characters	Specify the maximum number of characters to display. (1 to 256)
Maximum check	For the value field, select whether to check for the maximum value limit or not.
Maximum	Specify the maximum value for the maximum value check. (-2147483648 to 4294967295)
Minimum check	For the value field, select whether to check for the minimum value limit or not.
Minimum	Specify the minimum value for the minimum value check. (-2147483648 to 4294967295)
Comma	For the value field, select whether to display commas or not.
The number of integer part digits	Specify the number of digits in integer part of the value. (1 to 12)
The number of decimal part digits	Specify the number of digits in decimal part of the value. (1 to 12)
Zero suppress	Select whether or not to display 0s in the blank digits when the value does not have as many digits as specified.

Password

Item	Description
Password setup	Select "Yes" to display entered characters with asterisks (*). 

Character Attribute

Item	Description
Effect at the time of focus	To display the cursor in the text box where the focus is located, select "With cursor." To display no cursor, select "No effect." To select all characters, select "Selected."

Callback Function

Item	Description
OnChangeString	Select "Yes" to add a process to be executed after the caption character string has been changed.
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

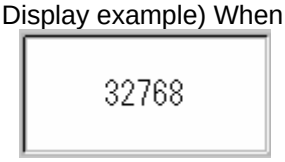
For the other properties, refer to "7.1 Common Functions of Controls".

7.3.13.2 Complements

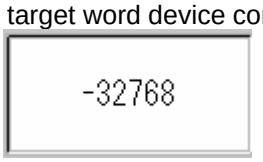
PLC Device Read Function

This function allows the text box to display the value in the PLC word device of NC.
The properties include “Display format (signed integer, unsigned integer, real number or hexadecimal integer)”, “The number of integer part digits”, “The number of decimal part digits”, “Magnification”, “Offset” and “Zero suppress”. They change the display as follows.

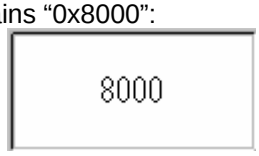
Display example) When the target word device contains “0x8000”:



Display format:
Unsigned integer



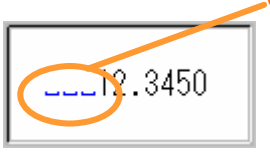
Display format:
Signed integer



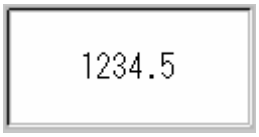
Display format:
Hexadecimal integer

Display example) When the target word device contains the value corresponding to “12.345”:

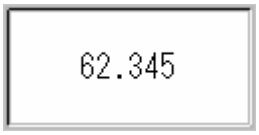
Display format:
Real number



Display format:
Real number
Integer part: 5 digits
Decimal part: 4 digits



Display format:
Real number
Magnification: 100



Display format:
Real number
Offset: 50

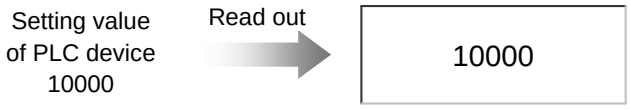
PLC Device Write Function

This function allows the value in the text box to be written into the PLC word device of NC, which is specified in the property settings, by pressing INPUT.

Functional Specifications

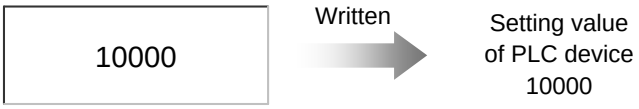
Reading the Value from the PLC Device

PLC device read function allows the value in the specified PLC device to be read at fixed intervals and displayed in the specified numeral form.



Writing the Value into the PLC Device

PLC device write function allows the value to be written into the PLC device at the time the value has been input and confirmed. The value is “confirmed” when the user presses the ENTER key while inputting the value and then the value is recognized to be within the setting range of “Type".



Inputting the Setting Value to be Written into the PLC Device

The followings are the operations required to write the setting value into the PLC device.

(1) Inputting the value

The operation is as same as that of the usual text box control: display the character strings by inputting with keys when the focus is located. Any change of the PLC value does not change the display on the PLC text box control while the value is being input.

(2) Confirming the value

Pressing the ENTER key writes the value into the PLC device. Pressing the ENTER key does not move the focus. The focus is still located at the control.

Display format

Set the display by specifying the type, size, the number of integer part digits, the number of decimal part digits, magnification, offset and comma.

(1) Type

Type	Display range	
Signed decimal integer	2 byte: -32768 to 32767	4 byte: -2147483648 to 2147483647
Unsigned decimal integer	2 byte: 0 to 65535	4 byte: 0 to 4294957296
Hexadecimal integer	2 byte: 0 to FFFF	4 byte: 0 to FFFFFFFF
Real number	1.0E-32 to 1.0E32	

• Signed decimal integer (2 byte)

-32768
32767

• Unsigned decimal integer (2 byte)

0
65535

• Signed decimal integer (4 byte)

-2147483648
2147483647

• Unsigned decimal integer (4 byte)

0
4294967295

• Hexadecimal integer (2 byte)

0
FFFF

• Hexadecimal integer (4 byte)

0
FFFFFFFF

• Reall number

-XXXX.XX
XXXX.XX

(2) Size

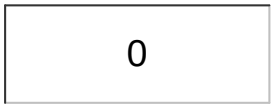
Select 2 or 4 byte for the PLC device size used for reading and writing. When the “actual number” is set for the display type, the device size is fixed to 4 byte.

(3) The number of integer part digits

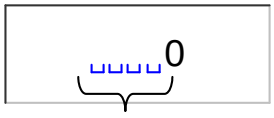
Specify the minimum number of digits displayed in integer part. If the value in integer part has the smaller number of digits than the minimum, spaces are output to the blank digits. Spaces are not output if the value in integer part has the larger number of digits than the minimum.

Display example)

• 1 digit for integer part



• 5 digits for integer part



Spaces in 4 digits

• 5 digits for integer part

(4) The number of decimal part digits

Specify the number of digits in decimal part to be displayed when the “actual number” is set for the display type. .If the decimal part of the value has the smaller number of digits than specified, “0”s are output to the blank digits. The digit behind the specified digits in decimal part is rounded off to the nearest value.

Display example) Displaying “23.45” in actual number

• 0 digit behind decimal

• 1 digit behind decimal

• 3 digits behind decimal

The value is rounded off as follows.

Positive value

0<X<0.5 -> Round-down (to 0)

0.5≤X<1 -> Round-up (to 1)

Negative value

0>X>-0.5 -> Round-down (to 0)

-0.5≥X>-1 -> Round-up (to -1)

(5) Magnification

Specify the magnification to the value, which is read from the PLC device, to display. The specified magnification becomes the divisor for the value, after reduced by the offset amount, to be written into the device. The magnification is available for all types. However, if the value after the magnification exceeds the available number of digits, the display will not be exact.

(6) Offset

Specify the offset value to add to the value, which is read from the PLC device, to display. The magnification, when specified, is carried out to the PLC device value before the offset value is added to. When written into the device, the input value is reduced by the offset value. The offset adjustment is available for all types. However, if the value exceeds the available number of digits after the offset adjustment, the display will not be exact.

(7) Comma

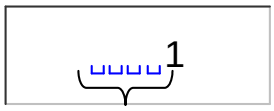
Setting the display with commas is available if the type is set to “decimal integer”. A comma is inserted after every three digits, if “comma” is set to the display.

(8) Zero suppress

Zero suppress is used to display “0”s in the blank digits of integer part when the value does not have as many digits as specified. Setting “Yes” outputs spaces to the blank digits in integer part when the value to be displayed does not have as many digits as specified. Setting “No” outputs “0”s to the blank digits of integer part.

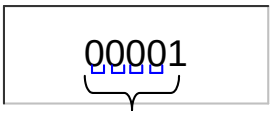
Display example) Display the value “1” in integer part with 5 digits

• Zero suppress is set



Spaces in 4 digits

• Zero suppress is not set



"0"s in 4 digits

Magnification and Offset

The followings are the process of the magnification and offset adjustment when the value is read and written.

(1) Process to read

The PLC device value, when read, is displayed after the following process:

1) Magnification

2) Offset adjustment

(2) Process to write

The input value is written into the device after the following process:

1) Offset adjustment

2) Magnification

If the value to be written into is not integer, the digit after the decimal point is rounded down.

7.3.14 NC Data Textbox (GNCDataTextBox)

The NC data text box is a text box part that enables reading and writing from and to the NC's internal data.

To create an NC data text box, select [NC data text box] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.14.1 Property Settings

The property settings for the NC data text box are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and the size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).
NC number	:	Specify the No. of NC to connect.
Part system number	:	Specify the No. of part system.
Ground	:	Specify the foreground and background.
Color/pattern	:	Specify the color and pattern of the control.
NC data	:	Specify the target NC data.
Display type/ display format	:	Specify the format of the character string displayed on the control.
Update cycle	:	Specify the update cycle of the NC data display.
Character attribute	:	Specify the character attribute of captions.
Solid frame	:	Specify the solid frame of the control.
Callback function	:	Specify whether or not the callback functions are provided.

NC number

Item	Description
NC number	Specify the No. of NC to connect (Usually set to "1".)

Part system number

Item	Description
Part system number	Specify the No. of the part system to which the NC data to read/write belongs (1 to 10).

Ground

Item	Description
Ground	Specify the foreground and background (Usually set to "0").

Color/Pattern

Item	Description
Existence of a background color	Select if the background color is provided or not. If "None" is selected, the background is transparent.
Background color	Specify the background color.
Background color at the time of focus	Specify the background color when the focus is located.
Background color at the time of disable	Specify the background color when the entry is disabled.

NC Data

Item	Description
Axis number	Specify the No. of the axis to which the NC data to read/write belongs (0 to 16).
AxisCross	Specify the basic part system (0) and the current part system during cross control (1) of the NC data to be read or written.
Number of Section	Set the section No. of the NC data to be read or write (0 to 999).
Number of Sub-section	Set the sub-section No. of the NC data to be read or write (0 to 1000000000).
Data Type	Specify the type of the NC data to read/write. (1-byte integer, 2-byte integer, 4-byte integer, 8-byte real and Character string types)
Type	Specify the format to display the NC data value (Signed decimal integer/Unsigned decimal integer/Binary integer/Hexadecimal integer/Real number/Character string).
Magnification	Specify the magnification to the NC data value to display.
Offset	Specify the offset value to be added to the NC data value to display.

Display Type/Display Format

Item	Description
Number of the maximum characters	Specify the maximum number of characters to display. (1 to 256)
Comma	For the value field, select whether to display commas or not.
Half-size number	For entry in the text box, select whether one-byte numbers are allowed or not.
Half-size English small letter	For entry in the text box, select whether one-byte lower case letters are allowed or not.
Half-size English capital letter	For entry in the text box, select whether one-byte upper case letters are allowed or not.
Half-size sign	For entry in the text box, select whether one-byte symbols are allowed or not.
The number of integer part digits	Specify the number of digits in integer part of the value, when the NC data is the real number type or decimal integer type (1 to 12).
The number of decimal part digits	Specify the number of digits in decimal part of the value, when the NC data is the real number type (1 to 10).
Exponential notation	Select whether or not to display exponential notation, when the NC data is the real number type.
Zero suppress	Select whether or not to display 0s in the blank digits when the value does not have as many digits as specified.

Update Cycle

Item	Description
RefreshFrequency	Specify the number of times to thin out the timer event processes (0 to 100). When set to "0", the display is not updated at the timer event.
RefreshTiming	The display is refreshed when the counted number of "RefreshFrequency" reaches the number of counts specified with "RefreshTiming" (0 to 99).

Character Attribute

Item	Description
Effect at the time of focus	To display the cursor in the text box where the focus is located, select "With cursor." To display no cursor, select "No effect." To select all characters, select "Selected.

Callback Function

Item	Description
OnChangeString	Select "Yes" to add a process to be executed after the caption character string has been changed.
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.14.2 Complements

NC Data Read Function

This function allows the text box to display the value the value set in the internal data of NC.
The properties include "Type (signed decimal integer, unsigned decimal integer, binary integer, hexadecimal integer, real number and character string)", "The number of integer part digits", "The number of decimal part digits", "Magnification", "Offset", "Exponential notation", "Comma" and "Zero suppress". They change the display as follows.

Display example) When the NC data contains 0xABCD (short type)

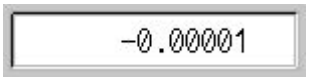
Type: Binary integer

Type : Hexadecimal integer

Type: Unsigned decimal integer

Type: Signed decimal integer

Display example) When the NC data (common variable #500) contains -0.00001 (real number type).



Type: Real number
Number of integer part digits: 6
Number of decimal part digits: 5
No exponential notation

Type: Character string

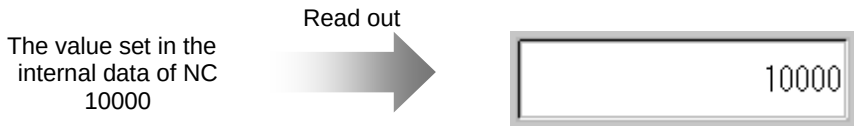
NC Data Write Function

This function allows the value set in the text box to be written to the NC, which is specified in the property settings, by pressing INPUT.

Functional Specifications

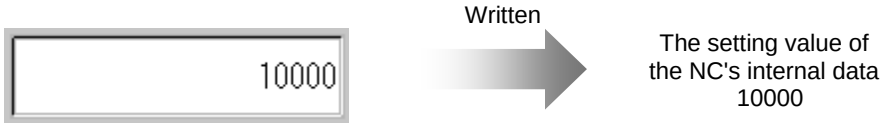
Reading the Value from NC

NC data read function allows the value in the NC data to be read at fixed intervals and displayed in the specified format.



Writing the Value to NC

NC data write function allows the value to be written to the NC at the time the value has been input and confirmed. The value is "confirmed" when the user presses the ENTER key while inputting the value and then the value is recognized to be within the setting range of "Type".



Inputting the Setting Value to be Written to NC

The followings are the operations required to write the setting value to the NC.

(1) Inputting the value

The operation is the same as of the usual text box control: display the character strings by inputting with keys when the focus is located. Any change of the NC's internal data value does not change the display on the NC data text box control while the value is being input.

(2) Confirming the value

Pressing the ENTER key writes the value into the NC. Pressing the ENTER key does not move the focus. The focus is still located at the control.

Display Format

Set the display format by specifying the type, data type, the number of integer part digits, the number of decimal part digits, magnification, offset, exponential notation, comma and zero suppress.

(1) Type and data type

The display format is determined as shown below according to the combination of the type and data type. Note that if the character string type is selected for the data type, the NC is notified of the number of characters (the maximum number of characters + the number of NULL characters). If the value read from the NC has a larger number of characters than the maximum, it is left blank.

	Data type				
Type	1-byte integer	2-byte integer	4-byte integer	8-byte real	Character string
Binary integer	0 or 1 (8bit)	0 or 1 (16bit)	0 or 1 (32bit)		
Hexadecimal integer	00 to FF	0000 to FFFF	00000000 to FFFFFFFF		
Unsigned decimal integer	0 to 255	0 to 65535	0 to 4294967295	Same as on the left	
Signed decimal integer	-128 to 127	-32768 to 32767	-2147483648 to 2147483647	Same as on the left	
Real number	-128.0 to 127.0	-32768.0 to 32767.0	-2147483648.0 to 2147483647.0	-999999999999.9999999999 to 999999999999.9999999999 (At the exponential notation, the minimum setting unit is 99 digits after the decimal point (1.0E-099).)	-999999999.0 to 999999999.0 (The minimum setting unit is 99 digits after the decimal point (1.0E-099).)
Character string					Character string

* To display in the same format as the standard screen, select the "Character string" type for Data type, and "Character string" for Type.
* Select "Real number" for Type to input an exponential notation value.

(2) Number of integer part digits/number of decimal part digits

When the type is real number, a real number is displayed with the specified number of integer part digits and the specified number of decimal part digits. When "Yes" is set for the exponential notation, the exponential notation display is enabled.
When "No" is selected, the value after the specified number of decimal part digits is rounded off. However, if the number of the integer part digits of the value read from the NC exceeds the set number of digits, the data displayed is all "*".
When "Yes" is set for the exponential notation, and when the value read from the NC is smaller than 1 and is exceeding the specified number of decimal part digits, or when the value is exceeding the specified number of integer part digits, exponential notation is carried out. When a value is displayed with exponential notation, the value is displayed by rounding off, depending on the number of decimal part digits.
If the value read from the NC is within the display range, but is exceeding the maximum number of characters, it is left blank.

When a data in the real number type is displayed, what is displayed is as below (Example: common variable).

	Data type		
Setting value	Real number type (6.4 digits/no exponential notation)	Real number type (6.4 digits/exponential notation)	Character string type
0.00001	0.0000	1.0000E-05	1.0000E-005
0.00005	0.0000	5.0000E-05	5.0000E-005
0.00015	0.0001	0.0002	0.0002
0.00045	0.0004	0.0004	0.0004
0.00046	0.0004	0.0005	0.0005
0.00054	0.0005	0.0005	0.0005
0.00055	0.0005	0.0006	0.0006
123456.00015	123456.0001	123456.0002	123456.0002
1234567.00000	**********	1.2346E+06	1.2346E+006

When the type is decimal integer, the number of digits specified with the number of integer part digits is displayed. However, if the value read from the NC exceeds the number of integer part digits, the data displayed is all "*".

(3) Magnification and offset

When the type is decimal integer or real number, "Magnification" and "Offset" can be reflected on the displayed data. When data is read out from the NC, the magnification is applied to the NC's internal data, and then the offset is added.
When data is written to the NC, the value obtained by subtracting the offset value from the entered value and then divided it with the magnification is written to the NC.
If the result, in which the magnification and offset are reflected, is exceeding the available number of digits, it is not possible to display the correct value.

(4) Exponential notation

When the type is real number, exponential notation is carried out by selecting "Yes" for the exponential notation.
When "Yes" is selected for the exponential notation, and when the value read from the NC is smaller than 1 and is exceeding the specified number of decimal part digits, or when the value is exceeding the specified number of integer part digits, exponential notation is carried out. When a value is displayed with exponential notation, the value is displayed by rounding off, depending on the number of decimal part digits.
When "No" is selected for the exponential notation, the value after the number of decimal part digits is rounded down. Note that if the integer part of the value read from the NC exceeds the number of integer part digits, the data displayed is all "*".

(5) Comma

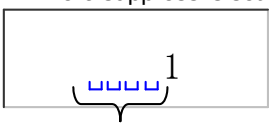
Setting the display with commas is available if the type is set to "decimal integer". A comma is inserted after every three digits, if "comma" is set to the display.

(6) Zero suppress

Zero suppress is used to display “0”s in the blank digits of integer part when the value does not have as
many digits as specified. Setting “Yes” outputs spaces to the blank digits in integer part when the value to
be displayed does not have as many digits as specified. Setting “No” outputs “0”s to the blank digits of
integer part.

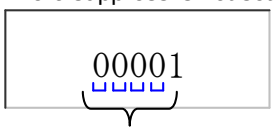
Display example) Display the value "1" in integer part with five digits

• Zero suppress is set



Spaces in four digits

• Zero suppress is not set



0s in four digits

7.3.15 PLC extension button (GNCPLCExButton)

PLC extension button is a control part that enables reading and writing from and to the PLC device in NC, and switching the ON/OFF state of a button in accordance with the device condition. This part is equivalent to a PLC button control, but is different from the PLC button in the following points.

- "Actuator" has been added to the button types.
- "Interlock", "Disable" and "Blink" have been added to PLC devices.
- The group designation function has been added.

To create a PLC extension button, select [PLC extension button] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.15.1 Property Settings

The property settings for the PLC extension button are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and the size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
Input permission	:	Select whether the entry is accepted (permission) or rejected (prohibition).
NC number	:	Specify the No. of NC to connect.
Ground	:	Specify the foreground and background.
Button type	:	Select the button action.
Display type	:	Select the display type of the button.
PLC device	:	Specify the target PLC device.
Color/pattern	:	Specify the color and pattern of the control.
Image	:	Specify the image of the control.
Caption	:	Specify the caption (character string) displayed on the control.
Character attribute	:	Specify the character attribute of captions.
Solid frame	:	Specify the solid frame of the control.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Set the blink of the caption character string.
Group	:	Specify the group number.
Callback function	:	Specify whether or not the callback functions are provided.

NC number

Item	Description
NC number	Specify the No. of NC to connect (Usually set to "1".)

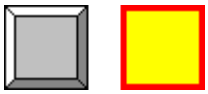
Ground

Item	Description
Ground	Specify the foreground and background (Usually set to "0").

Button Type

Item		Description
Button type		Select the button action among the following three types.
	Momentary (Type 1)	The button turns ON when it is pressed. It turns OFF when it is released.
	Momentary (Type 2)	The display action is different between Type 1 and Type 2.
	Alternate	The button alternates ON and OFF each time it is pressed.
	Actuator	The display action is different between Alternate and Actuator.
	None (Lamp)	The button does not turn ON or OFF even if it is pressed.

Display Type

Item		Description
Display type		Select the button type among the following three types.
	Square	Rectangular button. The button is indicated in the designated color and pattern. 
	Circle	Round button. The button is indicated in the designated color and pattern. 
	Image	The button is indicated with the designated image resource. 

PLC Device

Item	Description
PLC device 1	Specify the address of the PLC device for the read or write operation.
Action of PLC device 1	Specify the operation to the PLC device specified in “PLC device 1“ (Read, Write or None).
Bit position of PLC device 1	Specify the bit position of the word device when the word device (D/R) is set to PLC device 1 (0 to 15).
PLC device 2	Specify the address of the PLC device for the read or write operation.
Action of PLC device 2	Specify the operation to the PLC device specified in “PLC device 2“ (Read, Write or None).
Bit position of PLC device 2	Specify the bit position of the word device when the word device (D/R) is set to PLC device 2 (0 to 15).
Disable function	Specify whether to enable the function that controls the entry using a PLC device. When "Permission" is selected for Input permission, this setting is enabled.
Disable device	Specify the address of the PLC device for disabling the entry. This setting is enabled when the disable function is valid.
Bit position of disable device	Specify the bit position of the word device when the word device (D/R) is set to Disable device (0 to 15). This setting is enabled when the disable function is valid.
Blink device	Specify the address of the PLC bit/word device for controlling the blink display.
Bit position of blink device	Specify the bit position of the word device when the word device (D/R) is set to Blink device (0 to 15).
Interlock device 1	Specify the address of the PLC device for the interlock operation.
Action of interlock device 1	Specify the type of Interlock 1 (At the time of ON / At the time of OFF/No).
Bit position of interlock device 1	Specify the bit position of the word device when the word device (D/R) is set to Interlock device 1 (0 to 15).
Interlock device 2	Specify the address of the PLC device for the interlock operation.
Action of interlock device 2	Specify the type of Interlock 2 (At the time of ON / At the time of OFF/No).
Bit position of interlock device 2	Specify the bit position of the word device when the word device (D/R) is set to Interlock device 2 (0 to 15).
PLC device OFF	Select whether to write OFF(0) of the target PLC device at screen switching between "Yes" and "No". This setting is enabled when "Alternate" or "Actuator" is selected for the button type.
Automatic OFF time[*1]	Specify this time to automatically turn OFF the PLC device to be written after the specified time has passed since the button has been pressed (0 to 3600 seconds. 0 is invalid).
Delay ON time	Specify this time to turn ON (1) the PLC device for the writing operation by keeping pressing the button (0 to 60 seconds. 0 is invalid).

*1: If the button is kept pressed, the device is automatically turned OFF after the specified time has passed since the button has been pressed.

Color/Pattern

Item	Description
Pattern at the time of ON[*1]	Specify the pattern of the ON button.
Foreground color at the time of ON[*1]	Specify the foreground color of the ON button.
Background color at the time of ON[※1]	Specify the background color of the ON button.
Design at the time of ON[*2]	Specify the image of the ON button.
Pattern at the time of OFF[*1]	Specify the pattern of the OFF button.
Foreground at the time of OFF[*1]	Specify the foreground color of the OFF button.
Background at the time of OFF[*1]	Specify the background color of the OFF button.
Design at the time of OFF[*2]	Specify the image of the OFF button.
Pattern at the time of interlock[*1]	Specify the pattern in the interlock state.
Foreground color at the time of interlock[*1]	Specify the foreground color of the button in the interlock state.
Background color at the time of interlock[*1]	Specify the background color of the button in the interlock state.
Design at the time of interlock[*2]	Specify the image to display in the interlock state.

*1: This setting is valid if [Display Type] is "Square" or "Circle".
*2: The setting is valid if the [Display Type] is "Image."

Image

Item	Description
Effect at the time of focus	Specify whether the color of the button when the focus is located changes or not. Select between "change color" and "no change."
Pattern at the time of focus[*1]	Specify the pattern of the button when the focus is located.
Foreground color at the time of focus[*1]	Specify the foreground color of the button when the focus is located.
Background color at the time of focus[*1]	Specify the background color of the button when the focus is located.
Design at the time of focus[*2]	Specify the image of the button when the focus is located.
Pattern at the time of disable[*1]	Specify the pattern of the button when the entry is disabled.
Foreground color at the time of disable[*1]	Specify the foreground color of the button when the entry is disabled.
Background color at the time of disable[*1]	Specify the background color of the button when the entry is disabled.
Design at the time of disable[*2]	Specify the image of the button when the entry is disabled.

*1: This setting is valid if [Display Type] is "Square" or "Circle".
*2: The setting is valid if the [Display Type] is "Image."

Caption

Item	Description
Character string at the time of ON	Specify the character string to be displayed at the time of ON. The setting is valid if "provided" is selected for the caption existence.
Character string at the time of OFF	Specify the character string to be displayed at the time of OFF. The setting is valid if "provided" is selected for the caption existence.
Character string at the time of interlock	Specify the character string to be displayed in the interlock state. The setting is valid if "provided" is selected for the caption existence.

Character Attribute

Item	Description
Character color at the time of ON	Specify the character color to be displayed at the time of ON.
Character color at the time of OFF	Specify the character color to be displayed at the time of OFF.
Character color at the time of interlock	Specify the character color to be displayed in the interlock state.

These settings are valid if the [caption] is "provided."

Solid frame

Item	Description
Existence of a solid frame at the time of ON	Select the presence of the solid frame to be displayed at the time of ON between "Yes" and "None".
Solid frame at the time of ON	Select the ID of the solid frame resource to be displayed at the time of ON.
Existence of a solid frame at the time of OFF	Select the presence of the solid frame to be displayed at the time of OFF between "Yes" and "None".
Solid frame at the time of OFF	Select the ID of the solid frame resource to be displayed at the time of OFF.
Existence of a solid frame at the time of interlock	Select the presence of the solid frame to be displayed at the time of interlock between "Yes" and "None".
Solid frame at the time of interlock	Select the ID of the solid frame resource to be displayed at the time of interlock.
Existence of a solid frame at the time of disable	Select the presence of the solid frame to be displayed at the time of disable between "Yes" and "None".
Solid frame at the time of disable	Select the ID of the solid frame resource to be displayed at the time of disable.

Blink

Blink the caption character string of the control.

Item	Description
Blink	Select the existence of blink from "Yes", "No" or "Blink device".

Group

Item	Description
Group No.	Specify the group number to which the PLC extension button belongs (0 to 100. 0 is invalid). Only one PLC extension button among those belonging to the same group number is allowed to be active on a screen.

Callback Function

Item	Description
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.15.2 Complements

Functional Specifications

PLC Device Setting

Up to two PLC devices can be set for the read and write operations.
The combinations of the operation modes (Read/Write/None) are limited according to the specified button type.
If a combination other than below is selected, the operation is the same as when the input permission is set to "Prohibition".

Item	PLC device	Write	Read	None
Momentary (Type 1)	PLC device 1	○	×	○
	PLC device 2	×	○	×
Momentary (Type 2)	PLC device 1	○	×	×
	PLC device 2	×	○	×
Alternate	PLC device 1	○	×	×
	PLC device 2	○	○	×
Actuator	PLC device 1	○	×	×
	PLC device 2	×	○	×
None (Lamp)	PLC device 1	×	○	×
	PLC device 2	×	○	×

Priority of PLC devices

The image displayed by the PLC extension button changes in accordance with the states of PLC devices.
Each PLC device has a priority level, therefore when a PLC device with a higher priority is active, the state of a PLC device with a lower priority is not referred to.
The priority order of the PLC devices is determined as shown below.

Priority	PLC device
Higher ↑ ↓ Lower	Disable device
	Interlock device 1/Interlock device 2
	Blink device
	PLC device 1/PLC device 2

Button Action Type

(1) Momentary (Type 1)

While the button is kept pressed, ON(1) is written to the PLC device specified with PLC device 1 (Write). When the button is released, OFF(0) is written.

The button can correspond to the state of PLC device 2 (Read) and can display the ON/OFF state according to the property settings.

Display action changes as follows according to the ON/OFF of the PLC device 2.

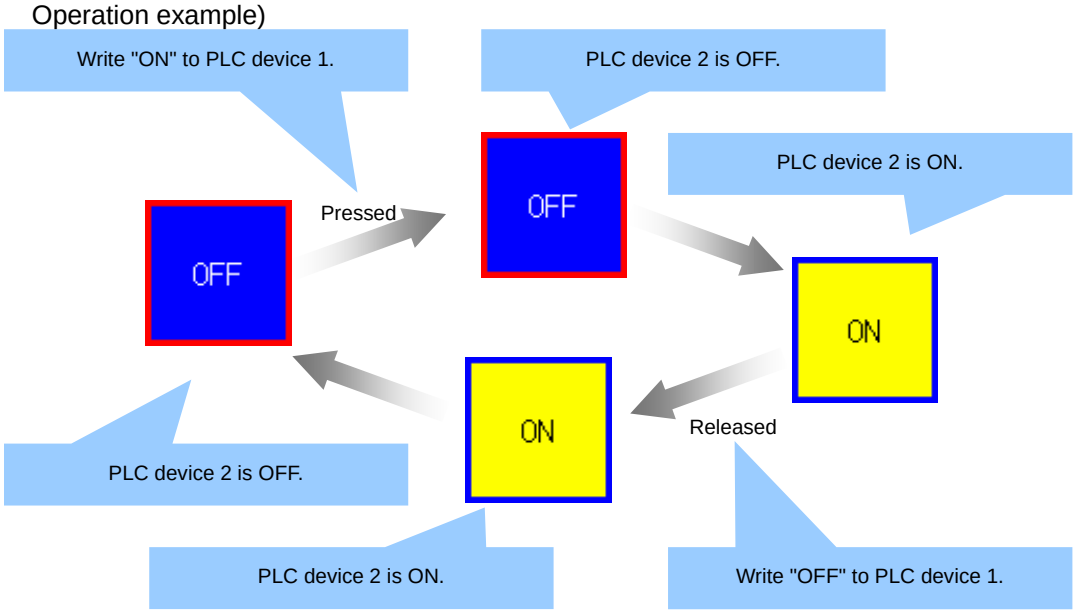
PLC device 2 (Read)	OFF(0)	ON(1)
Color/Pattern	At the time of OFF*1	At the time of ON*1
Caption/Character attribute	At the time of OFF*2	At the time of ON*2
Solid frame	At the time of OFF*3	At the time of ON*3

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.

*2: Caption/Character attribute refers to the character strings and character colors at the time of ON and OFF.

*3: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



NOTE

- When the action mode of PLC device 1 is set to "None", the states at the time of ON/OFF, which are set in the properties, can also be displayed in line with the state of PLC device 2 (Read).
- Blink display cannot be controlled with the blink device.
- The group No. setting is invalid.

(2) Momentary (Type 2)

While the button is kept pressed, ON(1) is written to the PLC device specified with PLC device 1 (Write). When the button is released, OFF(0) is written.

The button can correspond to the combination of PLC device 1 (Write) and PLC device 2 (Read) and can display the ON/OFF state according to the property settings.

Display action changes as follows according to the ON/OFF of PLC device 1 and 2.

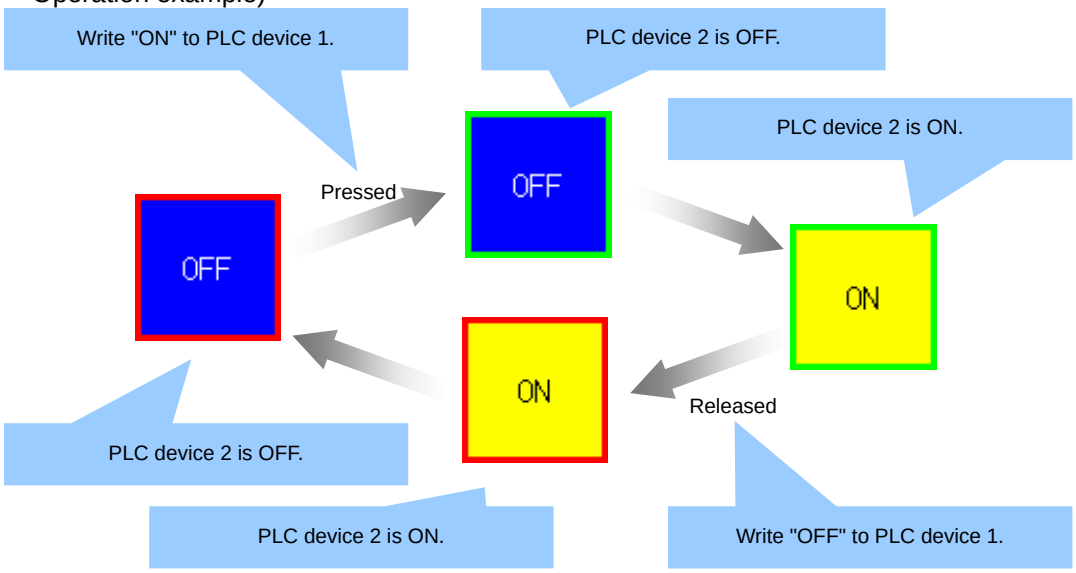
PLC device 1 (Write)	**OFF(0)**		**ON(1)**	
PLC device 2 (Read)	**OFF(0)**	**ON(1)**	**OFF(0)**	**ON(1)**
Color/Pattern	At the time of OFF*1	At the time of ON*1	At the time of OFF*1	At the time of ON*1
Caption/Character attribute	At the time of OFF*2	At the time of ON*2	At the time of OFF*2	At the time of ON*2
Solid frame	At the time of OFF*3		At the time of ON*3	
Blink device	Enable	Disable	Disable	

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.

*2: Caption/Character attribute refers to the character strings and character colors at the time of ON and OFF.

*3: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



NOTE

- Only when both PLC device 1 and PLC device 2 are OFF, blink display can be controlled with the blink device.
- The group No. setting is invalid.

(3) Alternate

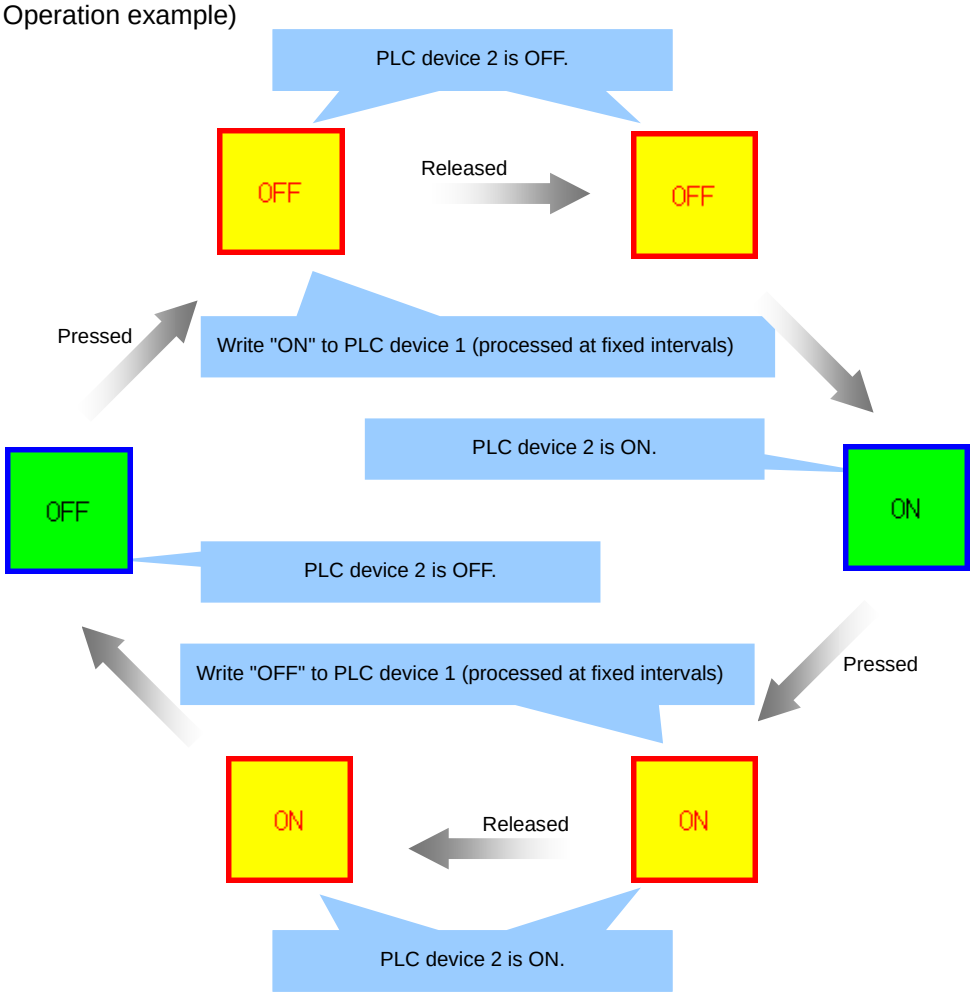
Each time the button is pressed, ON(1)/OFF(0) of PLC device 1 (Write) is written alternately.
When PLC device 1 (Write) is OFF, the button can correspond to the state of PLC device 2 (Read) and can display the ON/OFF state according to the property settings.
When PLC device 1 (Write) is ON, the display action is reversed to the above-mentioned operation, except for a character string.

Display action changes as follows according to the ON/OFF of PLC device 1 and 2.

PLC device 1 (Write)	**OFF(0)**		**ON(1)**	
PLC device 2 (Read)	**OFF(0)**	**ON(1)**	**OFF(0)**	**ON(1)**
Color/Pattern	At the time of OFF *1	At the time of ON *1	At the time of ON *1	At the time of OFF *1
Character attribute (character color)	At the time of OFF	At the time of ON	At the time of ON	At the time of OFF
Solid frame	At the time of OFF *2	At the time of ON *2	At the time of ON *2	At the time of OFF *2
Caption (character string)	At the time of OFF	At the time of ON	At the time of OFF	At the time of ON
Blink device	Enable	Disable	Disable	

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.
*2: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



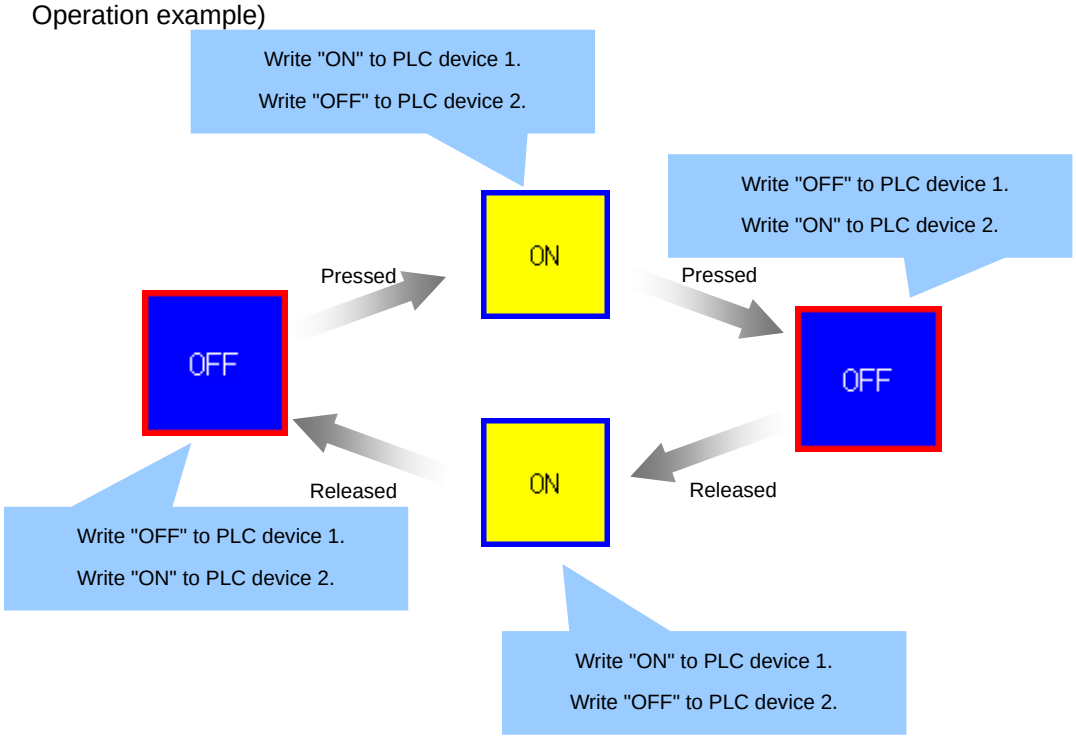
With the alternating button, the action mode of PLC device 2 can be set to "Write".
When the action mode of PLC device 2 is set to "Write", ON(1)/OFF(0) of PLC device 2 (Write) is written alternately each time the button is pressed.
The button can correspond to the state of PLC device 1 (Write) and can display the ON/OFF state according to the property settings.

Display action changes as follows according to the ON/OFF of PLC device 1.

PLC device 1 (Write)	OFF(0)	ON(1)
Color/Pattern	At the time of OFF*1	At the time of ON*1
Caption/Character attribute	At the time of OFF*2	At the time of ON*2
Solid frame	At the time of OFF*3	At the time of ON*3

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.
*2: Caption/Character attribute refers to the character strings and character colors at the time of ON and OFF.
*3: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



(4) Actuator

Each time the button is pressed, ON(1)/OFF(0) of PLC device 1 (Write) is written alternately. The solid frame corresponds to the state of PLC device 1 (Write) and the items other than the solid frame correspond to the state of PLC device 2 (Read).

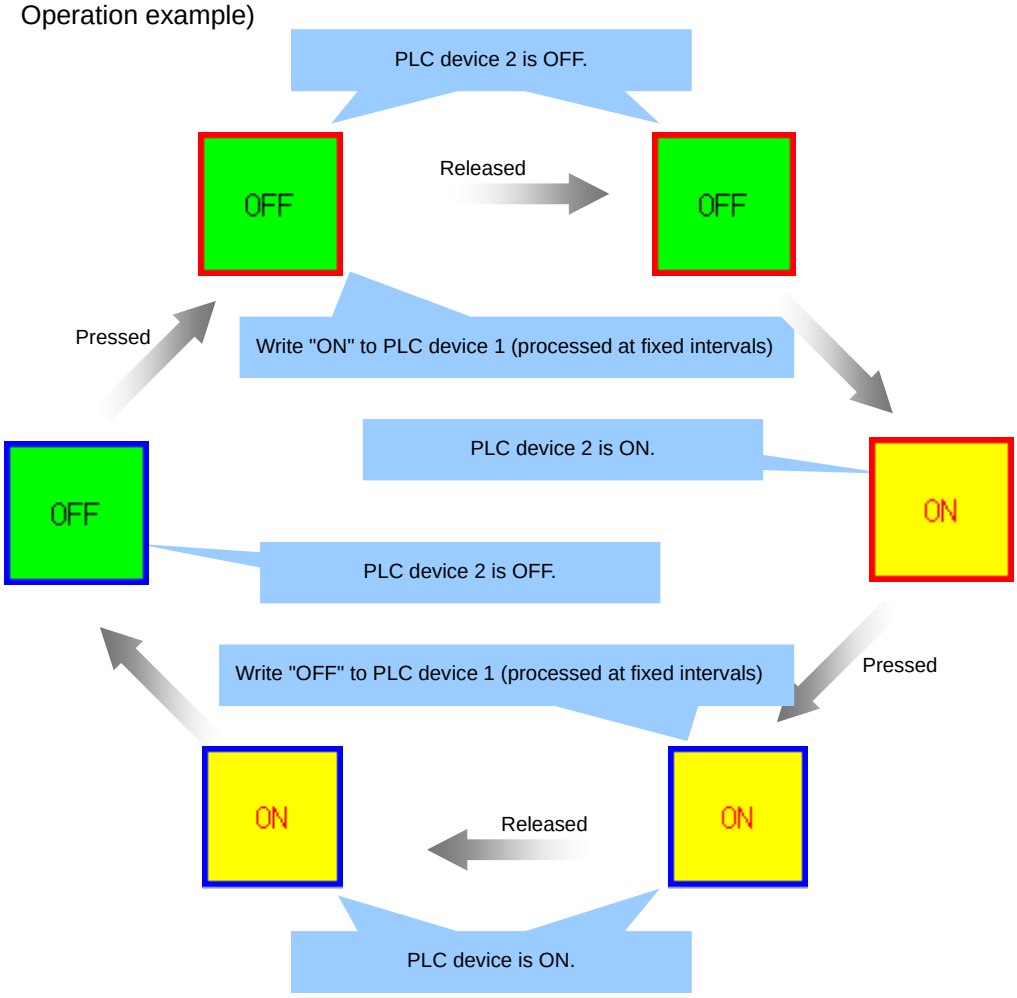
Display action changes as follows according to the ON/OFF of PLC device 1 and 2.

PLC device 1 (Write)	**OFF(0)**		**ON(1)**	
PLC device 2 (Read)	**OFF(0)**	**ON(1)**	**OFF(0)**	**ON(1)**
Color/Pattern	At the time of OFF *1	At the time of ON *1	At the time of OFF *1	At the time of ON *1
Character attribute (character color)	At the time of OFF	At the time of ON	At the time of OFF	At the time of ON
Solid frame	At the time of OFF *2	At the time of OFF *2	At the time of ON *2	At the time of ON *2
Caption (character string)	At the time of OFF	At the time of ON	At the time of OFF	At the time of ON
Blink device	Enable	Disable	Enable	Disable

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.

*2: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



(5) None (Lamp)

Nothing is written to the PLC device, but the display action changes according to the combination of PLC device 1 (Write) and PLC device 2 (Read).

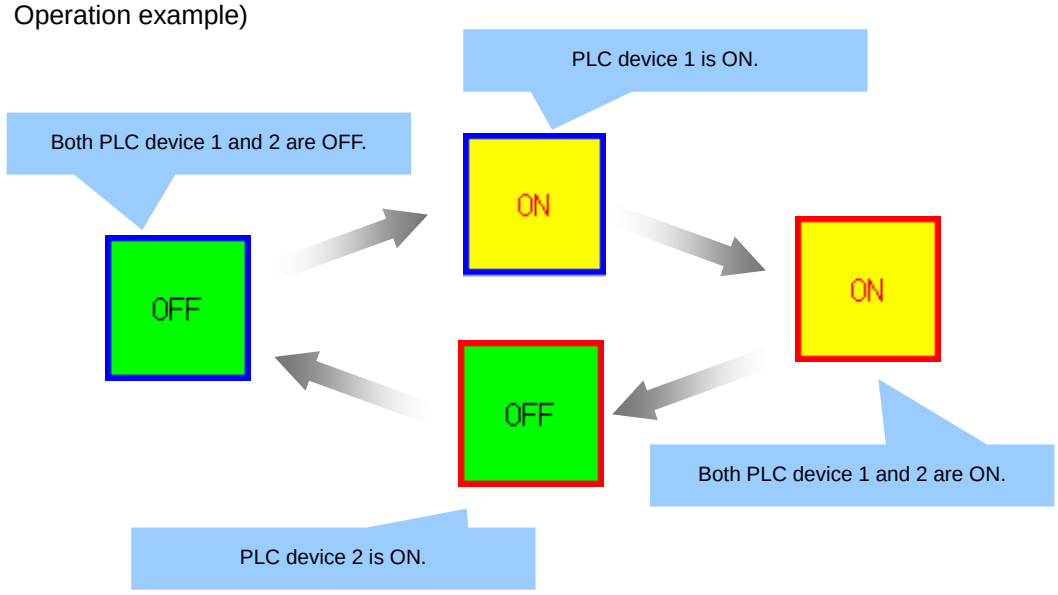
Display action changes as follows according to the ON/OFF of PLC device 1 and 2.

PLC device 1 (Write)	**OFF(0)**		**ON(1)**	
PLC device 2 (Read)	**OFF(0)**	**ON(1)**	**OFF(0)**	**ON(1)**
Color/pattern	At the time of OFF *1	At the time of OFF *1	At the time of ON *1	At the time of ON *1
Character attribute (character color)	At the time of OFF	At the time of OFF	At the time of ON	At the time of ON
Solid frame	At the time of OFF *2	At the time of ON *2	At the time of OFF *2	At the time of ON *2
Caption (character string)	At the time of OFF	At the time of OFF	At the time of ON	At the time of ON
Blink device	Enable			

*1: Color/Pattern refers to the patterns, foreground colors, background colors and images at the time of ON and OFF.

*2: Solid frame refers to the solid frames at the time of ON and OFF.

Operation example)



NOTE

- When the action of PLC device 1 is set to a mode other than "Read", PLC device 1 is displayed always in the OFF state.
- When the action of PLC device 2 is set to a mode other than "Read", PLC device 2 is displayed always in the OFF state.
- When Disable device is ON with Disable function set to "Enable", the image at the time of disable is displayed.
- When the settings of Interlock device 1 and 2 are valid, the image at the time of interlock is displayed.
- The group No. setting is invalid.

7.3.16 PLC Message (GNCPLCMessage)

PLC message is a control that displays a message according to the status of PLC device in NC, by obtaining it from the message definition text file (UNICODE text).

To create a PLC message, select [PLC message] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.16.1 Property Settings

The property settings for the PLC message are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Show/Hide	:	Specify whether the control is displayed or hidden.
NC Number	:	Specify the No. of NC to connect.
Ground	:	Specify the foreground and background.
Default	:	Specify the default character color and default background color to be applied when they are not specified in the message definition text file.
Message file	:	Specify the message definition text file to be displayed at the control.
PLC device	:	Specify the target PLC device.
Character attribute	:	Specify the character attribute of captions.
Caption character string scroll	:	Specify the scroll of the caption character string.
Blink	:	Set the blink of the caption character string.
Callback function	:	Specify whether or not the callback functions are provided.

NC number

Item	Description
NC Number	Specify the No. of NC to connect (Usually set to "1").

Ground

Item	Description
Ground	Specify the foreground and background (Usually set to "0").

Default

Item	Description
Default character color	Specify the default character color. This color is applied when a character color is not specified in the message definition text file.
Default background color	Specify the default background color. This color is applied when there is no message or when a background color is not specified in the message definition text file.

Message File

Item	Description
Message folder	Select the folder in which the message definition text file is stored.
Message file	Select the message definition text file name from the file resource ID.

NOTE

- To store the message definition text file in the selected message folder, the absolute path has to be defined in the Config.ini file. The relationship between the description in the combo box and the actual folder is shown below.

  <Example of M700/M700VW(WindowsXPe)>
  [MESS_CONTROL]
  MESSDATA0=C:\MESSDATA0\
  MESSDATA1=C:\MESSDATA1\
  MESSDATA2=C:\MESSDATA2\
  :
  MESSDATA7=C:\MESSDATA7\

  <Example of M700(WindowsCE)>
  [MESS_CONTROL]
  MESSDATA0=\Memory Card\Custom\MESSDATA0\
  MESSDATA1=\Memory Card\Custom\MESSDATA1\
  MESSDATA2=\Memory Card\Custom\MESSDATA2\
  :
  MESSDATA7=\Memory Card\Custom\MESSDATA7\

  <Example of M70/M70V/M700VS/E70 Series>
  [MESS_CONTROL]
  MESSDATA0=/custom/MESSDATA0/
  MESSDATA1=/custom/MESSDATA1/
  MESSDATA2=/custom/MESSDATA2/
  :
  MESSDATA7=/custom/MESSDATA7/

- When the message file size is large, the file consumes the custom release data storage capacity. Therefore, the size of each message file should be 200K byte or less.

- Up to eight message files are available.
  When more than one message file is used, the total number of lines in the message files has to be 65535 or less.
- The file name to be described in the file resource ID has to be 40 or less characters in length (including the extension).

PLC Device

Item	Description
PLC device method	Select the method to display a message between "Bit designation" and "No. designation".
PLC device	Specify the address of the PLC bit device (X,Y,M)/word device (D,R) for the PLC message display.
Starting bit position	Specify the start position to read the PLC device (0 to 15). This setting is valid when a word device (D,R) is selected for the PLC device for which "Bit designation" is selected (but not used when a bit device is selected (fixed to zero)). When a word device (D,R) is selected, a message is displayed according to the state between the starting bit position and the number of bits to use.
Number of bits to use	Specify the number of bits fro reading the PLC device. When "Bit designation" is selected, the bit devices between the starting bit position and the number of bits to use are read in the ascending order to display the message (1 to 512). When "No. designation" is selected, specify the number of bits to be handled as numbers (Specify the value from 1 to 16. "16" is specified when a value other than 1 to 16 is set).
Display action	Specify whether to retain the message or delete it when a message to display does not exist after the state of the PLC device has changed. Select between "Keep the display" and "Cancel the display".
Turn back	Select this to display the message in two lines when the message length has exceeded the control's display range. Select between "None" and "Yes".

NOTE

◆ Set as shown below according to the PLC device types.

(1) Setting example of "Bit designation"

• To display a message using the bit devices M0 to M10.

PLC device → M0
Start bit position → 0
Number of bits used → 11

• To display a message using the word devices R0 (bit 8) to R10 (bit 7).

PLC device → R0
Start bit position → 8
Number of bits used → 160

(2) Setting example of "No. designation"

• To display a message using the one byte between M0 and M7.

PLC device → M0
Start bit position → 0
Number of bits used → 8

• To display a message using the two bytes between the R0's fourth bit and R1's third bit

PLC device → R0
Starting bit position → 4
Number of bits to use → 16

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.16.2 Complements

Message Definition Text File

To use a PLC message, it is necessary to prepare the message definition text file.
This file has to be described by UNICODE text.

Up to 65535 lines of message can be included in the file, when one message is regarded as one line.
It is possible to describe messages in more than one language in the message definition text file.
Use a comma or tab to separate each item.

The following items are described in the message definition text file.

Item	Setting value	Description
Message No.	1 to 65535 (Omissible)	Describe the message No. when "No. designation" is selected for the PLC device method. When the PLC device value corresponds to the message No., the message character string is displayed. Describe the message No. in a decimal number. When "Bit designation" is selected for the PLC device method, the message No. description is disabled.
Bit position	0 to 511 (Omissible)	Describe the bit position when "Bit designation" is selected for the PLC device method. The bit position is searched in the ascending order within the range starting from the starting bit position and made up of the number of bits to use. When the bit position is matched, the message character string is displayed. Describe the bit position in a decimal number. When "No. designation" is selected for the PLC device method, the bit position description is disabled.
Character color No. *1	0 to 255 (Omissible)	Describe the character color to display a message character string. When the description about the character color is omitted, the message is displayed in a color specified with "Default character color".
Background color No. *1	0 to 255 (Omissible)	Describe the background color to display a message character string. When the description about the background color, the message is displayed with a background color specified with "Default background color".
Message character string	Up to 256 character (the number of characters per language)	Describe the message character string. Enclose the message with double quotations ("). If you wish to describe a double quotation (") or back slash (\), use "\"" or "\\". To display the message character string in two lines, describe "\n" at the line feed position (up to 20 lines).

*1: For the color Nos., refer to "Appendix 5 Default Palette Color".

Message definition text file is described as below.

(1) Description example for "Bit designation"

(Message No.),	(Bit position),	(Character color),	(Background color),	Message character string(English, Japanese)
,	0,	0(black),	15(white),	" bit type "," Bit designation "
,	511,	12(red),	2(green),	" \" bit \" \n type"," \" Bit \" \n designation "

(2) Description example for "No. designation"

(Message No.),	(Bit position),	(Character color),	(Background color),	Message character string (English, Japanese)
1,	,	15(white),	0(black),	" number type "," No. designation "
65535,	,	9(blue),	14(yellow),	" \" number \" \n type "," \" No. \" \n designation "

NOTE

- To describe messages in more than one language, the messages are described in the following order.

1: English	2: Japanese	3: German	4: French	5: Italian
6: Spanish	7: Chinese (traditional)	8: Korean	9: Portuguese	10: Dutch
11: Swedish	12: Hungarian	13: Polish	14: Chinese (simplified)	15: Russian
16: Turkish	17: Czech			

- For the unused languages, insert delimiters (commas or tabs) to the omitted languages to align the languages and message positions.
  To use the message character strings in English, Japanese, French and Portuguese, describe as follows.
  (Example) 1,,0,15,English,Japanese,,French,,,,,Portuguese

- The language to display messages is changed according to "#1043 lang (Select language displayed)" [base common parameter].
  When a language is switched to the one in which messages are not described, the messages are not displayed.
- If a semicolon (;) is described at the top of the line, the line is handled as a comment line.

7.3.17 Menu (GNXMenu) ; Menu display part

The menu display part includes the menu ON, OFF and Disable statuses, which can change the menu selection state.
Two types of settings are available for the menu: one-stage menu (icon + menu name) and two-stage menu (two stages of menu name).

To create a menu, select [Menu] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.17.1 Property Settings

The property settings for displaying the menu are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Menu type	:	Select the menu type.
Color	:	Select the color.
Callback function	:	Specify whether or not the callback functions are provided.

Menu Type

Item	Description
MenuType	Select the menu type from the following two types. 1StepMenu(VGA)・・・・・・・・・・・・・・・・・・・・・・・・・・・・ One-row menu for VGA 2StepMenu(VGA)・・・・・・・・・・・・・・・・・・・・・・・・・・・・ Two-row menu for VGA 1StepMenu(XGA)・・・・・・・・・・・・・・・・・・・・・・・・・・・・ One-row menu for XGA 2StepMenu(XGA)・・・・・・・・・・・・・・・・・・・・・・・・・・・・ Two-row menu for XGA
Horizontal position	Set the position to display the menu character string (Left-justifying, Centering)

Color

Item	Description
StringForeColor	Set the normal character color.
StringPushedColor	Set the character color when selected.
1StepForeBackColor	Set the normal background color of the one-row menu.
1StepPushedBackColor	Set the background color of the one-row menu when selected.
2StepUpperForeBackColor	Set the normal background color of the upper row of the two-row menu.
2StepUpperPushedBackColor	Set the background color of the upper row of the two-row menu when selected.
2StepLowerForeBackColor	Set the normal background color of the lower row of the two-row menu.
2StepLowerPushedBackColor	Set the background color of the lower row of the two-row menu when selected.

For the other properties, refer to "7.1 Common Functions of Controls".

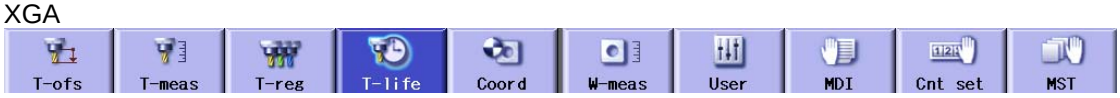
7.3.17.2 Complements

Screen Specifications

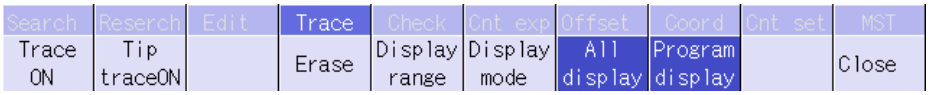
Screen Images

"One-row menu"
VGA

XGA

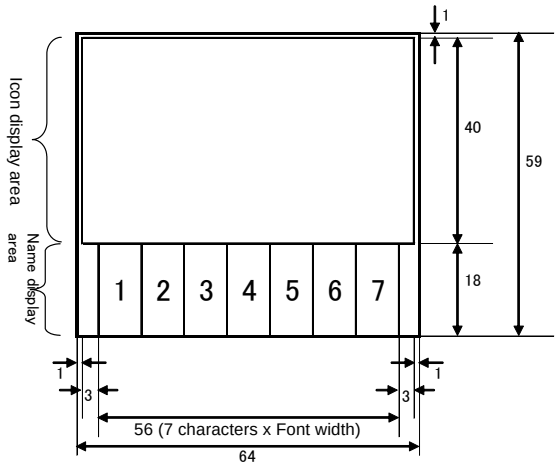


"Two-row menu"
VGA

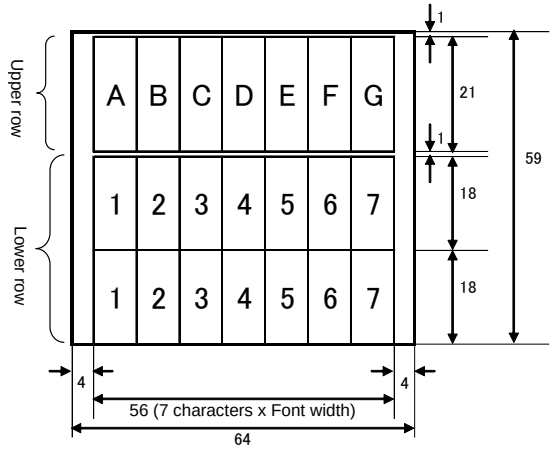


XGA

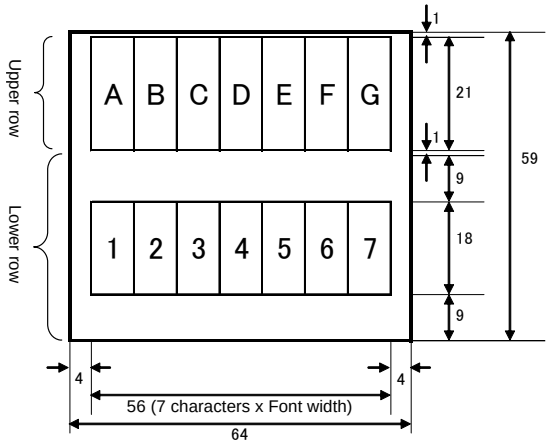
<Dimensions of one menu button>
Height: 59 pixels Width: 64 pixels



One-row menu

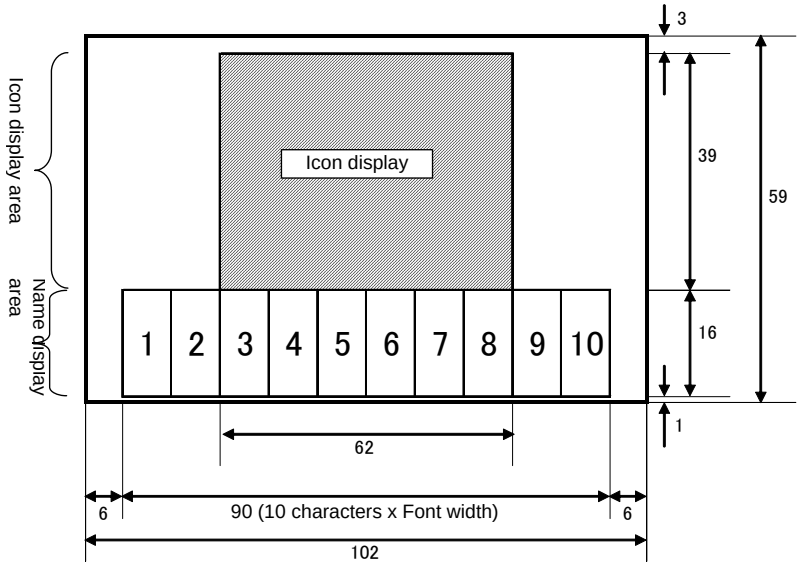


Two-row menu (two lines in the lower row)

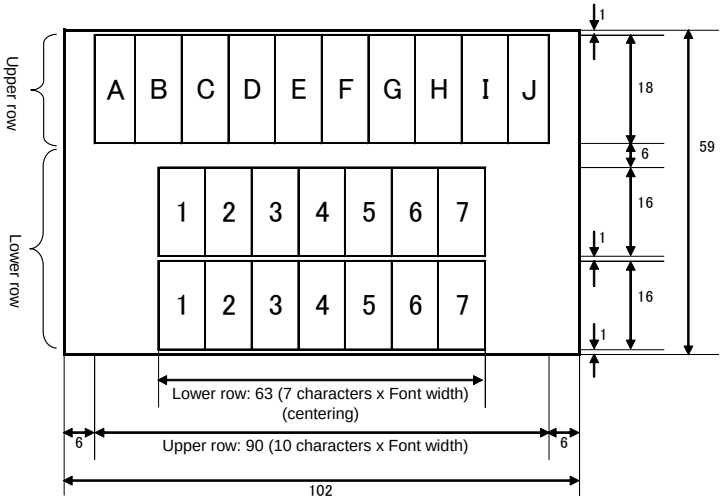


Two-row menu (one line in the lower row)

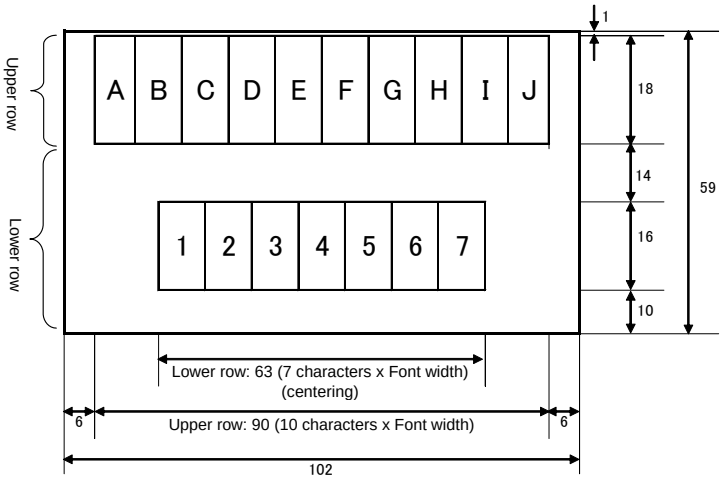
<Dimensions of one menu button for XGA>



One-row menu



Two-row menu (two lines in the lower row)

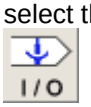


Two-row menu (one line in the lower row)

7.3.18 FileInOut (GNXFileTransfer) ; Input/Output Control

The input/output control part is used to input and output NC files between the NC memory and an external device.
The hard disk built in the NC unit is also handled as an external device.

To create an input/output control, select [FileInOut] from the [Control] menu of NC Designer, or select the icon shown below.



7.3.18.1 Property Settings

The property settings for the input/output control are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Main part area	:	Specify the color attribute, solid frame and message font type of the main part area.
Bar graph area	:	Specify the width and color attribute of the bar graph area, and whether to enable the gradation effect on the area.
Message area	:	Specify the color attribute, and whether to display a message on this area.
Callback function	:	Specify whether or not the callback functions are provided.

Main Part Area

Item	Description
MainBackColor	Specify the background color of the input/output control.
FrameVisible	Select whether to use the solid frame. Select between "True" and "False".
FontType	Specify the message font type. Normal・・・・・・・・・・・・・・・・・Normal font Normal Bold・・・・・・・・・・・・・Normal bold font Middle・・・・・・・・・・・・・・・・・Middle-sized font Big・・・・・・・・・・・・・・・・・・・・Font with double height and width

Bar Graph Area

Item	Description
BarGraphForeColor	Specify the color of the bar graph that is refreshed at the file transfer.
BarGraphBackColor	Specify the color of the bar graph that is drawn at the initial display.
BarGraphWidth	Specify the width of the bar graph area (100 to 1800).
SetBarGradation[*1]	Specify whether to enable gradation effect when drawing the foreground of the bar graph. Select between "True" and "False".

*1 Gradation is not supported by M70, M70V, M700VS and E70 Series. Thus, even when "True" is selected, the bar graph is displayed in a plain color.

Message Area

Item	Description
MessageVisible	Switch "True (show)" and "False (hide)" of the message on the control.
NormalMessageFontColor	Specify the normal message character color.
NormalMessageBackColor	Specify the normal message background color.
ErrorMessageFontColor	Specify the error message character color.
ErrorMessageBackColor	Specify the error message background color.

Callback Function

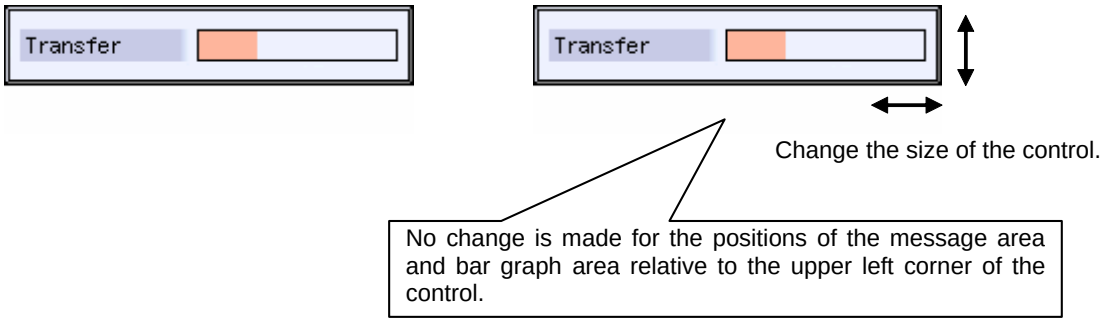
Item	Description
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.18.2 Complements

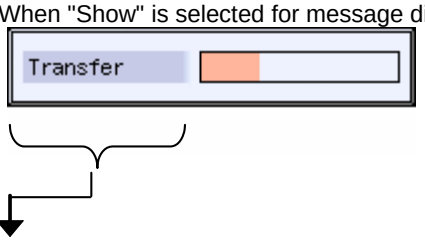
Screen Specifications

Screen Images

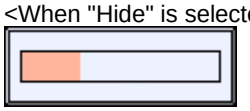


[Switching of Show/Hide of message]

<When "Show" is selected for message display>



<When "Hide" is selected for message display>

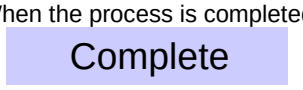


The message area is aligned to the left.

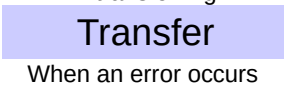
[Types of message to display]

Initial state

When the process is completed



In transferring



When an error occurs

* A message is not displayed in the initial state. Therefore, the background color of the message area is the same as the background of the control.

List of Available File Paths

A GCS function has to be described to input/output NC files using the input/output control. The file path information is given to the GCS function argument. The list below shows the available file path information.

Device	Device name	Data type	Directory	File name
NC memory	M01:	Machining program	/PRG/USER/	(Program No.)
		Parameter	/PRM/	ALL.PRM
		NC data	/DAT/	
		Tool compensation data		TOOL.OFS
		Tool life management data		TLIFE.TLF
		Common variable data		COMMON.VAR
External device - HD - FD - Memory card - DS - USB memory	HD: FD1: MEM: M01:/IC1 USB :	To specify a file in the HD device, specify the file under D:/NCFILE. Example) For HD:/ABC/100.PRG: Specify the path D:/NCFILE/ABC/100.PRG. For DS and USB, set "M01:IC1/" or "USB:/" to specify the root directory. "USB:" is available only for M700VS/M70V/E70.		
External device (Direct designation)	C: D:	A direct path designation is possible only for the files in HD (Drive C or D). Example) For C:/WINDOWS/ABC.TXT: Specify the path C:/WINDOWS/ABC.TXT.		

[Precautions relating to file transfer]

(1) If the transfer destination becomes full during the file transfer, the data transferred up to the point is registered and then an error occurs.
(2) Up to 223 files can be registered to the root directory of an FD (including the directory).
(3) Designation of multiple files using a wild card "*" is not possible. Therefore, if "*" is included in the file name, an error occurs.
(4) f the same file name is specified for the transfer source and destination, the file transfer is not carried out, and an error code is returned as the return value.
(5) If you wish to save a file under the same name as the transfer source file, it is not necessary to give the file name to the transfer destination file path.
[File path setting example] When common variables in NC memory is saved in HD
Transfer source file path: M01:/DAT/COMMON.VAR
Transfer destination file path: D:/NCFILE/
To specify a directory as the file information, "/" has to be added at the end of the directory name.
(6) "/" is used for the paths in the table, but it is also possible to use "\" to specify the path.
(7) When the file information is specified, the case (uppercase and lowercase) is ignored.
(8) If the input/output control part is set to "hidden" during transferring, the transfer is interrupted, and when the control part is displayed, the transfer is resumed.
(9) If the input/output control part is deleted during transferring, the transfer is interrupted. Therefore, do not delete the part during the transfer.
(10) Do not use two-byte characters in specifying a file path.

Related NC Parameters

The available file name varies depending on the setting of "#1166 fixpro". Refer to the table below.

Status	#1166 fixpro	Available file name
General program	0	32 characters
Fixed cycle	1	9-digit number only
Machine tool builder macro	(Machine tool builder password)	32 characters

Restrictions

Restrictions for creating a control object are shown below.

(1) Unsupported device
RS232C and Ethernet are not supported.
(2) All file input/output function
All file input/output function is not supported
(3) Deletion of directory
When a file is included in a directory to delete, it is not possible to delete the directory.
(4) Program name
There are the following restrictions for the name of the files to be created or transferred in or from NC memory.
(a) Up to 32 characters including the extension
(b) The characters available in file name and directory name are one-byte numerals, one-byte uppercase alphabets and the one-byte symbols that can be recognized by Windows.
Unavailable characters: \ / : , * ? " < > | lowercase letters (a to z) and a space
(c) Edit lock B and C and Program display lock are effective only on the files in NC memory whose names are made up of one-byte numerals.
ex) When Edit lock B (8000 to 9999) is active

File name	Characteristics	Change
8000	One-byte numerals only	Disable
8000.PRG	With an extension	Enable
08000	One-byte numerals only. Zero at the head.	Disable
8000A	Characters other than numerals	Enable

(d) The following files can not be handled as a file name.
- The extension is "$$$", “$$0”, “$$1”, “$$2”, “$$3”, “$$4”, “$$5”, “$$6”, “$$7”, “$$8” or “$$9”.
- "0" (the file name is made up of a one-byte zero)
(5) Up to one input/output control part should be located per screen. Set the screen on which the input/output control part is located so that the instance is not held. Not doing so may cause a memory shortage.

7.3.19 AlarmMessage (GNXAlarmMessage) ; Alarm Display Part

The alarm display part is used to display the alarm No. and alarm message character string when an alarm occurs.
This part displays NC alarms and PLC alarms, but does not display the stop code.
When more than one alarm occurs at a time, the messages are alternately displayed in a two-second cycle. Up to 60 characters can be included in a message.

To create an alarm display part, select [AlarmMessage] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.19.1 Property Settings

The property settings for the alarm display are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color	:	Specify the display color of the control.
Character attribute	:	Specify the character attribute of captions.
Character string display form	:	Specify the part system to be displayed on the control.

Character Attribute

Item	Description
FontType	Specify the font size and thickness. Normal・・・・・・・・・・・・・・・・・・Normal font Normal Bold・・・・・・・・・・・・・Normal bold font Middle・・・・・・・・・・・・・・・・・Middle-sized font Big・・・・・・・・・・・・・・・・・・・・・Font with double height and width

Character String Display Form

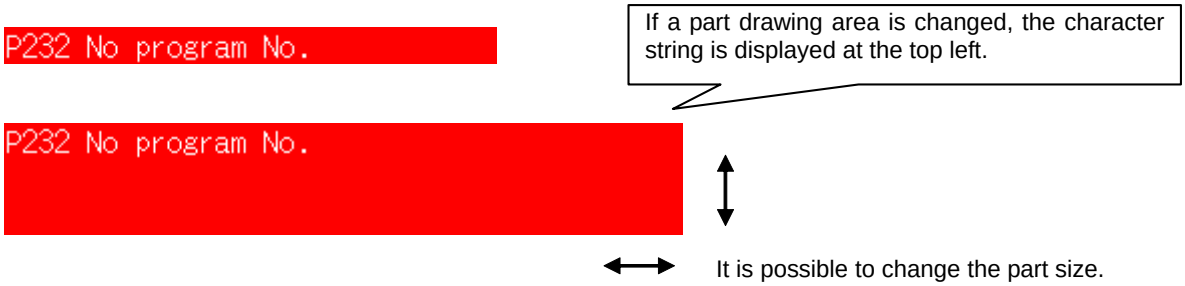
Item	Description
DisplayMessage	Specify how to display the character string (0 to 2). 0: Not display the residual when the message is split. 1: Display the residual when the message is split. 2: Dependent on the parameter setting (#11021 PLC mesg disp type) When #11021 is 0: Not display the residual when the message is split. When #11021 is 1: Display the residual when the message is split.

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.19.2 Complements

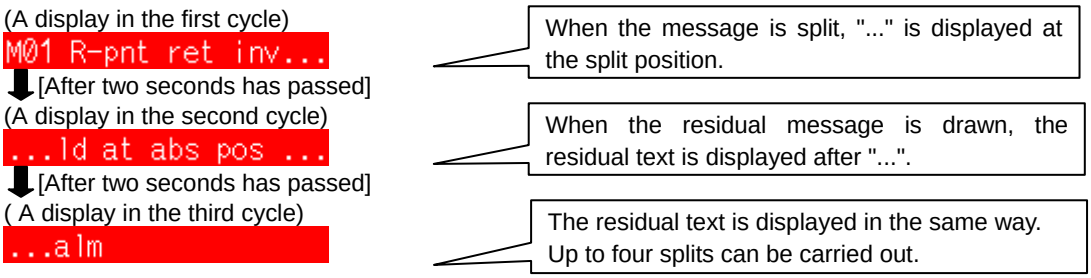
Screen Specifications

Screen Images



[Message split display (When "1" or "2" is selected for the property "Character string display form")]
Up to 60 characters can be included in a message. If the specified part size is too small to display 60 characters at a time, the message to display is split. If the message is split (up to four splits), they are displayed in a two-second cycle, and "... (three characters)" is displayed at the split position.

[Example of split message display] When a part of a specified size can display up to 20 characters



(Note1) The height of the cell is the same as the height of the font.

(Note2) Even when the height of the part is extended, the message is displayed in one line. If the specified height of the part is smaller than the font height, the message is not displayed.

(Note3) The drawing area in a cell (the number of characters to display) is determined depending on the part width. However, if the specified width cannot display the character string of 20 or more characters, the message is not displayed.

(Note4) For a PLC alarm with a classification No., the message and the classification No. are displayed. When "Not display the residual when the message is split" is selected, only the message is split.

(Note5) When "2: Dependent on the parameter setting" is selected, up to 40 characters are displayed for a PLC alarm and operator message at a time, even if the control width is extended enough.

[Alarm display for multiple part system control]
• The part system name is displayed at the top of the message. The message is displayed in order from $1.

Part system name

List of Alarms

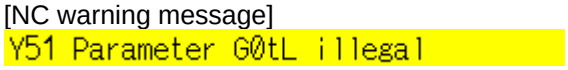
Type	Character	Background	Description
NC alarm message	White	Red	An operation alarm, program error, MCP alarm, servo alarm or system alarm is displayed.
NC warning message	Black	Yellow	
PLC alarm message	White	Red	A message such as the details of machine error is displayed by use of user PLC.
Operator message	Black	Yellow	The operator message is displayed by use of user PLC.
NC alarm message during background check	White	Orange	When a program error occurs during the check, or when the macro alarm message (a message displayed with #3000 variable command) is displayed, the message is displayed.

(Note1) If more than one alarm occurs simultaneously, the messages are alternately displayed in a two-second cycle (when messages are split, the next message is displayed after all the messages have been displayed).

(Note2) The character color and background color of the messages are fixed to the above, and cannot be changed.

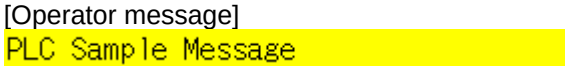
[NC alarm message]

[NC warning message]



[PLC alarm message]

[Operator message]



[NC alarm message during background check]

7.3.20 MonitorStatus (GNXMonitorStatus) ; Operation Status Display Part

Operation status is a control that displays the NC operation status. This can display the operation status separately for each part system when multiple part system control is performed.

To create an operation status display part, select [MonitorStatus] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.20.1 Property Settings

The property settings for the operation status display are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color	:	Specify the display color of the control.
Solid frame	:	Specify the solid frame of the control.
Character attribute	:	Specify the character attribute of captions.
Part system designation	:	Specify the part system to be displayed on the control.
Update cycle	:	Specify the update cycle of the operation status display.
Callback function	:	Specify whether or not the callback functions are provided.

Character attribute

Item	Description
FontType	Specify the font size and thickness. Normal・・・・・・・・・・・・・・・・・・Normal font Normal Bold・・・・・・・・・・・・・Normal bold font Middle・・・・・・・・・・・・・・・・・Middle-sized font Big・・・・・・・・・・・・・・・・・・・・Font with double height and width
SystemNameColor	Specify the character color of the part system displayed on the control.

Part system designation

Item	Description
SystemNumber	Specify the No. of the part system to be displayed (0 to 4). When "0" is set, valid part systems are displayed.

Update Cycle

Item	Description
RefreshFrequency	Specify the number of times to thin out the timer event processes (1 to 100).
RefreshTiming	The display is refreshed when the counted number of "RefreshFrequency" reaches the number of counts specified with "RefreshTiming" (0 to 99).

Callback Function

Item	Description
OnInit	Select "Yes" to add a process to be executed after the panel/window is displayed.
OnQuit	Select "Yes" to add a process to be executed before the panel/window is hidden.

For the other properties, refer to "7.1 Common Functions of Controls".

7.3.20.2 Complements

Screen Specifications

Screen Images

- Without frame

- With frame

If a part drawing area is changed, the character string is displayed at the top left. However, if the specified area is smaller than the character string drawing area, the character string is not displayed.

It is possible to change the part size.

[Display the operation status of the selected part system]

By setting the property (Part system designation = 1 to 4), the operation state of the selected part system can be displayed. However, if the setting is greater than the number of valid part systems, the state of the 1st part system is displayed. If the number of valid part systems is 1, the state is left aligned without indicating "$1").

- To display the 1st part system

- To display the 4th part system

- For a system controlling one part system

[Display the operation status of valid part systems]

By setting the property (Part system designation = 0), the operation states of the valid part systems can be displayed. In this case, the number of part systems to be displayed varies depending on the part size.

- To display the 1st and the 2nd part systems

- To display the 1st to the 3rd part systems

- To display the 1st to the 4th part systems

- For a system controlling one part system

List of Operation Status

The operation status symbol displayed on the control changes as shown below according to the NC operation state.

Symbol	Operation status	Character color
EMG	In emergency stop	Red
RST	Resetting NC	White
LSK	Paper tape reader is in label skip state	White
BST	In block stop	White
HLD	Operation halted	White
SYN	Synchronizing	White
CRS	Waiting for cross conversion	White
AUT	In automatic operation	White
RDY	Operation completed state	Green

(Note 1) The character colors for the operation status symbol are fixed as shown above.

7.3.21 Time (GCNXTime) ; Time Display Part

The time display part is used to display the current time.

To create a time display part, select [Time] from the [Control] menu of NC Designer, or select the icon shown below.

7.3.21.1 Property Settings

The property settings for the time display part are divided into the followings.

Control name	:	Specify the control name.
Position/size	:	Specify the position and size of the control.
Color	:	Specify the display color of the control.
Solid frame	:	Specify the solid frame of the control.
Character attribute	:	Specify the character attribute of captions.

Character attribute

Item	Description
FontType	Specify the font size and thickness. Normal・・・・・・・・・・・・・・・・・・Normal font Normal Bold・・・・・・・・・・・・・Normal bold font Middle・・・・・・・・・・・・・・・・・Middle-sized font Big・・・・・・・・・・・・・・・・・・・・Font with double height and width
ForeColor	Specify the character color.

For the other properties, refer to "7.1 Common Functions of Controls".

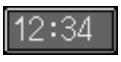
7.3.21.2 Complements

Screen Specifications

Screen Images

- Without frame

- With frame



If a part drawing area is changed, the character string is displayed at the top left. However, if the specified area is smaller than the character string drawing area, the character string is not displayed.

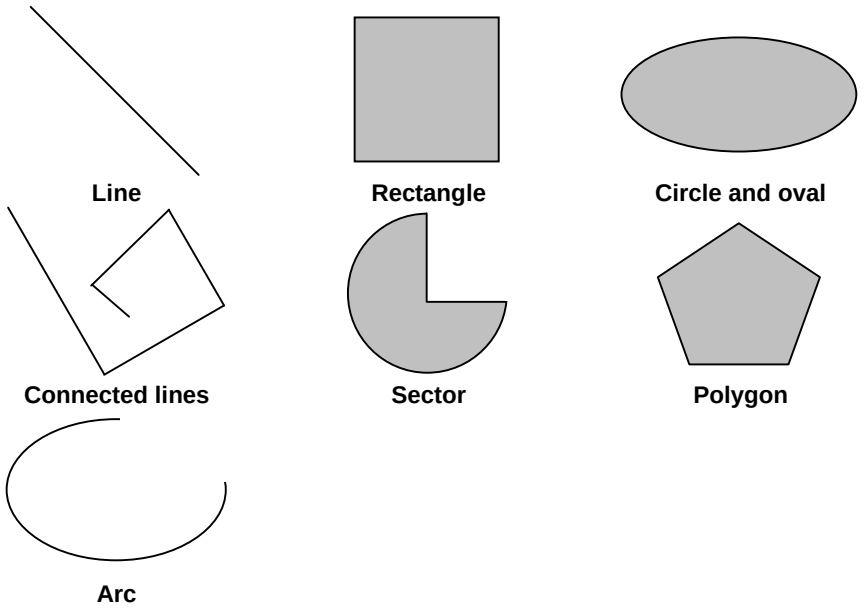
It is possible to change the part size.

8. Figure

This section describes each figure and property settings.

8.1 What Is Figure?

Figure is graphic data displayed as a fixed matter. The figure includes the following seven variations.



Name	Description
Rectangle	A rectangle is drawn.
Circle&Oval	The inscribed circle of the designated rectangle is drawn.
Straight Line	A line is drawn.
Poryline	Connected lines are drawn.
Polygon	Lines are connected to draw a polygon.
Sector	After a circle is drawn, the angle is designated to draw a sector.
Arc	An arc is drawn.

8.2 Figure Creation Method

The method for arranging the figure in the screen and specifying properties is described.

8.2.1 Drawing a New Figure

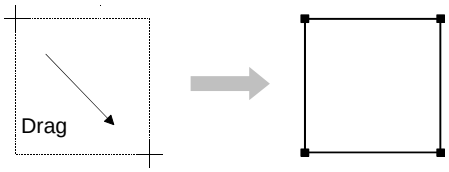
To draw a new figure, open the [Figure] menu, or select the figure tool bar icon.

Rectangle, Circle, Oval and Line

1. Move the cursor to the starting point of the rectangle, circle, oval or line.

2. The cursor changes to the shape shown below.

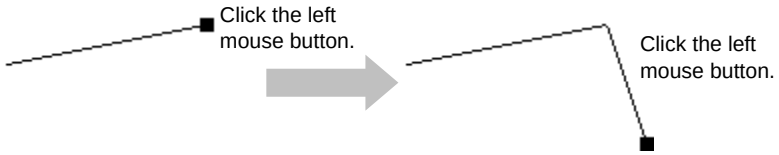
3. Drag the cursor to the end point of the rectangle, circle, oval or line.



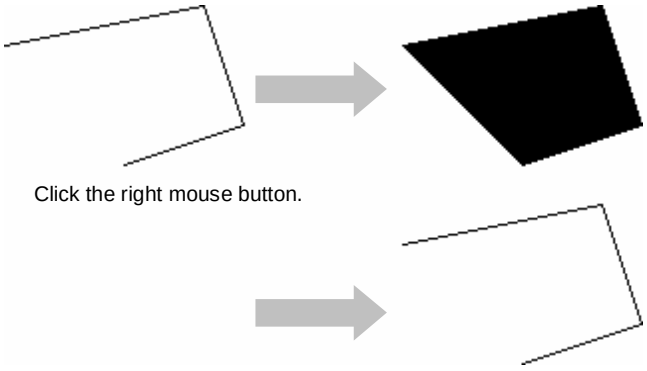
Polygon and Connected Lines

1. Move the cursor to the starting point of the polygon or connected lines and click the left mouse button.

2. Move the cursor to the next point and click the left mouse button. Repeat the operation to draw all vertexes of the polygon or connected lines.



3. Click the right mouse button at the last vertex to exit from the polygon or connected line drawing mode.

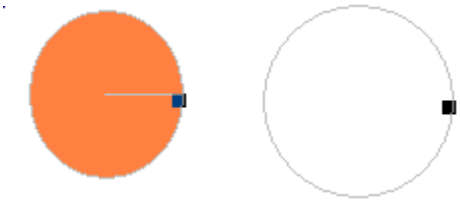


Sector and Arc

1. Move the cursor to the starting point of the sector or arc, and click the left mouse button.

2. Drag to draw a circle or oval.

3. A black box is placed at the 3 o'clock position on the perimeter of the drawn circle or oval.



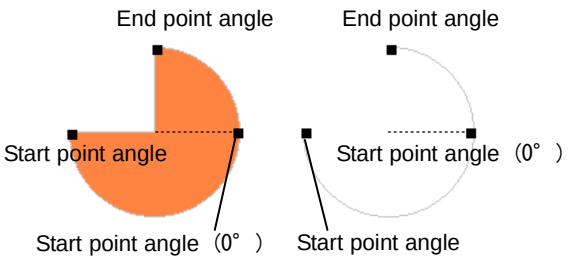
4. Place the cursor at the black box and, after the cursor changes to "+," drag the mouse to the desired position. Dragging should be done within the range of 360°.

5. Click the right mouse button to exit from the sector or arc drawing mode.

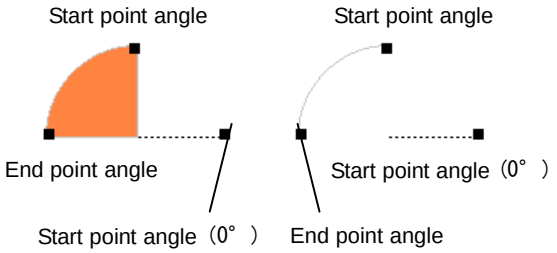
NOTE

- Drag while holding down the [Shift] key to create a figure having an equal vertical and horizontal ratio.
- Drag up/down or left/right while holding down the [Ctrl] key to change the size of the figure evenly up/down or to the left/right.
- The sector or arc can be set from "Starting point angle･Terminal point angle" in "Property".
  Display example)

When setting start point angle 180°/
end point angle 90°



When setting start point angle 90°/
end point angle 180°



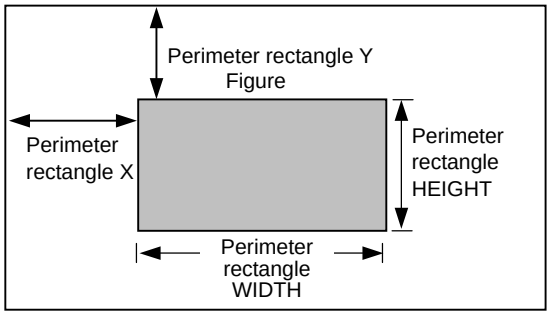
- Press [Esc] button to exit from the drawing mode.

8.3 Common Functions of Figure

8.3.1 Position/Size

Specify the position and size of the control.

Item	Description
Perimeter rectangle X	Specify the horizontal position from the upper left of the page/view frame of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position from the upper left of the page/view frame of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (0 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (0 to 1920).



Page/view frame

NOTE

- Note that the figure may be arranged outside the page/view frame according to some position and size settings.
- The size changes, serving the upper left point of the figure as the origin, when the size is changed.
- For the line, designate the coordinates of the starting and end points.

8.3.2 Perimeter Line

Specify the presence, color and other particulars of the perimeter line.

Item	Description
Perimeter line	Select whether the line is given around the figure or not ("Yes" or "None").
Perimeter line color	Specify the color of the perimeter line.
Perimeter line width	Specify the width of the perimeter line (1 to 20).
The kind of perimeter line	Select the line pattern of the perimeter line among "Solid line," "Pattern 1," ... and "Pattern 8." The line type is valid if the "perimeter line width" is "1."

No perimeter line With line pattern "2" or thicker line size

NOTE

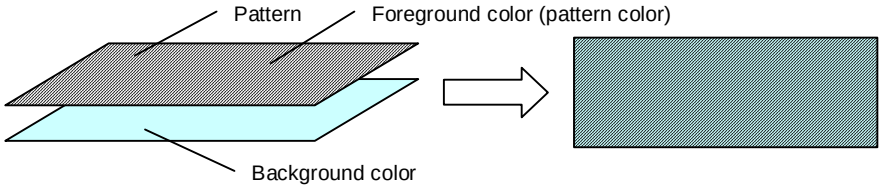
- The perimeter line is expressed inside and outside the perimeter. For perimeter line width "2," a single-dot line is drawn both inside and on the perimeter line. For "3," a single-dot line is drawn inside and outside the perimeter line. For "4," a two-dot line is drawn inside the perimeter line and a single-dot line is drawn outside.

8.3.3 Color/Pattern

Specify the color and pattern of the figure.

Item	Figure
Painting out foreground color	Specify the background color of the figure.
Painting out background color	Specify the foreground color (pattern color) of the figure.
Painting out pattern	Select the filling pattern among "Background painting out," "Foreground painting out," "Pattern 0" to "Pattern 37," and "With no painting out."

The relationship among the background color, foreground color and pattern is shown in the figure below.



NOTE

- For the displayed patterns, refer to Appendix.

8.4 Figure Settings

8.4.1 Rectangle

Item	Description
Perimeter rectangle X	Specify the horizontal position of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Perimeter line	Select whether or not to draw the line around the rectangle.
Perimeter line color	Specify the color of the perimeter line.
Perimeter line width	Specify the width of the perimeter line (1 to 20).
The kind of perimeter line	Select the line pattern of the perimeter line among the "Solid line" and "Pattern 1" to "Pattern 8."
Painting out foreground color	Specify the color of the pattern.
Painting out background color	Specify the background color.
Painting out pattern	Select the filling pattern among "Background painting out," "Foreground painting out," "Pattern 0" to "Pattern 37," and "With no painting out."

8.4.2 Circle and Oval

Item	Description
Perimeter rectangle X	Specify the horizontal position of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Perimeter line	Select whether or not to give a line around the circle or oval.
Perimeter line color	Specify the color of the perimeter line.
Perimeter line width	Specify the width of the perimeter line (1 to 20).
The kind of perimeter line	Select the line pattern of the perimeter line among the "Solid line" and "Pattern 1" to "Pattern 8."
Painting out foreground color	Specify the color of the pattern.
Painting out background color	Specify the background color.
Painting out pattern	Select the filling pattern among "Background painting out," "Foreground painting out," "Pattern 0" to "Pattern 37," and "With no painting out."

8.4.3 Line

Item	Description
Starting point X coordinates	Specify the X coordinate of the starting point in dots (0 to 2559).
Starting point Y coordinates	Specify the Y coordinate of the starting point in dots (0 to 1919).
Terminal point X coordinates	Specify the X coordinate of the end point in dots (0 to 2559).
Terminal point Y coordinates	Specify the Y coordinate of the end point in dots (0 to 1919).
Line color	Specify the line color.
Line width	Specify the width of the line in dots (1 to 20).
The kind of perimeter line	Select the line pattern among the "Solid line" and "Pattern 1" to "Pattern 8."

8.4.4 Connected Line

Item	Description
Perimeter rectangle X	Specify the horizontal position of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Line color	Specify the color of the connected lines.
Line width	Specify the width of the connected lines (1 to 20).
The kind of perimeter line	Select the line pattern of connected lines among the "Solid line" and "Pattern 1" to "Pattern 8."

8.4.5 Sector

Item	Description
Perimeter rectangle X	Specify the horizontal position (X coordinate) of the figure in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position (Y coordinate) of the figure in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Starting point angle	Specify the starting angle of the sector in degrees (°) (0 to 359).
Terminal point angle	Specify the end angle of the sector in degrees (°) (0 to 359).
Perimeter line	Select whether or not to give a line around the sector.
Perimeter line color	Specify the color of the perimeter line.
Perimeter line width	Specify the width of the perimeter line (1 to 20).
The kind of perimeter line	Select the line pattern of the perimeter line among "Solid line" and "Pattern 1" to "Pattern 8."
Painting out foreground color	Specify the color of the pattern.
Painting out background color	Specify the background color.
Painting out pattern	Select the filling pattern among the "Background painting out," "Foreground painting out," "Pattern 0" to "Pattern 37" and "With no painting out."

8.4.6 Polygon

Item	Description
Perimeter rectangle X	Specify the horizontal position of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Perimeter line	Select whether or not to give a line around the polygon.
Perimeter line color	Specify the color of the perimeter line.
Perimeter line width	Specify the width of the perimeter line (1 to 20).
The kind of perimeter line	Select the line pattern of the perimeter line among the "Solid line" and "Pattern 1" to "Pattern 8."
Painting out foreground color	Specify the color of the pattern.
Painting out background color	Specify the background color.
Painting out pattern	Select the filling pattern among "Background painting out," "Foreground painting out," "Pattern 0" to "Pattern 37" and "With no painting out."

8.4.7 Arc

Item	Description
Perimeter rectangle X	Specify the horizontal position of the figure (X coordinate) in dots (0 to 2559).
Perimeter rectangle Y	Specify the vertical position of the figure (Y coordinate) in dots (0 to 1919).
Perimeter rectangle WIDTH	Specify the width of the figure in dots (1 to 2560).
Perimeter rectangle HEIGHT	Specify the height of the figure in dots (1 to 1920).
Starting point angle	Specify the starting angle of the arc in degrees (°) (0 to 359).
Terminal point angle	Specify the end angle of the arc in degrees (°) (0 to 359).
Line color	Specify the color of the arc.
Line width	Specify the width of the arc (1 to 20).
The kind of perimeter line	Select the line pattern of the arc among the "Solid line" and "Pattern 1" to "Pattern 8."

9. Screen Editing

This section describes the screen editing operations of NC Designer.

9.1 Editing Operation

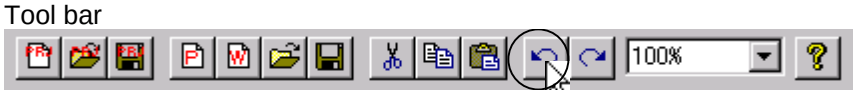
The editing methods of the object arranged in the screen are described.
The object indicates controls and figures arranged in the panel, window or screen.

9.1.1 Undo

Abandon a change and restore the original state before the change. Up to 10 operations can be undone. To undo, there are the following two methods.

- Select [Undo] from the [Edit] menu.
- Select the [Undo] button in the tool bar.

Tool bar



NOTE

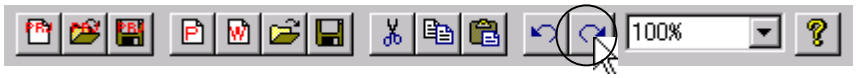
- The shortcut key corresponding to [Undo] is [Ctrl] + [Z].
- Note that the following operations cannot be undone with [Undo].
  - Entry of various properties of project, control, etc.
  - Registration, deletion and editing of resource

9.1.2 Redo

Redo the operation undone with the "undo" function.
Up to 10 operations can be redone. (Operations executed earlier than the "undo" record may not be executed.)

1. Select [Redo] from the [Edit] menu, or select the [Redo] button in the tool bar.

Tool bar



To redo further, execute [Redo] again.

NOTE

- The shortcut key corresponding to [Redo] is [Ctrl] + [Y].
- Note that the following operation may not be repeated with [redo].
  - Entry of various properties of project, control, etc.
  - Registration, deletion and editing of resource

9.1.3 Cut

Delete the selected object and store it in the clipboard.

1. Select an object.
   To cut multiple objects simultaneously, select all the desired objects to be cut.

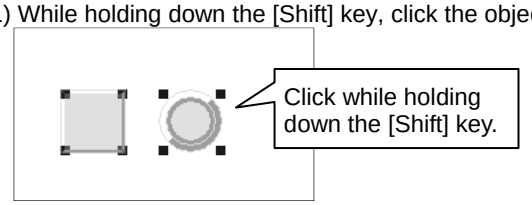
2. Select [Cut] from the [Edit] menu, or select the [Cut] button in the tool bar.

   Tool bar

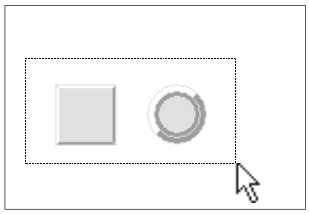
3. A confirmation dialog box is displayed. To continue, click on the [Yes] button.

NOTE

- The shortcut key corresponding to [Cut] is [Ctrl] + [X].
  How to select multiple objects
  (1) While holding down the [Shift] key, click the object with the mouse button.



  (2) Enclose the desired objects, using the cursor.

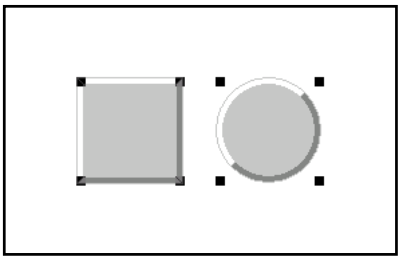


9.1.4 Copy

Copy the selected object and save it in the clipboard.

1. Select an object.

   To copy multiple objects simultaneously, select all the desired objects to be copied.



2. Select [Copy] from the [Edit] menu, or select the [Copy] button in the tool bar.

   Tool bar

NOTE

◆ The shortcut key corresponding to [Copy] is [Ctrl] + [C].

9.1.5 Paste

Paste the object(s) having been copied or cut and saved in the clipboard. The object is pasted with the same properties as those of the original object.

1. Display the destination screen.

2. Select [Paste] from the [Edit] menu, or select the [Paste] button in the tool bar.

Tool bar

NOTE

- The regular shortcut key corresponding to [Paste] is [Ctrl] + [V].
- When the control is pasted, the following control names are automatically specified.

Basic control object: GBasicControlxxxxx
Button object: GButtonxxxxx
Picture object: GPicturexxxxx
Label object: GLabelxxxxx
Text box object: GTextBoxxxxxx
List object: GListxxxxx
Check box object: GCheckBoxxxxxx
Radio button object: GRadioButtonxxxxx
Progress bar object: GProgressBarxxxxx
HTML browser object: GhtmlBrowserxxxxx
Scroll bar object: GScrollBarExxxxxx
Edit control object: Geditxxxxx
Input box: GInputBoxxxxxx
Ten-key: GSoftKeyxxxxx
NC data text box: GNCDataTextBoxxxxxx
PLC button: GNCPLCButtonxxxxx
PLC extension button: GNCPLCExButtonxxxxx
PLC text box: GNCPLCTextBoxxxxxx
PLC message: GNCPLCMessagexxxxx
Table: GNCTablexxxxx
Counter : GNXCounterxxxxx
CycleTime: GNXCycleTimexxxxx
Feedrate: GNXFeedratexxxxx
GModal M: GNXGModalxxxxx
GModal L: GNXGModal_Lxxxxx
GModal Simple: GNXGModalSimplexxxxx
LoadMeter: GNXLoadMeterxxxxx
MSTB: GNXMSTBxxxxx
ONB: GNXONBxxxxx
ProgramBuffer: GNXPrgBuffxxxxx
SPCommand: GNXSPCommandxxxxx
Menu: GNXMenuxxxxx
FileInOut: GNXFileInOutxxxxx
AlarmMessage: GNXAlarmMessagexxxxx
MonitorStatus: GNXMonitorStatusxxxxx
Time: GNXTimexxxxx
xxxxx: The smallest free number is automatically assigned in the range from 00000 to 99999.
The automatically assigned control name can be changed later.

9.1.6 Delete

Delete the selected object.

1. Select the object to be deleted.
   To delete multiple objects simultaneously, select all the desired objects to be deleted.

2. Select [Delete] from the [Edit] menu.

3. A confirmation dialog box is displayed. To continue deletion, click on the [Yes] button.

NOTE

- The shortcut key corresponding to [Delete] is [Delete].
- To delete all controls and figures from the screen, use [Select All] in the [Edit] menu.

IMPORTANT

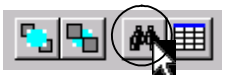
- Different from cutting, deleted controls or figures are not pasted.

9.1.7 Find

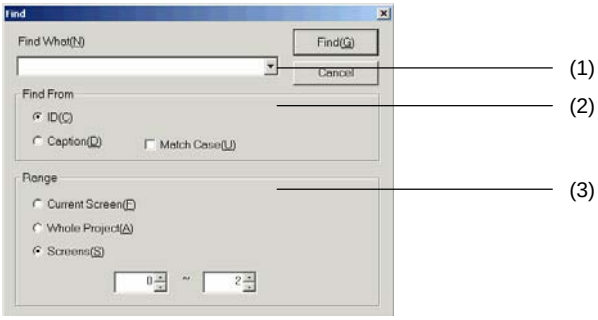
Search for a control with a specific control name or caption having been set.

1. Select [Find] from the [Edit] menu, or select the [Find] button in the tool bar.

Tool bar



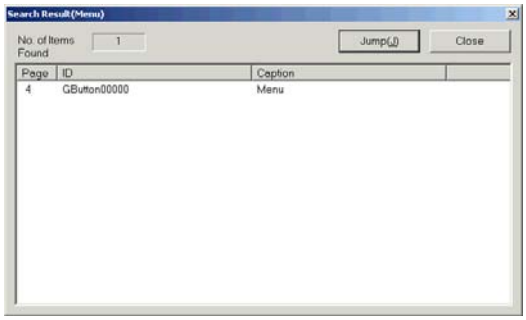
2. The Find dialog box is displayed.



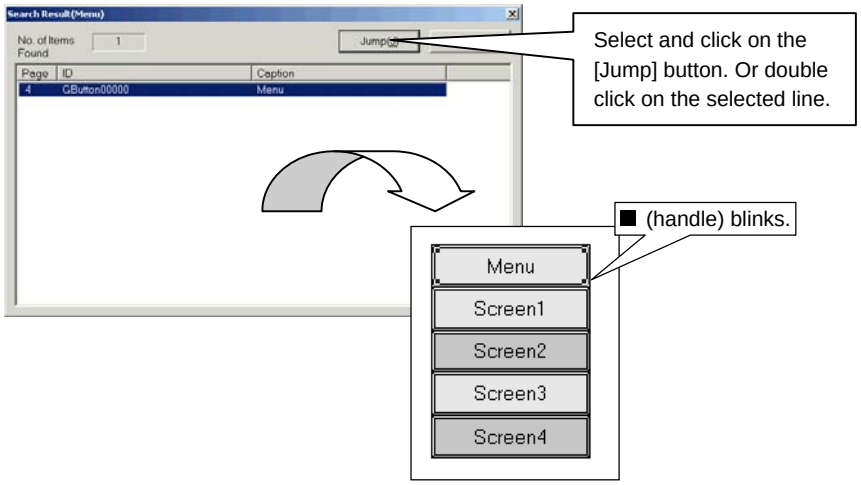
No.	Item	Description
(1)	Search string	Designate the desired control name or caption.
(2)	Target of search	Select either the control name or caption to be searched. Place a check mark at "match case" to search the exact match with the search string.
(3)	Range	Select the search range from the following options.
	Current screen	The current foreground screen is searched.
	Whole project	The whole project is searched.
	Screens	Pages in the designated range are searched.

3. Click on the [Find] button to start to search.

4. When the search is finished, a [Search result list] dialog box is displayed.



5. Select the control in the search result list and click on the [Jump] button, or double click on the selected line.
   The screen including the selected control is displayed, and the control blinks in the selected state.



NOTE

◆ The shortcut key corresponding to [Find] is [Ctrl] + [F].

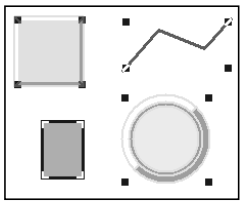
9.1.8 Select All

Use this function to select all objects on the screen or select objects belonging to the same type.

All Objects

Select all objects arranged on the screen.

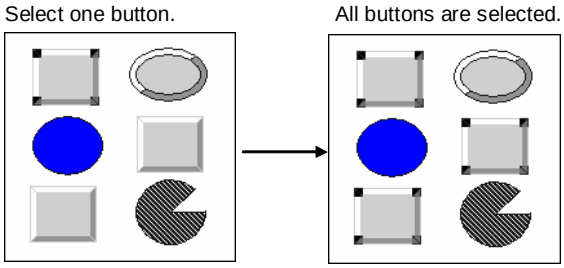
1. From the [Edit] menu, select [Select All] - [All Objects].



Objects Belonging to Same Type

Select all the objects belonging to the same type as that of the selected object.

1. Select a desired object. From the [Edit] menu, select [Select All] - [Same Object Type.]



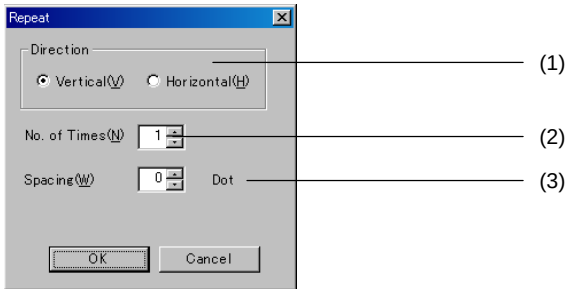
9.1.9 Repeat

Duplicate the selected object horizontally or vertically by the designated number.

1. Select the desired object.
   To duplicate multiple objects simultaneously, select all the desired objects to be duplicated.

2. Select [Repeat] from the [Edit] menu.

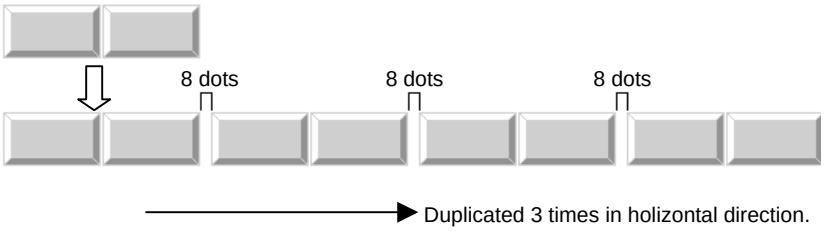
3. The [Repeat] dialog box is displayed.



No.	Item	Description
(1)	Direction	Select either the vertical or horizontal direction in which the object is duplicated. When vertical is selected, the duplicate is placed down. When horizontal is selected, the duplicate is placed on the right side.
(2)	No. of Times	Designate the number of duplicates (1 to 10).
(3)	Spacing	Designate the gap placed between the duplicated object. Specify "0" to place the duplicated object without a gap.

4. After entry is finished, click on the [OK] button.

[To duplicate three times at 8-dot intervals]



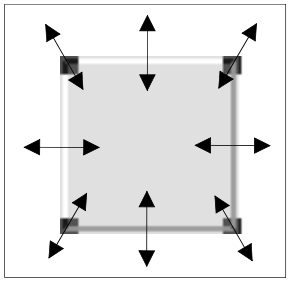
9.2 Layout Function

The layout function for changing the size and position of the object arranged on the screen is described.

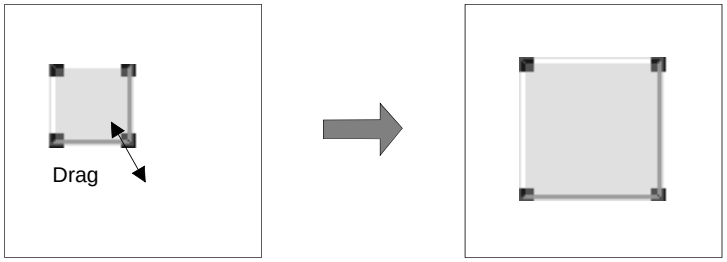
9.2.1 Size Change

1. Select the desired object.

2. Move the cursor to the solid box mark at four corners. The cursor shape changes as shown below.



3. Drag in the arrow direction until the object is deformed to the target size.



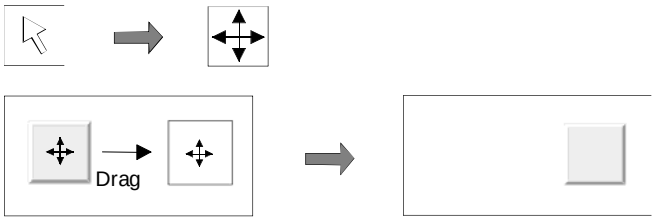
NOTE

- Drag while holding down the [Shift] key to change the size while keeping the original aspect ratio.
- Drag up/down or left/right while holding down the [Ctrl] key to change the size evenly up/down or to left/right.

9.2.2 Move

1. Move the cursor to the desired object.
   To move multiple objects simultaneously, select all the desired objects to be moved.

2. After the cursor shape changes to the one shown below, drag to the desired position.



9.2.3 Arrangement and Alignment

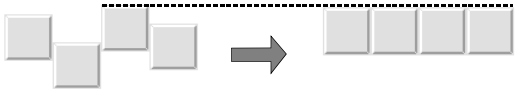
Align multiple objects up, down, left or right, or at even intervals between up/down/left/right.

Example: Aligning to the highest object

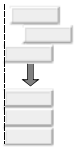
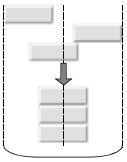
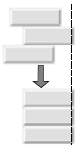
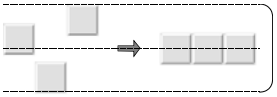
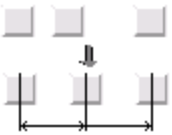
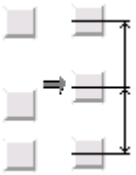
1. Select the objects to be aligned.

2. From the [Layout] menu, select [Align/Distribution] - [Align Top].

3. Align the selected objects along the upper coordinate of the object placed the highest.



Each function is described below.

Function	Description
Align Left	Arrange to the left end. 
Center in a Column	 Arrange to the center in the left/right direction.
Align Right	 Arrange to the right end.
Align Top	 Arrange to the top.
Center in a Row	 Arrange to the center in the vertical direction.
Align Bottom	 Arrange to the bottom.
Distribute Horizontally	 Arrange so that the center points are equally distributed.
Distribute Vertically	 Arrange so that the center points are equally distributed.

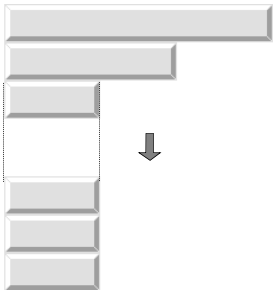
9.2.4 Arrange to Uniform Size

Arrange the width or height of selected multiple objects.

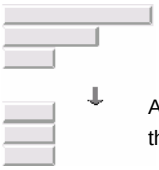
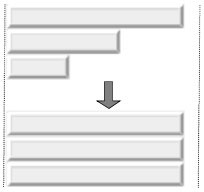
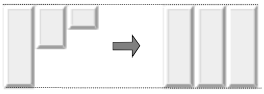
Example: Arranging the size of objects to the narrowest object

1. Select all the desired objects whose width is to be arranged

2. From the [Layout] menu, select [Make Same Size] - [Smallest Width].



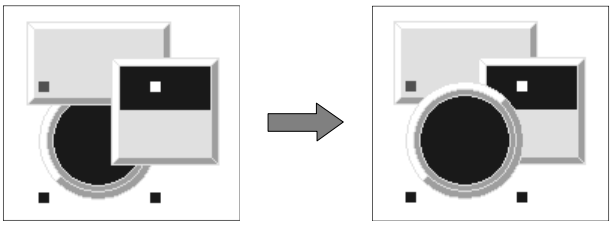
Each function is described below.

Function	Description
Smallest Width	 Arrange to the smallest width.
Largest Width	 Arrange to the largest width.
Smallest Height	 Arrange to the smallest height.
Largest Height	 Arrange to the largest height.

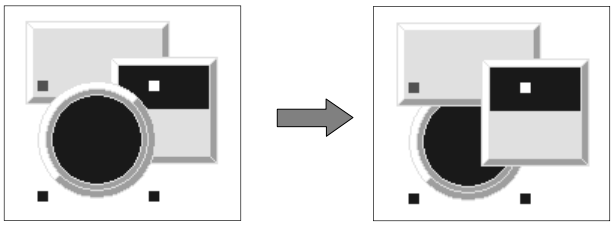
9.2.5 Order

Change the order in which overlapped objects are displayed.

[Bring to front]



[Send to back]



1. Select the desired objects for order change.

2. From the [Layout] menu, select [Order] - [Bring to Front]/[Send to Back] or select [Bring to Front] or [Send to Back] in the tool bar.

9.2.6 Fine Adjustment

Move the selected object up/down or left/right by increments of one dot.
If the grid is valid, the object moves by the set grids.

1. Select the desired object for fine adjustment.

2. From the [Layout] menu, select [Fine Adjustment], and select the desired direction of move.

NOTE

◆ Press the arrow key (→,←, ↑ or ↓) to obtain the same result.

9.2.7 Rotation/Flip

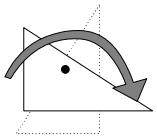
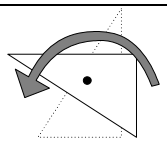
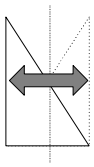
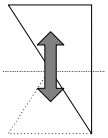
Rotate or flip the object vertically or horizontally. Grouped multiple objects can be rotated or flipped, too.

Rotate/Flip Around the Rectangle of the Object

Rotate or flip the object around the rectangle's center coordinate of the object.

1. Select the object to be rotated or flipped.

2. From the [Layout] menu, select [Rotate/Flip] and the direction of rotation or flip.

Function	Description
Rotate Right 90 Degrees	
Rotate Left 90 Degrees	
Flip Horizontal	
Flip Vertical	

NOTE

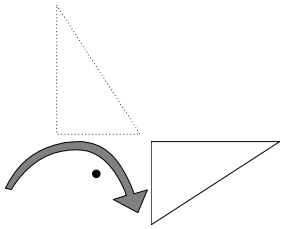
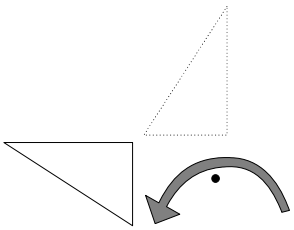
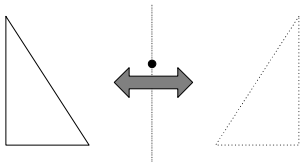
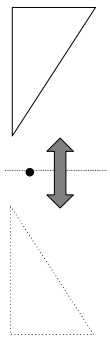
- When grouped objects are rotated or flipped, they rotate or flip around the center of the grouped rectangle.
- The caption character string does not rotate or flip.

Rotate or Flip Around the Center of the Page/View Frame

Rotate or flip the object around the coordinates of the center of the editing page or view frame.

1. Select the object to be rotated or flipped.

2. From the [Layout] menu, select [Rotate/Flip] and select the direction of rotation or flip.

Function	Description
Rotate Right 90 Degrees Around Center of Screen/Frame	
Rotate Left 90 Degrees Around Center of Screen/Frame	
Flip Horizontal Around Center of Screen/Frame	
Flip Vertical Around Center of Screen/Frame	

9.2.8 Deformation

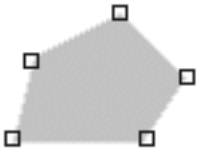
Change the position of the vertex of connected lines, polygon, sector or arc, to change its shape. Or some vertexes can be deleted from or added to the connected lines or polygon.

Editing the Vertex

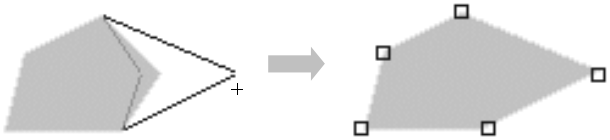
1. Select the desired figure.

2. From the [Layout] menu, select [Modify] - [Edit Node].

3. The vertexes of the figure appear.



4. Move the cursor to the desired vertex and, after the cursor shape has changed to "+," drag the cursor to the new position.



5. Click the right mouse button to exit from the vertex editing mode.

Adding a Vertex

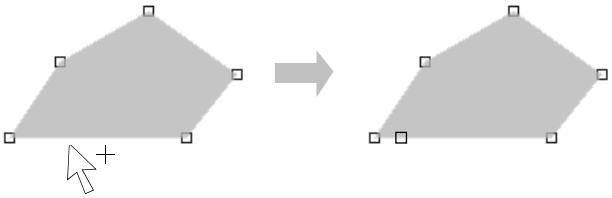
1. Select the desired figure.

2. From the [Layout] menu, select [Modify] - [Add Node].

3. The vertexes of the figure appear.

4. Place the cursor on the contour line. The cursor shape changes as shown below.

5. Click the mouse button at the position of the new vertex on the contour line.



6. Click the right mouse button to exit from the vertex addition mode.

Deleting a Vertex

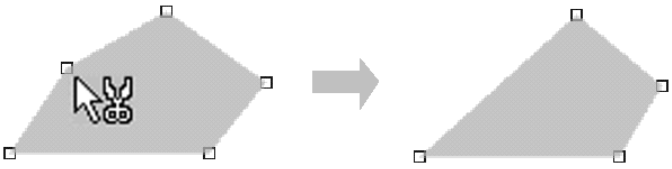
1. Select the desired figure.

2. From the [Layout] menu, select [Modify] - [Delete Node].

3. The vertexes of the figure appear.

4. Move the cursor to the desired vertex. The cursor shape changes as shown below.

5. Click on the desired vertex.



6. Click the right mouse button to exit from the vertex deletion mode.

9.2.9 Grouping and Ungrouping

Grouped multiple controls or figures can be edited or operated as if they are a single object. Grouped controls or figures can be grouped with another group or other controls or figures.

Grouping

1. Select the desired controls or figures to be grouped.

2. From the [Layout] menu, select [Group] - [Group].

NOTE

- The shortcut key corresponding to [Group] is [Ctrl] + [G].
- The view frame cannot be grouped.

Ungrouping

Ungroup to release grouped controls and figures into original objects.

1. Select the desired group to be ungrouped.

2. From the [Layout] menu, select [Group] - [Ungroup].

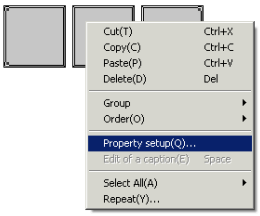
NOTE

- The shortcut key corresponding to [Ungroup] is [Ctrl] + [U].
- Only one group of objects can be ungrouped at a time.

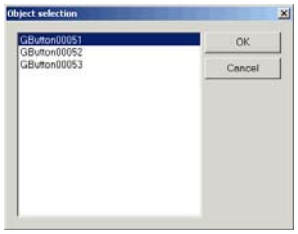
Property Settings of Grouped Objects

Enter the property settings of each object while they are grouped together.

1. Move the mouse cursor to a grouped object, and select [Property Setting] from the popup menu displayed upon a click of the right mouse button.



2. The [Object Selection] dialog box is displayed. Select the desired object and click on the [OK] button. Properties of the selected object are displayed in the property window.

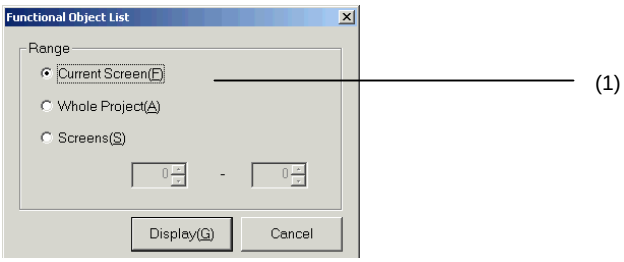


NOTE

- Double click on a desired object in the group while the group is selected, then the properties of the object is displayed in the property window.

9.3 Control List

Settings of the created control are displayed in a list.



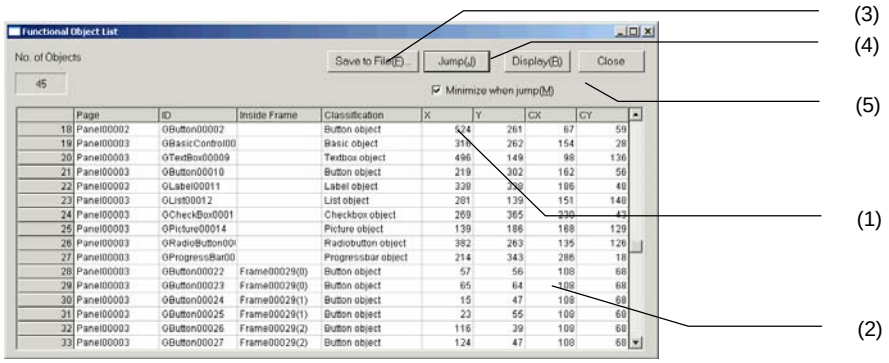
1. From the [Tool] menu, select [Functional Object List].

2. The [Functional Object List] dialog box is displayed.

No.	Item	Description
(1)	Range	Select the range of the list among "Current Screen", "Whole Project" and a range of designated "Screens".

3. After entering the settings, click on the [Display] button.

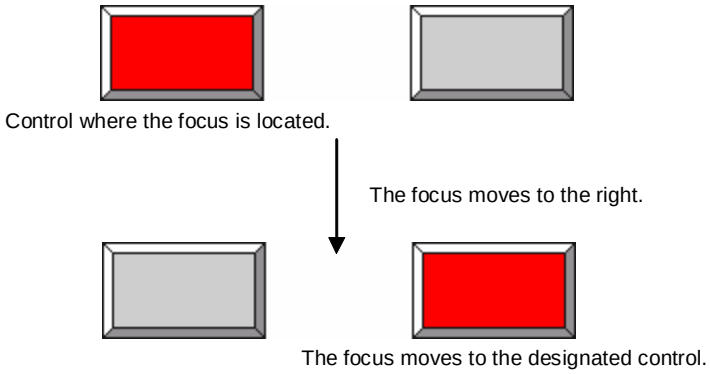
4. The [Functional Object List] window is displayed.



No.	Item	Description
(1)	Property settings	Property settings of each control are displayed.
(2)	Property settings	Property settings of each control are displayed.
(3)	Save to file	Click to output the data in the [functional object list] to the designated file in the designated directory in the CSV format.
(4)	Jump	You can jump to the designated page position of each control. 1. Click on the number part of the desired control to select the entire line.  2. Click on the [Jump] button or double click on the line. 3. The screen containing the selected control is displayed, and the control blinks in the selected mode. 
(5)	Display	To re-designate the range of the list or update the list data, click on the [Display] button in the [functional object list] to refresh.

9.4 Focus Setup

With NC Designer, and the focusing order of each control can be specified.

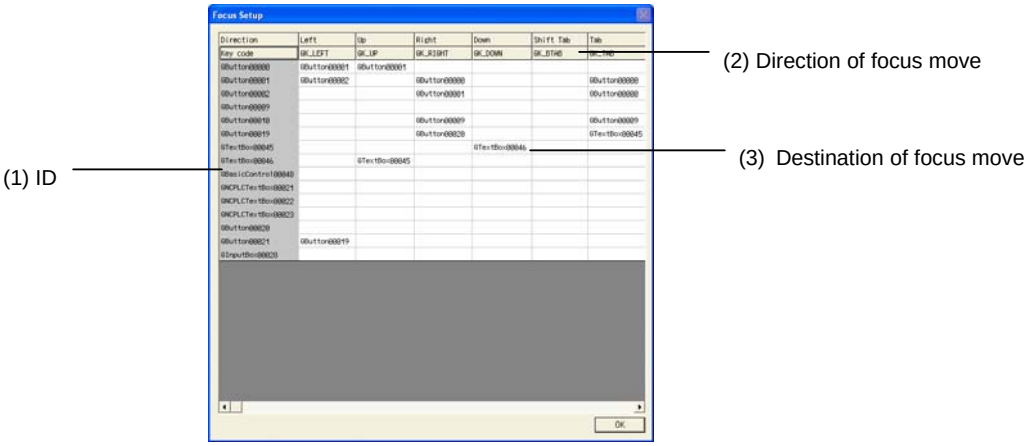


Give the focus setting on each page/frame.

1. Open the desired page/frame.

2. From the [Settings] menu, select [Focus Setup].

3. The [Focus Setup] dialog box is displayed.

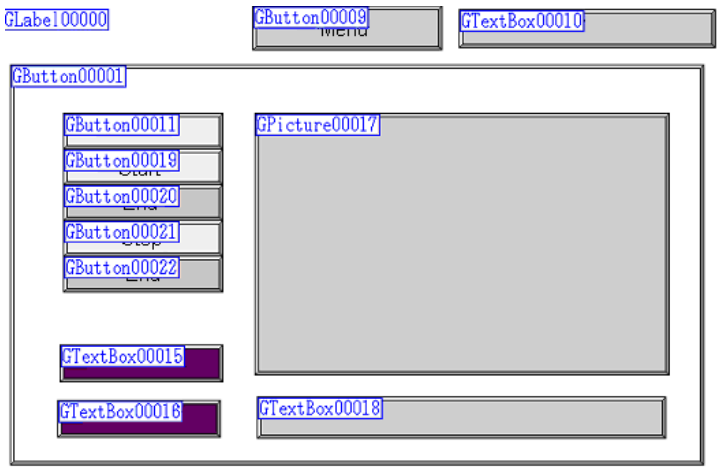


No.	Item	Description
(1)	ID	Display the names of the controls included in the page.
(2)	Direction of focus move	Specify six directions of focus move. Specify the destinations of the focus move after the [GK_LEFT], [GK_UP],[GK_RIGHT], [GK_DOWN], [GK_BTAB] and [GK_TAB] key codes defined in NC Designer are received.
(3)	Destination of focus move	Select the control name, which is the destination of the focus move, from the list. Select "NULL" or specify no data to refrain from moving the focus in the direction. To cancel the destination having been entered, select "NULL."

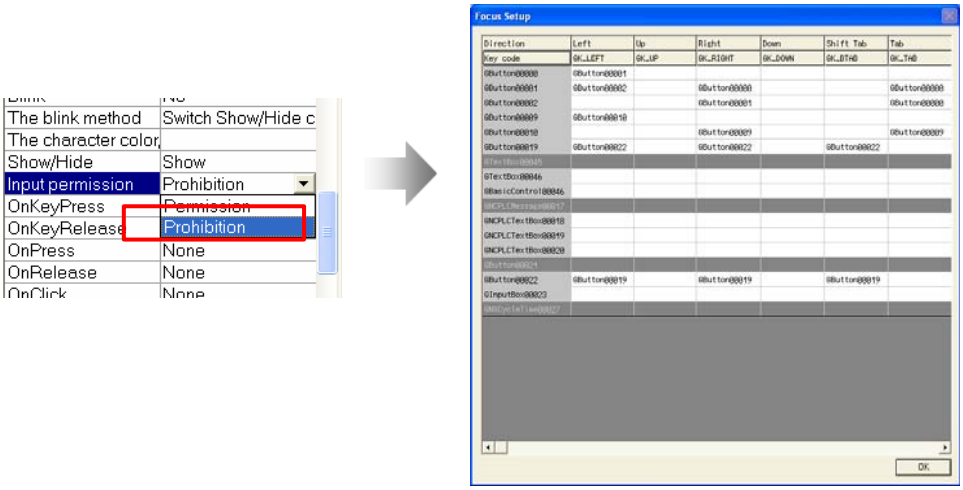
4. After finishing data entry, click on the [OK] button.

NOTE

◆ While the focus setup is given, control names are displayed at the objects located in the page view.



◆ Cells are colored in gray for the initial focus which is set from [Input permission] – [Prohibition] in the property of the control or the control with no "Input permission" in the property settings.

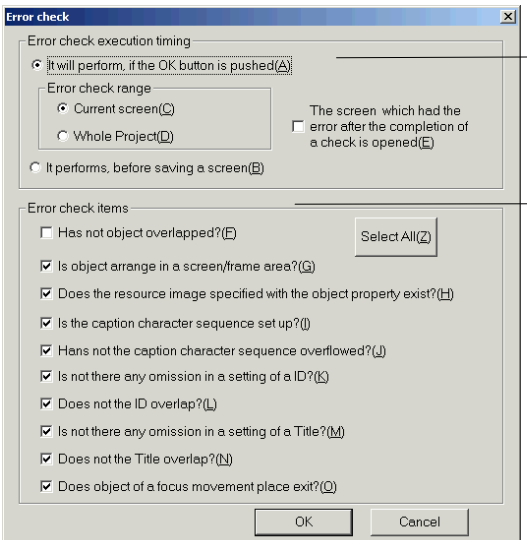


◆ The destination set in the program is given priority over the destination set in [Destination of focus move].

9.5 Error Check

1. From the [Tool] menu, select [Error Check].

2. The [Error Check] dialog box is displayed.
   Enter each item and click on the [OK] button.



No.	Item	Description
(1)	Error check execution timing	Specify the timing and execution range of the error check.
	It will perform if the OK button is pushed	Execute an error check when the [OK] button is clicked on in the [error check] dialog box. Select the range of execution of the error check between "Current Screen" and "Whole Project".
	It performs before saving a screen	Execute an error check upon screen saving operation before the screen is stored. The error check execution range is only the current screen.
	The screen which had the error after the completion of a check is opened	Place a check mark to display the error page after error check.
(2)	Error check items	Place a check mark at desired error check items.

9.5.1 Error Check Item List

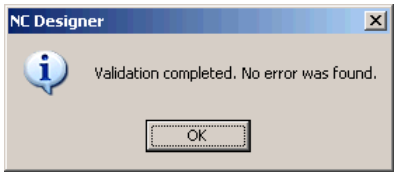
Item	Description
Has not object overlapped?	Checks if controls/view frames overlap.
Is object arranged in a screen/ frame area?	Checks if controls outside the page and controls arranged in the view frame are located outside the view frame.
Does the resource image specified with the object property exist?	Checks if BMP and JPG files registered in the image resources specified for controls exist.
Is the caption character sequence set up?	Checks if caption character strings are deleted from the character string resources designated in each control.
Has not the caption character sequence overflowed?	Checks if caption character strings of each locale overflow the character string area of the control.
Is not there any omission in a setting of an ID?	Checks if controls have a control/view frame name.
Does not the ID overlap?	Checks if control/view frame names are duplicated among multiple controls/view frames.
Is not there any omission in a setting of a Title?	Checks if the panels/windows have a panel/window name.
Does not the Title overlap?	Checks if panel/window names are duplicated among multiple panels/ windows.
Does object of a focus movement place exit?	Checks if control/view frames specified as a destination of the focus exist.

9.5.2 Result of Error Check

The result of the error check is displayed in the message window.

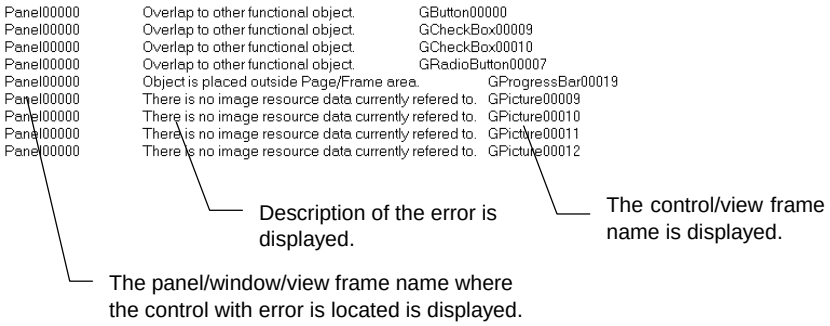
If No Error Is Detected

1. The dialog box shown below is displayed after the error check.



If an Error Is Found

1. The result of the error check is displayed in the message view after the error check.

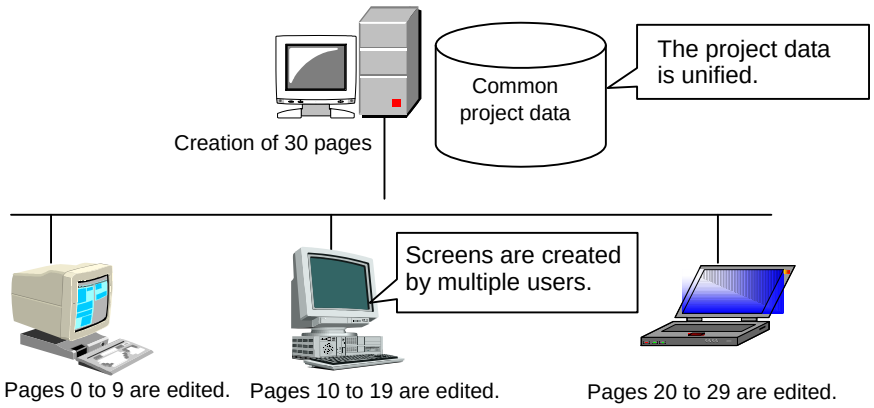


NOTE

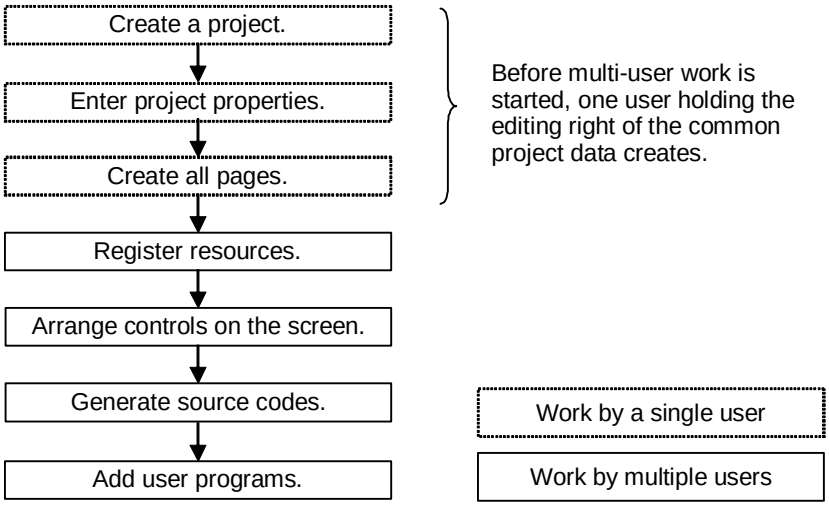
- If 100 or more errors are found, the following message is displayed and the error check is terminated.

9.6 Development by Multiple Users

To create screens by multiple users using NC Designer, common project data such as resource data and panel names must be unified for a management purpose. During multi-user development, no screen duplication is allowed among users. (Shown is an example of creation of 30 pages by 3 users.)



The screen development work is divided into the work that can be distributed to multiple users and the work that must be done by a single user holding the editing right.



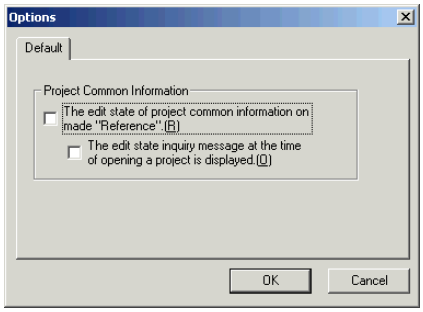
9.6.1 Option Setting

The option setting for multi-user screen creation is described.

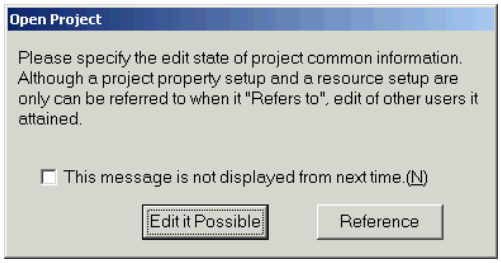
1. From the [Tool] menu, select [Options].

2. The [Options] dialog box is displayed.
   Only one worker who has the editing right can change common project data during multi-user development.
   Other users can refer to common project data.

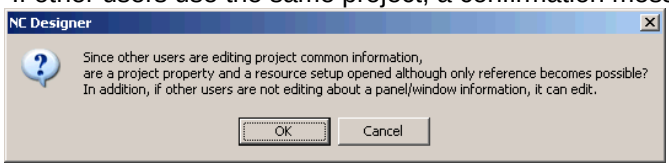
3. If the editing state of common project data is "Reference," panels and windows of the project can be edited while resources may not be changed.



4. If an inquiry message displayed at the project opening is selected, a confirmation dialog box is displayed.

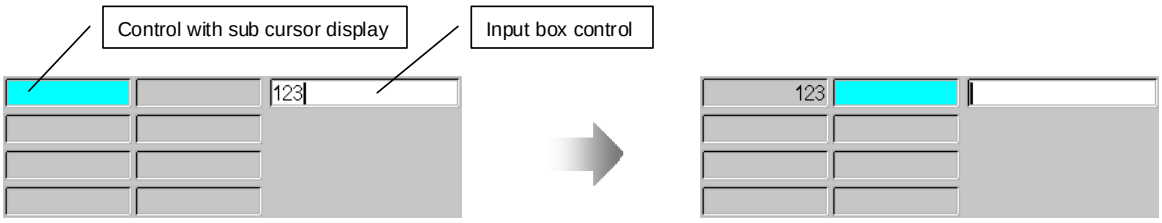


If other users use the same project, a confirmation message is displayed.



9.7 Sub Cursor Setting

The sub cursor setting enables to reflect the values which has been input in the input controls (input box, ten-key) to the controls where the sub cursor is located, by pressing the INPUT key. The display indicating the destination where the input data is to be reflected is called "sub cursor". The following functions are also available by sub cursor setting.



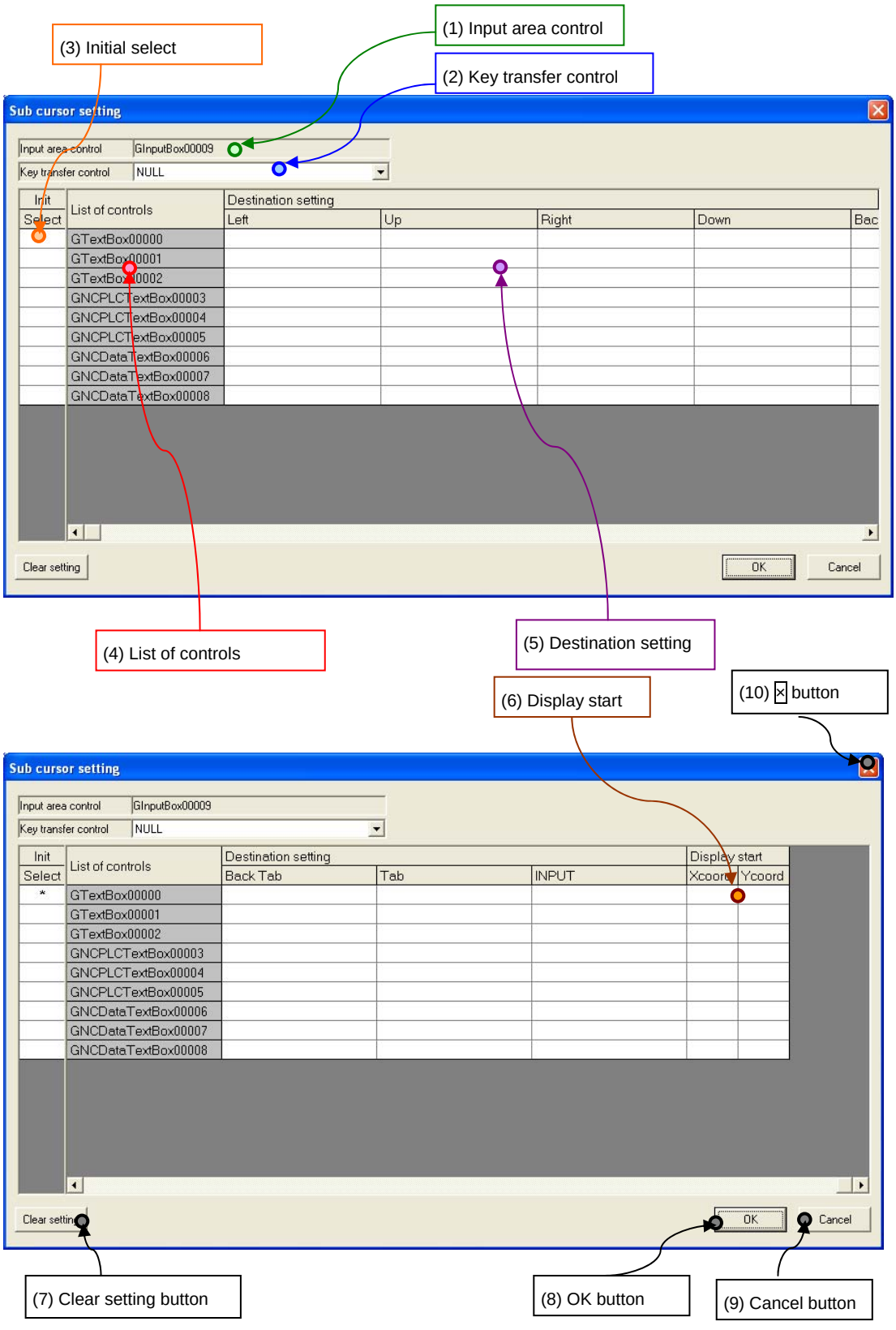
(1) Displays the sub cursor (Specify the sub cursor position at default)
(2) Moves the sub cursor with the arrow key/TAB key or a click
(3) Sets the key transfer control
(4) Moves the display location of the input area control

The Sub cursor setting dialogue will appear by selecting the pop-up menu [Extension setting]-[Sub cursor setting] which will appear by clicking the right mouse button on an input control.
For the details of the input box, refer to "7.2.14 Input Box (GInputBox)".
For the details of the ten-key, refer to "7.2.15 Ten-key(GSoftKey)".

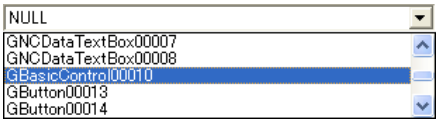
9.7.1 Screen Specifications

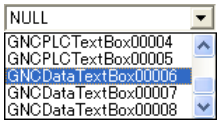
Screen Images

The Sub cursor setting screen is constructed as shown below.



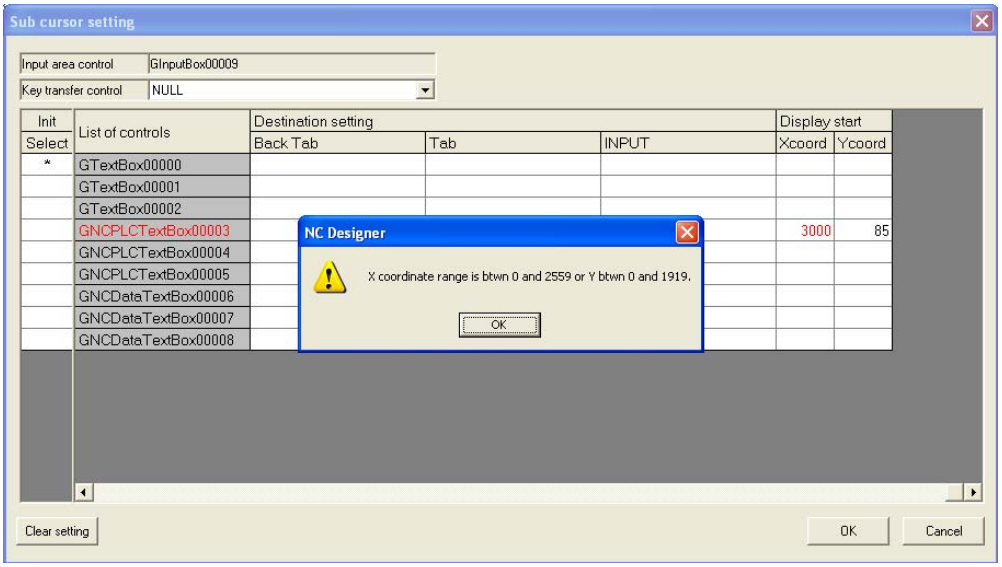
Displayed item

No.	Displayed item	Details
(1)	Input area control	The name of the control to make the "sub cursor setting" will be displayed. (Note 1) Input controls include the input box and the ten-key.
(2)	Key transfer control	Set this when processing keys other than those which are allowed to use in input controls (alphanumeric characters/arrow keys, etc.). When the key transfer control is set, the focus will move to the set key transfer control and key will move in the control. When "NULL" is set, key will move. Select a key transfer control from the list of controls which will appear by clicking the area.  The controls in the list are those located in the currently editing page (panel, window)/view frame. (Note 1) Input controls (input box, ten-key) will not be included in the list. (Note 2) For the details of the input box, refer to "7.2.14 Input Box (GInputBox)". For the details of the ten-key, refer to "7.2.15 Ten-key (GSoftKey)".
(3)	Initial select	Specify the control where the sub cursor will be located at first when the focus shifts to an input control. Click a cell to display "*" and the cell will be set as the control where the sub cursor will be located at first. If another cell without "*" is clicked, the "*" in the original cell will be cleared. (Note 1) If the control specified as where the sub cursor is to be located at first is deleted, "*" will appear at the top control of "(4) List of controls" and will be set instead. (Note 2) If any control specified as where the sub cursor is to be located at first is not set after pressing the [Clear setting] button, the top control of "(4) List of controls" will be set as the control.
(4)	List of controls	The list of controls available for sub cursor display in the page (panel, window) or the view frame where the input area control is located, will be displayed. The target controls are the text box, PLC text box, and NC data text box. (Note 1) For the input controls located on a view frame, only the controls located on the view frame will be listed. Also, when the view frame is included on the page and an input control located outside of the view frame is specified, the controls on the view frame will not be listed.

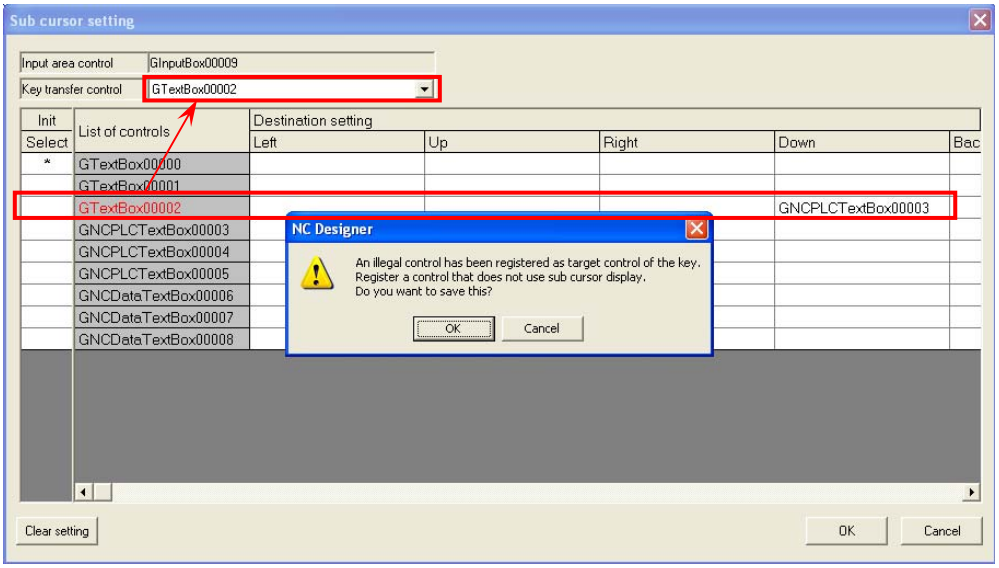
No.	Displayed item	Details
(5)	Destination setting	Set the destination control of sub cursor when an arrow key (←, ↑, →, ↓), a TAB key (|←, →|), or the INPUT key is pressed. When unset or when "NULL" is set, the sub cursor will not move. Select the destination control from the list of controls which appears by clicking the cell.  The listed controls are the same as "(4) List of controls". (Note 1) The control cannot be input by copy & paste. (Note 2) If a control is set as the destination of arrow keys (←, →), the cursor will not move in the input area control by arrow keys (←, →).
(6)	Display start	Set the display position of the input area control for each control where the sub cursor will be displayed. When no setting is made, it will be displayed at where it was pasted on the NC Designer. Set the display position (from the upper left of the page (panel/window)/view frame) of the input area control in dots. X coordinate setting range: 0 to 2559, Y coordinate setting range: 0 to 1919
(7)	Clear setting button	All the settings ((2),(3),(5),(6)) will be cleared.
(8)	OK button	The dialogue will be closed after saving the settings.
(9)	Cancel button	The dialogue will be closed after discarding the settings.
(10)	× button	The dialogue will be closed after discarding the settings.

NOTE

- The column width of "List of controls", "Key transfer control", "Display start" can be changed.
- If the display start position (X coordinate/Y coordinate) is out of the setting range or if either of the X/Y coordinate is not set, an error message will appear when [OK] button is clicked.
  For the details of error messages, refer to "Appendix 1. Error Message List".
  If an error occurs, the control name on the List of controls will be displayed in red and the character color or background color of the corresponding area will also turn red.



- If a control which is set to use the sub cursor display is registered in the key transfer control and click [OK] button, the following message will appear.
  The name of the control set in the key transfer control will turn red.



Click the [OK] button to save the setting and close the sub cursor setting screen.
Click the [Cancel] button to close the save confirmation message and return to the sub cursor setting screen.

If a control which is set to use the sub cursor display is registered in the key transfer control as above-mentioned, the key transfer may repeat endlessly.

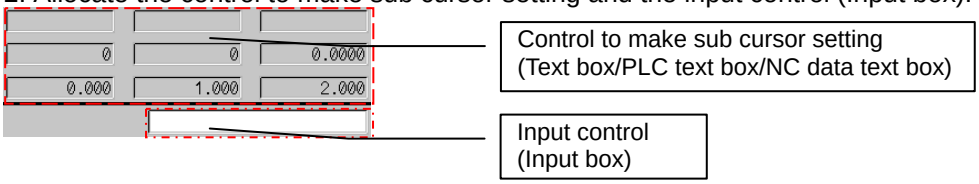
9.7.2 Sub cursor setting screen displaying method

How to call

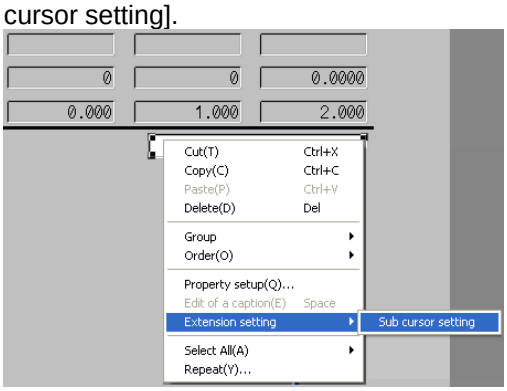
The sub cursor setting can be made for each input control allocated to each page (panel/window) or view frame.

1. Display the page or frame to make settings.

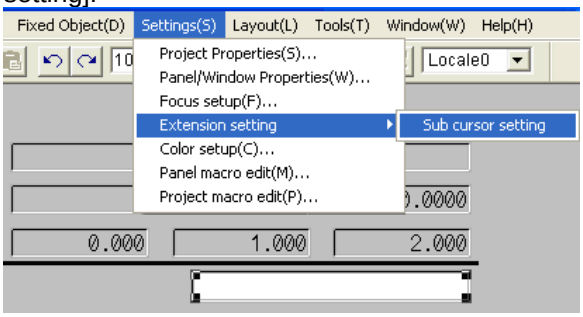
2. Allocate the control to make sub cursor setting and the input control (Input box).



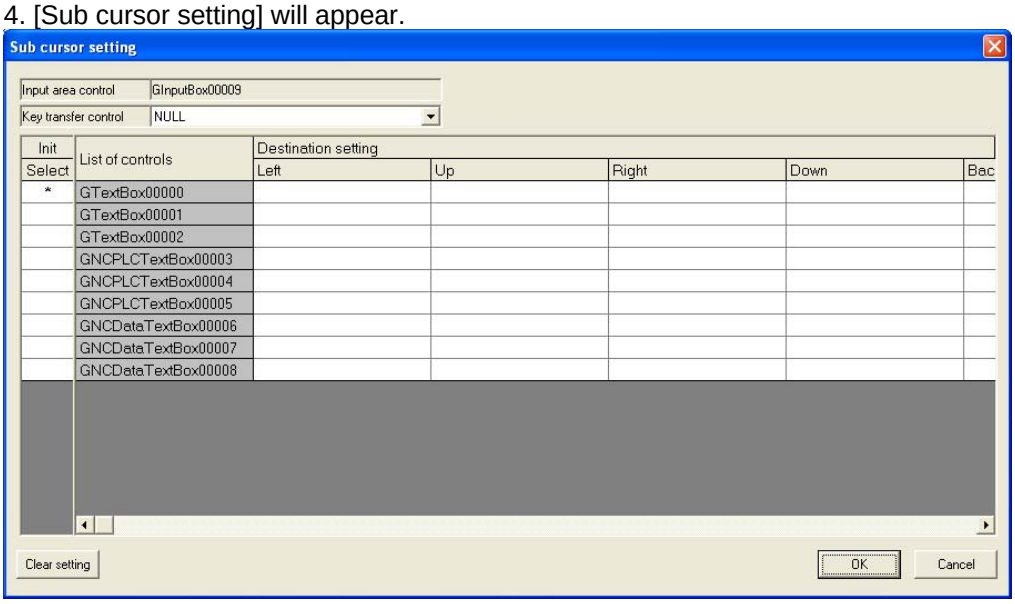
3. Right-click the input control to display the pop-up menu and select [Extension setting] - [Sub cursor setting].



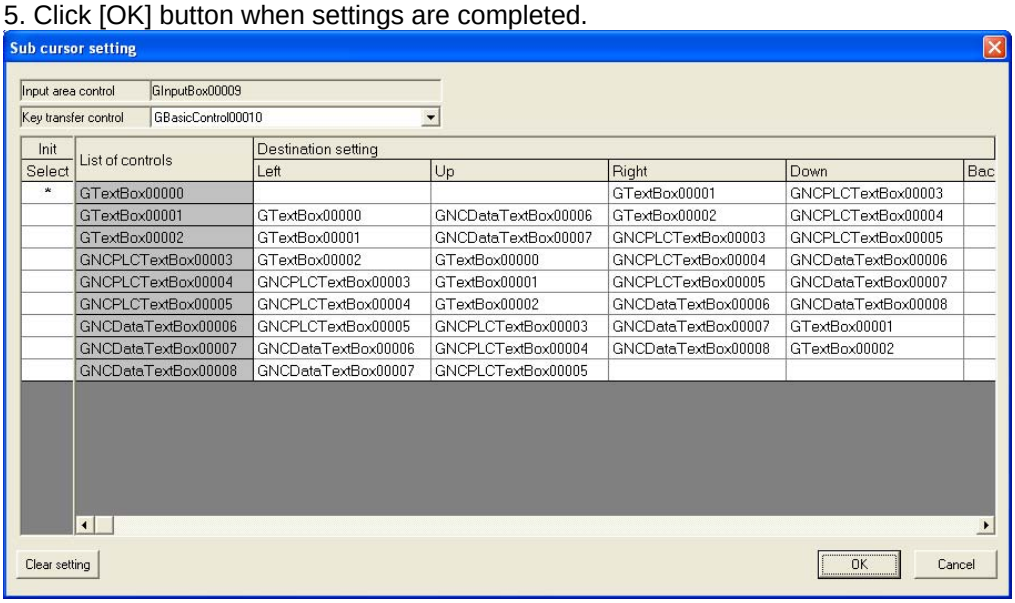
The access can also be made from the menu bar [Settings] - [Extension setting] - [Sub cursor setting].



4. [Sub cursor setting] will appear.

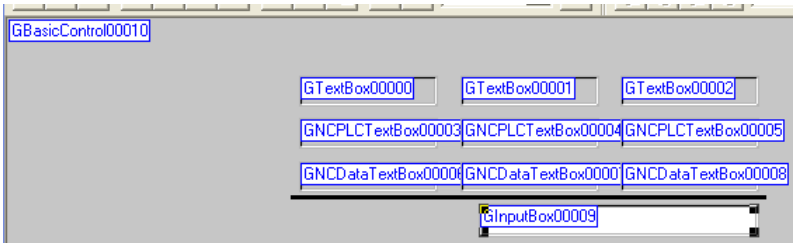


5. Click [OK] button when settings are completed.

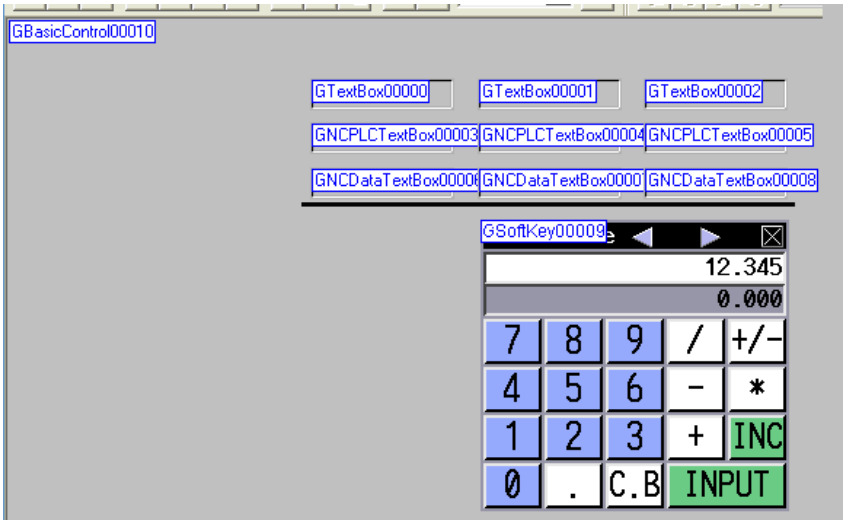


NOTE

◆ During the Sub cursor setting, the objects of page view will display the name of controls.



◆ When the ten-key is used instead of the input box, the display will be as shown below.



9.7.3 Sub cursor setting procedure

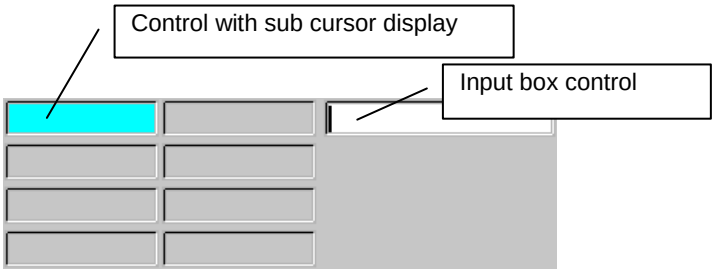
Sub cursor setting procedure does not differ between the input box control and the ten-key control. The following explanation uses the input box control.

9.7.3.1 Move the sub cursor by key input (arrow key, TAB key, input key)

With the focus on the input control (input box control), press [←], [↑], [→], [↓], [|←], [→|], [INPUT] key to move the sub cursor.

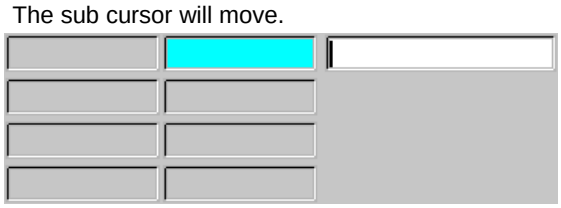
(1)

The control with the sub cursor display is located as shown in the right figure.

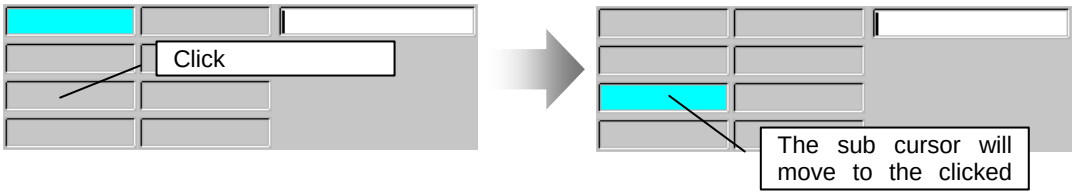


(2)

Press [→].



The sub cursor will also move by clicking the control to which the sub cursor setting is made.



NOTE

- The background color and the character color of the control in which the sub cursor will be displayed depend on the property setting of the input box control; "Sub cursor Background color" and "Sub cursor Character color".
- The sub cursor will not be displayed until the focus is placed on the input box or the ten-key control.
- The sub cursor will not move when the destination is not set.
- The right and left arrow keys (←, →) are used to move the cursor within the input destination control. But instead, when the right and left arrow keys (←, →) are set to move the sub cursor to the destination control, the cursor will move to the specified control.

Set the focus of input control

Interpreter method

```
$GInputBox00006-OnCreate
'Set the input box control as the focus point at default
GCSChangeActiveFocus(-1, "GInputBox00006");
$End
```

Compilation method

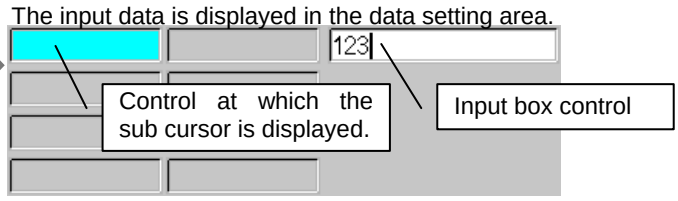
```
long GCPanel00000::GINPUTBOX00006OnCreate(unsigned short usMessage, long lLParam, long
lUParam)
{
	GBaseObject *pPanel = NULL;
	GBaseObject *pChild = NULL;
	pPanel = GetGBaseObject();
	pChild = GCSGetChild( pPanel, GINPUTBOX00006 ); // Get the input box control
	if (pChild != NULL) {				// When it succeeds to get the input box control
	GCSChangeActiveFocus( pPanel, pChild );		// Set the focus at the input box control
	}
	return TRUE;
}
```

9.7.3.2 Set the input value to the target control

With the focus on the input control (input box control), press the INPUT key to set the input value to the control at which the sub cursor is displayed.

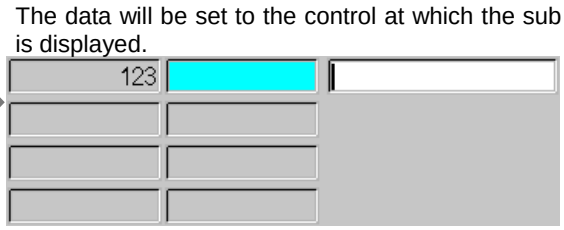
(1)

Input data keys.
1 2 3



(2)

Press the INPUT key.



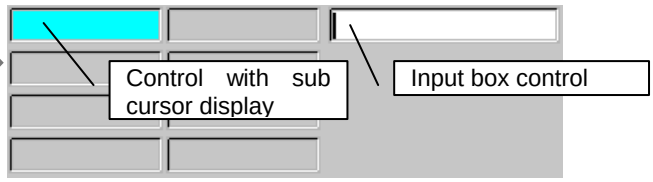
(NOTE) When the setting fails, the content of the input control (input box control) will not be cleared and (even when the destination for the INPUT key is set,) the sub cursor will not move.

9.7.3.3 Change the display start position

The display position of input control can be changed for each control with sub cursor display by setting the display position of input control (input box control).

(1)

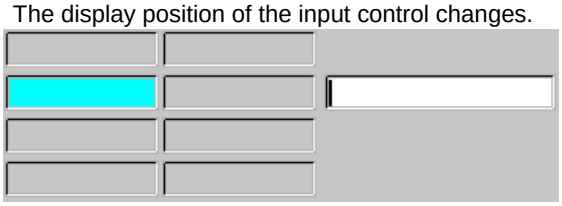
The target control and input box are located as shown in the right.



(2)

Press the ⇩ key.

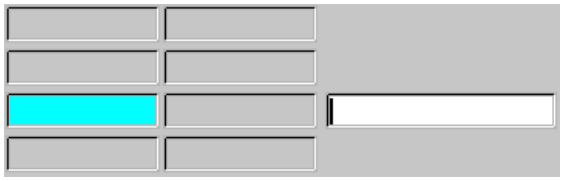
The display position of the input control changes.



(3)

Press the ⇩ key.

The display position of the input control changes.

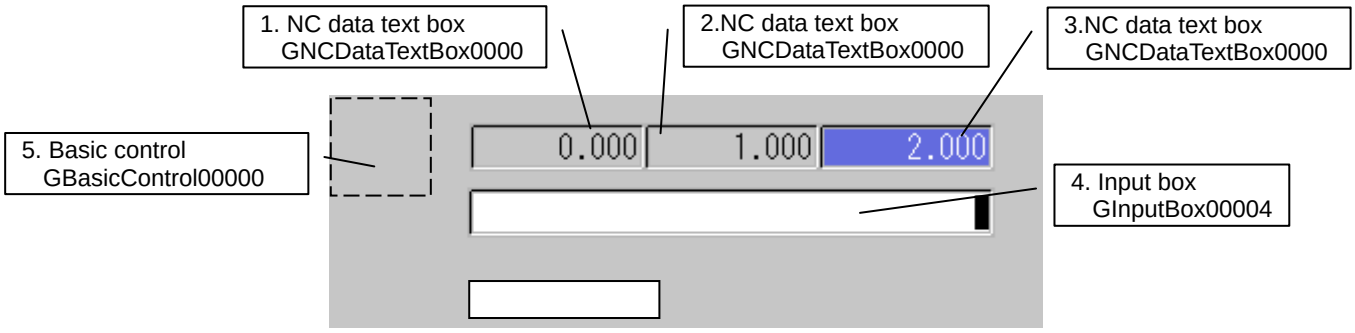


9.7.3.4 Transfer a key to other control

Set the key transfer control to use other control to handle the processing of a key other than those available for the input destination control (alphanumeric characters/arrow keys, etc.).
When the key transfer control is set, the focus will move to the transfer destination control and the input key will be transferred.
In the following example, the focus will move to the basic control and the input key will be transferred to the control.

Screen configuration

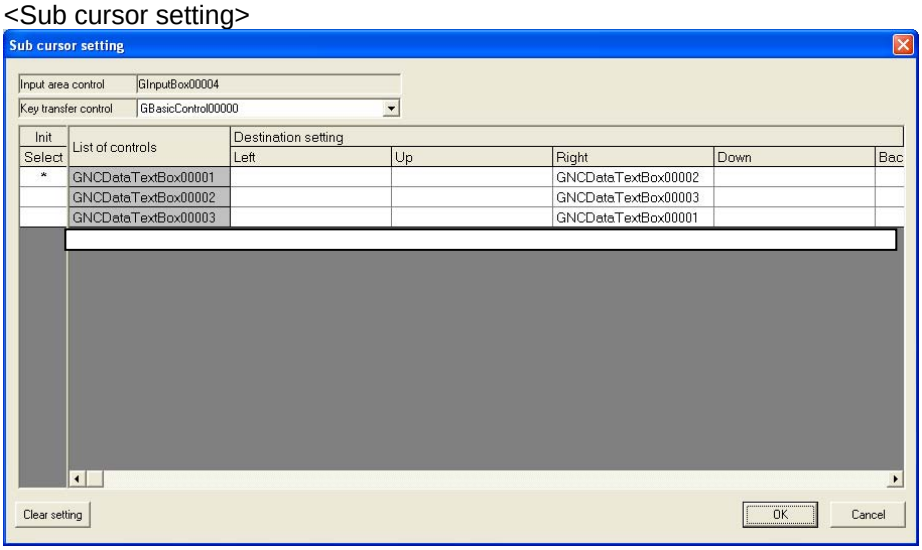
The key which cannot be handled by the input box (function key) is transferred to the basic control where the key processing (OnKeyPress) will be carried out.



Operation

(1) The focus is placed at the basic control when the screen opens.
(2) Mouse-click the NC data text box to display the sub cursor in it and transfer the focus to the input box.
(3) Press the cursor key (→) to move the sub cursor.
(4) Press "MONITOR" key (Shift+F1) to go to the Monitor screen.
(5) Press "SET UP" key (Shift+F2) to go to the Setup screen.
(6) Press "EDIT" key (Shift+F3) to go to the Edit screen.
(7) Press "DIAGN" key (Shift+F4) to go to the Diagnosis screen.
(8) Press "MAINTE" key (Shift+F5) to go to the Maintenance screen.

<Sub cursor setting>



Source code

Interpreter Method

```
$GBasicControl00000-OnCreate
'Set the default focus position at the basic control.
GCSChangeActiveFocus(-1, " GBasicControl00000");
$End

$GBasicControl00000-OnKeyPress
long _lShiftKey; ‘Shift key input status
_lShiftKey = LUPARAM & H1;
if((LLPARAM == 112) && (_lShiftKey == 1))          'When Shift + F1 key code is issued.
GCSGEShowPanel(1000);                               'Changing the screen to Monitor screen.
elseif((LLPARAM == 113) && (_lShiftKey == 1))      'When Shift + F2 key code is issued.
GCSGEShowPanel(2000);                               'Changing the screen to Setup screen.
elseif((LLPARAM == 114) && (_lShiftKey == 1))      'When Shift + F3 key code is issued.
GCSGEShowPanel(3000);                               'Changing the screen to Edit screen.
elseif((LLPARAM == 115) && (_lShiftKey == 1))      'When Shift + F4 key code is issued.
GCSGEShowPanel(4000);                               'Changing the screen to Diagnosis screen.
elseif((LLPARAM == 116) && (_lShiftKey == 1))      'When Shift + F5 key code is issued.
GCSGEShowPanel(5000);                              'Changing the screen to Maintenance screen.
endif;
$End
```

Compilation method

```
#define KEY_SHIFT    0x01
#define GK_F1          112
#define GK_F2          113
#define GK_F3          114
#define GK_F4          115
#define GK_F5          116

long GCPanel00000::GBASICCONTROL00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
GBaseObject *pPanel = NULL;
GBaseObject *pChild = NULL;
pPanel = GetGBaseObject();
pChild = GCSGetChild( pPanel, GBASICCONTROL00000);    // Get the basic control
if (pChild != NULL) {                                 // When getting the basic control succeeds.
GCSChangeActiveFocus( pPanel, pChild );               // Set the focus at the basic control.
   }
return TRUE;
}

long GCPanel00000::GBASICCONTROL00000OnKeyPress (unsigned short usMessage, long lLParam, long lUParam)
{
if((lUParam & KEY_SHIFT) == KEY_SHIFT)
   {
      if(lLParam == GK_F1)                            // When Shift + F1 key code is issued.
      {
// Changing the screen to Monitor screen.
         GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                  GCSGetScreen(GetGBaseObject()), 1000, 0), FALSE);
      }
      else if(lLParam == GK_F2)                       // When Shift + F2 key code is issued.
      {
// Changing the screen to Setup screen.
         GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                  GCSGetScreen(GetGBaseObject()), 2000, 0), FALSE);
      }
      else if(lLParam == GK_F3)                       // When Shift + F3 key code is issued.
      {
// Changing the screen to Edit screen.
         GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                  GCSGetScreen(GetGBaseObject()), 3000, 0), FALSE);
      }
      else if(lLParam == GK_F4)                       // When Shift + F4 key code is issued.
      {
// Changing the screen to Diagnosis screen.
         GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                  GCSGetScreen(GetGBaseObject()), 4000, 0), FALSE);
      }
```

```
        else if(lLParam == GK_F5)                    // When Shift + F5 key code is issued.
        {
// Changing the screen to Maintenance screen.
            GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                        GCSGetScreen(GetGBaseObject()), 5000, 0), FALSE);
        }
    }
}
```

9.7.4 Limitations

(1) If the sub cursor setting is made to the control at which the focus is set, the focus will not move by the key input (cursor key, input key) and the sub cursor will move instead.
(2) After deleting the control to which the sub cursor display was set, press the [Clear setting] button on the sub cursor setting screen and set the sub cursor display again.
(3) When a control to which the sub cursor setting is made is registered as the key transfer destination control, the key transfer may be repeated endlessly.

10. Simulation

The simulation method is described in this section.

10.1 Simulation Function

The simulation function is a function for testing actions of drawn data on NC Designer.
The simulation function allows you to test the state change of controls, page switching, and execution of callback functions, and the following items can be checked.

- Appearance of created panels and windows
- State changes according to value change of control, focus yes/no, show/hide, input permission/prohibition
- Focus move
- Execution timing of callback function displayed in message window
- Panel/window page switching

10.1.1 Starting Simulation

1. Before starting simulation, save the project and screens.

2. From the [Tool] menu, select [Simulation].

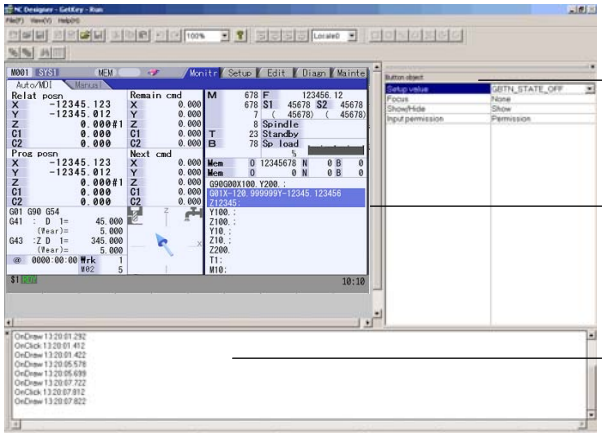
3. Simulation begins. Simulation begins from the first page of panels/windows having been created.

NOTE

- If [Simulation] is selected from the [Tool] menu without saving the project or window, a dialog box is displayed to urge to save data. Before starting simulation, save data.
- Only one panel/window can be checked during simulation. Simultaneous view of a panel and a window is impossible.
- When simulation is started, the currently edited locale is displayed.

10.1.2 Simulation Screen

When simulation is executed, a simulation view and simulation tools are displayed.
The simulation screen at a startup is the first page of panels/windows having been created.



Simulation tool allows you to change the state of the selected control.

The screen being simulated is displayed.

The message window displays the callback execution history.

Screen Selection

Select [Open Panel/Window] from the [File] menu and select and display the new screen in the [Open Panel/Window] dialog box.

10.1.3 Function List

The following menu functions can be used during simulation.

File

Item	Function
Open Panel/Window	Select the panel/window to be displayed.
Quit	Terminate simulation.

View

Item	Function
Test tools	Select whether the simulation tools are displayed or hidden.
Message Window	Select whether the message window is displayed or hidden.
Switch Locale	Switch the locale to be displayed.
Zoom	Specify the zoom scale of the page.

Help

Item	Function
About NC Designer	Display the version of NC Designer.

NOTE

- When simulation is started, the screen is displayed at the zoom specified at [Display magnification] in [Project Properties].
- If the zoom is changed during simulation, the new value is reflected on the [Display magnification] setting in [Project Properties].
- If [Fit] is selected as a zoom setting, the zoom changes in the range between 25 and 800% to fit the simulation screen size.

10.1.4 Quitting Simulation

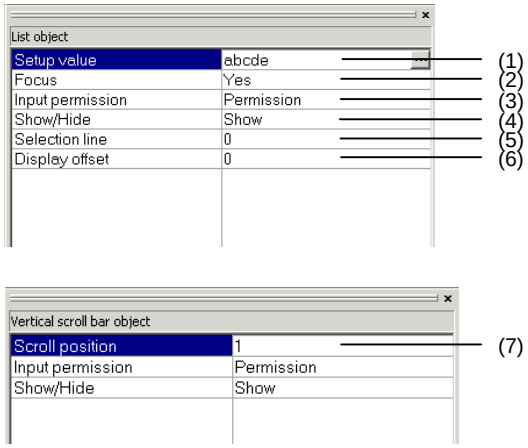
The simulation screen is terminated.

1. From the [File] menu, select [Quit].

10.2 Simulation Tools

Use simulation tools to change values or appearance of controls.

1. Click on a desired control. Settings are displayed at the simulation tool. The settings vary according to the selected control.

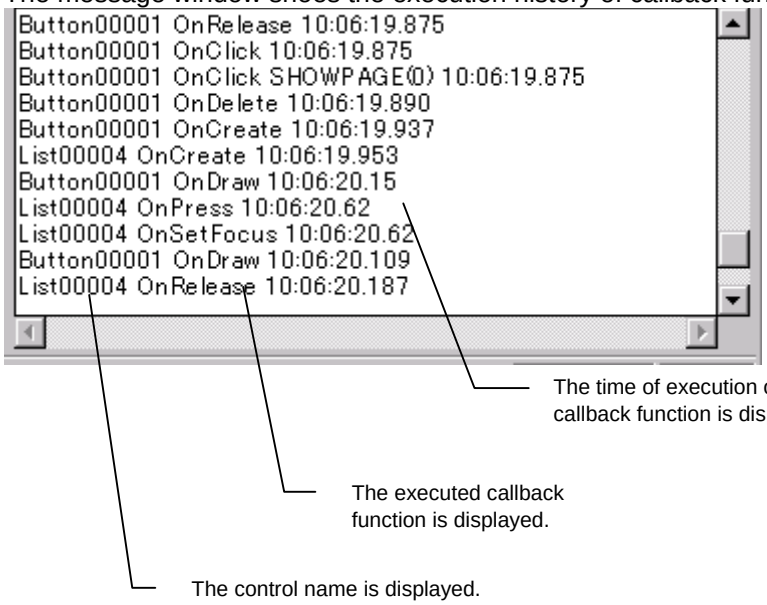


No.	Item	Description
(1)	Setup value	The entered value or character string is reflected on the control.
(2)	Focus	The focus state is displayed.
(3)	Input permission	Input permission and prohibition are switched over. Select prohibition to change to the image of the disabled control.
(4)	Show/Hide	The control is displayed or hidden.
(5)	Selection line	The designated line is selected.
(6)	Display offset	The designated line is displayed at the top.
(7)	Scroll position	Enter a value in the range from the minimum to maximum value of the scroll. The scroll bar moves according to the entered value.

2. Changes in the setting are reflected on the control.

10.3 Message Window

The message window shoes the execution history of callback functions.



11. Generating a Document

NC Designer is provided with a document creation function for outputting project settings and control properties into a rich text format file (hereinafter referred to as RTF file).
The document creation function is described in this section.

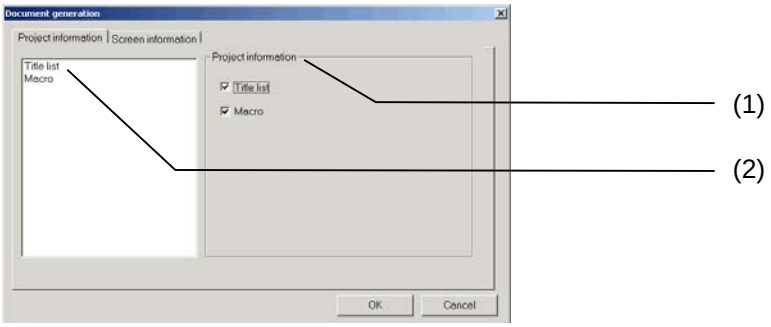
11.1 Document Generation Function

NC Designer can output project settings and control properties into an RTF file.
The items that can output in the file include the followings.

Data		Description
Project information		Project name, project macro and panel/window name list
Screen information		4 items of each panel/window specified below
	Screen hard copy	Hard copy of each panel/window
	Object list	List of controls/view frames used in each panel/window
	Property setup	Property settings of each control/view frame arranged on each panel/window
Macro		Screen macro specified for each panel/window

11.1.1 Generating a Project Information Document

1. From the [File] menu, select [Document Generation].
2. The [Document generation] dialog box is displayed. Select the [Project information] tab.



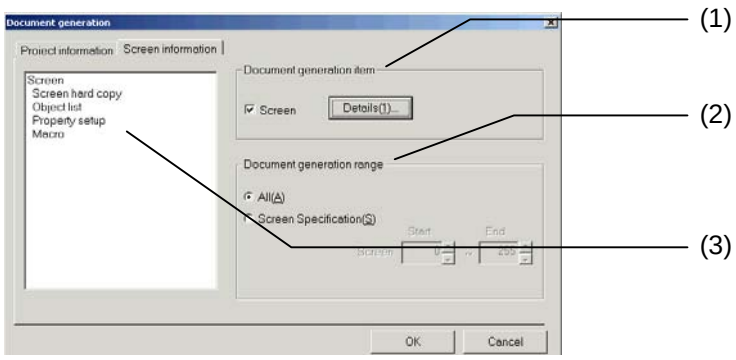
No.	Item	Description
(1)	Output item	Place a check mark on the items to be file-output. The output items include the followings. ・Panel/window name list ・Macro
(2)	Output item display field	A list of file-output data selected at (1) is displayed.

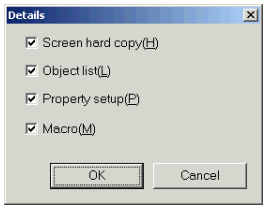
3. Place a check mark at the data to be file-output.
4. Click on the [OK] button. A [Save As] dialog box is displayed.

   A file is output after entering the name of the RTF file.

11.1.2 Generating a Screen Information Document

1. From the [File] menu, select [Document Generation].
2. The [Document Generation] dialog box is displayed. Select the [Screen Information] tab.



No.	Item	Description
(1)	Document generation item	Place a check mark for the item included in the output file. Click on the [Details] button and select the items to be file-output in the dialog box shown below.  The items that can be selected include the followings. ・Screen hard copy ・Object list ・Property setup ・Macro
(2)	Document generation range	Select the page range of the output file among the following options.
	All	Output the data about all panels/windows.
	Screen Specification	Output the data of the panels/windows in the designated page range.
(3)	Output item display field	A list of file-output data selected at (1) is displayed.

3. Place a check mark at the data to be file-output.
4. Click on the [OK] button. A [Save As] dialog box is displayed.

   A file is output after entering the name of the RTF file.

11.1.3 Output Image

The output images are shown below.

Project Information

Project name:test Title:

Panel/Window name list

Page0:Panel00000
Page1:Panel00001
Page2:Window00002

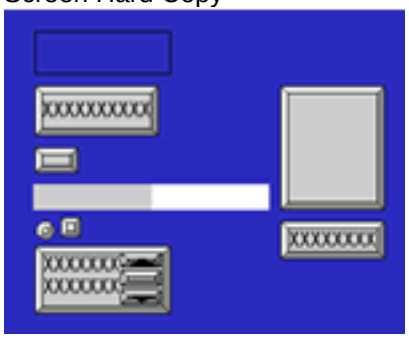
Project macro

```
$Project-OnCycle
GMEM mem;
mem=GMEMCREATE("memetest",1234);
GMEMsetshort(mem,0,0);
...
```

Screen Information

(1) Screen Hard Copy

Screen Hard Copy



(2) Object List

Functional object list

[Button object]
ID=GButton00015
ID=GButton00016
[Label object]
ID=GLabel00017
[Textbox object]
ID=GTextBox01

(3) Property Setup

```
Property setup
================================================================================
ID=GBasicControl00001
X=114
Y=307
WIDTH=115
HEIGHT=112
Show/Hide=Show
Input permission=Permission
OnKeyPress=None
OnKeyRelease=None
OnPress=None
OnRelease=None
OnClick=None
OnDraw=None
OnTimer=None
OnSetFocus=None
OnKillFocus=None
OnCreate=None
OnDelete=None
OnUser=None
================================================================================
```

(4) Page Macro

```
Page macro
$DATETIME 2004/08/19 01:47.01
$GBasicControl00001-OnCreate
GMEM mem;
mem_GMEMCREATE("memetest",1234);
GMEMsetshort(mem,0,0);
$End
$GBasicControl00001-OnClick
short a;
GMEM mem;
mem=GMEMselect("memetest");
a=GMEMgetshort(MEM,0);
if(a==0)
     GCSSetString(-1,"GTextBox00001","000");
elseif(a==1)
     GCSSetString(-1,"GTextBox00001","001");
endif
$End
```

(5) Top of Each Page

The page number and the panel/window name are output at the top.

```
Page number:  0  Panel/Window name:Panel00000
```

NOTE

- If an object list or property setup is output in a file, information about the view frame is also output in the file in addition to the objects.
- For the screen hard copy and property setup, the currently displayed locale data is output in the file.

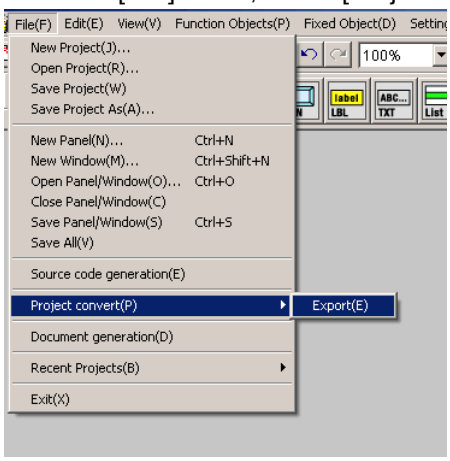
12. Project Convert

The method for exporting or importing the project created with NC Designer is described in this section.
Exportation is necessary when the project is executed in the interpreter mode or when the macro function is used.

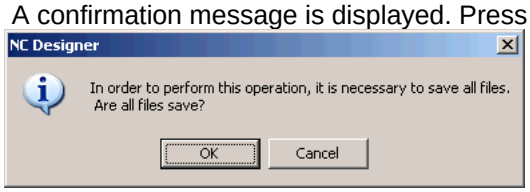
12.1 Export

Export the project to create a project for interpreter execution. Project exportation is also necessary when the macro function is used.

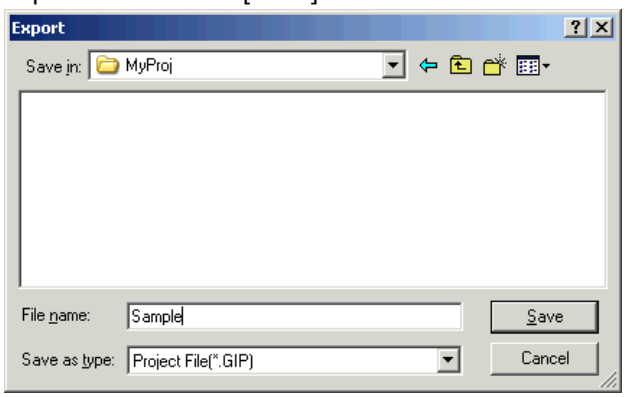
1. From the [File] menu, select [Project convert] - [Export].



A confirmation message is displayed. Press the [OK] button.



2. An "export" dialog box is displayed. Specify the location and file name of the project to be exported. Press the [Save] button.

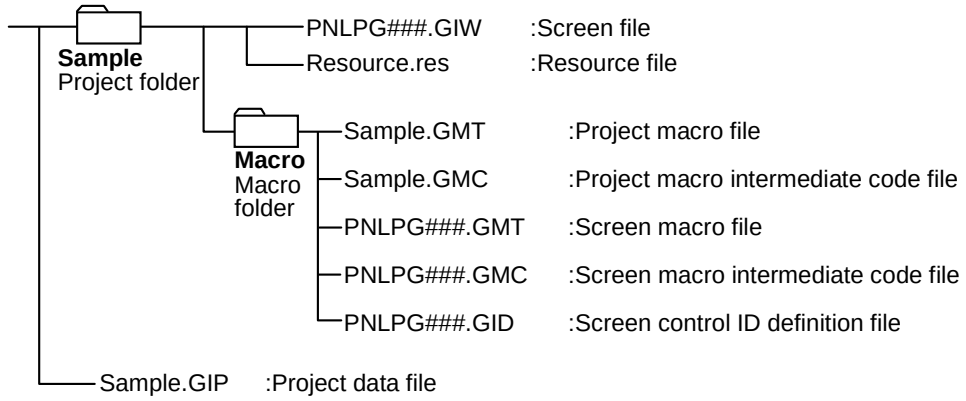


After exportation is successfully finished, a project exportation completion message is displayed. Exportation is finished.

3. To use the macro function, copy the exported project folder to the same folder where "melhmi.exe" is located. "melhmi.exe" is an application window for executing the project, and it is stored in the folder where NC Designer is installed. If the macro function is not used, this operation is unnecessary.

12.2 File Configuration

After the project created with NC Designer is converted to execute in the interpreter mode, a folder of screen files, resource files and macro folder and a project data file are created. Shown below is the file configuration of an interpreter execution project saved under "Sample".



* ###: 3-digit hexadecimal value indicating the page number

NOTE

- When using the project macro, specify the name of project macro intermediate code file, including the path, in config.ini.
  Refer to 15.6.4.1.1 for details.
- Only one project can be registered when the project macro is used.
  If more than one project is specified, unintended screen may be displayed.
- Use the screen macro for drawing screen such as the control operation.

13. Source Code Generation

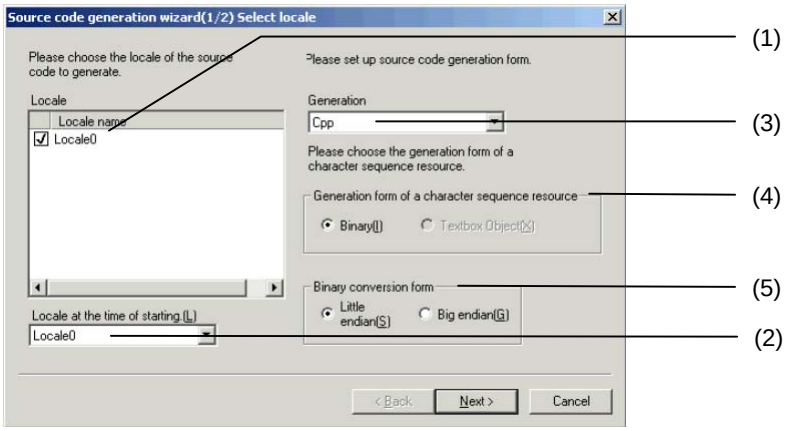
The method for generating source codes from the data created with NC Designer is described in this section.

13.1 Generating Screen Data Source Codes

With NC Designer, source codes are generated from the created screen data.

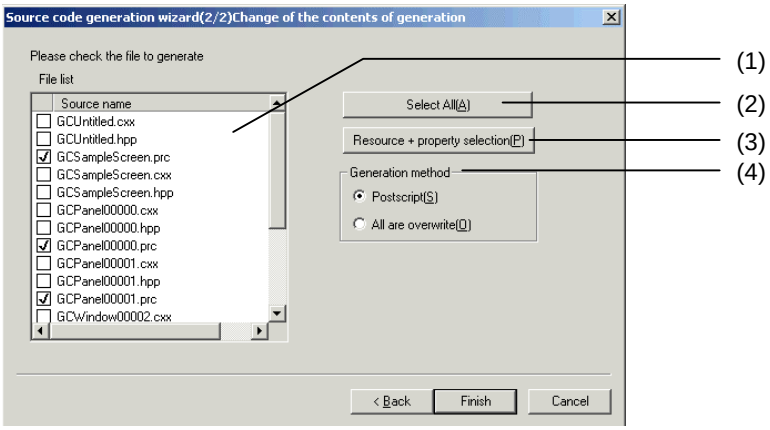
1. Select [Source Code Generation] from the [File] menu.

2. The [Source code generation wizard] is displayed.



No.	Item	Description
(1)	Locale	Select the locale of the source codes to be generated.
(2)	Locale at the time of starting	Select the initial locate at the time of starting the execution module.
(3)	Generation	Select C, C++ or other generation format. The items that can be selected depend on the BSP.
(4)	Generation form of a character sequence resource	Select the character sequence resource generation type between "binary" and "text." If two or more locales are selected, only "binary" can be selected.
(5)	Binary conversion form	Select the binary conversion type of "Little endian". The binary conversion type of M70, M70V, M700, M700VW, M700VS and E70 is a little-endian format.

3. After entering the settings, click on [Next].



No.	Item	Description
(1)	File list	Select the type of the file to be generated. Select the file and click on the box to alternate between check ON/OFF.
(2)	Select All	Select all files. The "file list" selection state does not change.
(3)	Resource + property selection	Click the button to select only the resource and property files. The "file list" selection state does not change.
(4)	Generation method	Select "add" to add only the data changed after previous source code generation into the source file. Select "overwrite all" to overwrite the entire source file.

4. Click on [Finish] to automatically generate source codes.
   After generation is finished, a completion notice dialog box is displayed. Click on the [OK] button.
   The generated source codes are saved in the folder designated during project creation.

Files created during source code generation include the followings.

File name	Description	User code protection
GCXXX.cxx (XXX: project name)	Application launch-related code file. Screen properties are entered to create screens.	Not protected
GCSampleScreen.cxx	Base screen-related code file. Instance of the resource and the created page are generated.	Protected
GCSampleScreen.hpp		
GCSampleScreen.prc		Not protected
GCXXX.cxx (XXX: window/panel name)	Page-related code file. Controls in each page are generated and callback functions of each control are generated.	Protected
GCXXX.hpp (XXX: window/panel name)		
GCXXX.prc (XXX: window/panel name)		Not protected
GCXXXYYY.cxx (XXX: window/panel name. YYY: view frame name)	View frame-related code file. The view frame is controlled.	Protected
GCXXXYYY.hpp (XXX: window/panel name. YYY: view frame name)		
GCXXXYYY.prc (XXX: window/panel name. YYY: view frame name)		Not protected
GCXXXYYYPanelZZZ.cxx (XXX: window/panel name. YYY: view frame name. ZZZ: view frame page number)	View frame page-related code file. Controls in each page of view frame are generated and callback functions of each control are generated.	Protected
GCXXXYYYPanelZZZ.hpp (XXX: window/panel name. YYY: view frame name. ZZZ: view frame page number)		
GCXXXYYYPanelZZZ.prc (XXX: window/panel name. YYY: view frame name. ZZZ: view frame page number)		Not protected
GResource.c	Resource-related code file.	Not protected
GResource.h		
GLoc_XXX.c (XXX: locale name)		

NOTE

- "User code protection" is a function for protecting the source codes created by the user against overwriting during next source code generation. For details, refer to Section 13.3 "User Code Protection."
- The source code specified as "not protected" in the "user code protection" field is not added even if "add" is selected with the source code generation wizard generation method; it is overwritten, instead.
- If GCYYY.cxx already exists at the time of source code generation, the part related to callback functions is not overwritten.
- The screen size and display zoom settings are stored in file CONFIG.INI. CONFIG.INI is located in the destination folder designated in the [BSP Selection] dialog box.

13.2 Source File Format (C++ language version)

The format of each source file of the C++ language version created with the source code generation function is described here.

13.2.1 GCXXX.cxx (XXX: project name)

The case of GCTest.cxx whose project name is "Test" is described below.

```
// GCSample.cpp : Definition of entry point for applications
//
#ifdef WIN32
#include "windows.h"
#endif/*WIN32*/
#include "genifa.h"
#include "gevent.h"
#include "GCSampleScreen.hpp"
#include "GResource.h"

char acMemory[18][32768];

GMemorySpaceInformation gmsiMemory[18]=
{
    {GMEM_SPACE_GDRAW,0,0,0,acMemory[0],32768,NULL},
    {GMEM_SPACE_GEVENT,0,0,0,acMemory[1],32768,NULL},
    {GMEM_SPACE_GRESOURCE,0,0,0,acMemory[2],32768,NULL},
    {GMEM_SPACE_GPI,0,0,0,acMemory[3],32768,NULL},
    {GMEM_SPACE_GWIN,0,0,0,acMemory[4],32768,NULL},
    {GMEM_SPACE_GWIN_PANEL,0,0,0,acMemory[5],32768,NULL},
    {GMEM_SPACE_GWIN_PANEL,1,0,0,acMemory[6],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,0,0,0,acMemory[7],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,1,0,0,acMemory[8],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,2,0,0,acMemory[9],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,3,0,0,acMemory[10],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,4,0,0,acMemory[11],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,5,0,0,acMemory[12],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,6,0,0,acMemory[13],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,7,0,0,acMemory[14],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,8,0,0,acMemory[15],32768,NULL},
    {GMEM_SPACE_GWIN_WINDOW,9,0,0,acMemory[16],32768,NULL},
    {GMEM_SPACE_USER,0,0,0,acMemory[17],32768,NULL},
};
```

```
#ifdef WIN32
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow )
#else/*WIN32*/
extern "C" void WinMain(void)
#endif/*WIN32*/
{
    // TODO: Describe the code in this position.
   GScreenProperty
   gpSceen={{GSCREEN,0,0,0,640,480,7,NULL},{{FILL_BACK_COLOR,0,0},0},ID_FONT00000,
   ID_BORDER00002,RGB16(255,255,255),RGB16(212,208,200),RGB16(10,36,106),RGB16(12
   8,128,128),256,ID_PALETTE00006};
    GMSetMemorySpace( 18, gmsiMemory );

#ifdef WIN32
    GDInitPlatform( hInstance );
#else/*WIN32*/
    GDInitPlatform( NULL );
#endif/*WIN32*/
    GEInitEvent();

    GCWGetScreen()->ImportProperty( (GBaseWindowProperty*)&gpSceen );
    GCWGetScreen()->Create( NULL );

    GCWGetScreen()->SetVisibleStatus( TRUE );

    GEventMessage msg;
    while(1){
                if( GEGetEvent( &msg,-1 ) == FALSE ){
                        continue;
                }
                if( msg.usMessage == GM_QUIT ) break;
                GCWGetScreen()->DispatchMessage( &msg );
    }

    GCWGetScreen()->Delete();

    GEQuitEvent();
    GDQuitPlatform();
#ifdef WIN32
    return 0;
#endif/*WIN32*/
}
```

Screen size specified during project creation

13.2.2 GCSampleScreen.cxx

The case of GCSampleScreen.cxx whose project name is "Test" is described below.

```
#include "GCSampleScreen.hpp"
#include "GWin.h"
#include "GCBasicControl.hpp"
#include "GCButton.hpp"
#include "GCLabel.hpp"
#include "GCList.hpp"
#include "GCPicture.hpp"
#include "GCTextBox.hpp"
#include "GCCheckBox.hpp"
#include "GCRadioButton.hpp"
#include "GCProgressBar.hpp"
#include "GCSLine.hpp"
#include "GCSLines.hpp"
#include "GCSArc.hpp"
#include "GCSRect.hpp"
#include "GCSOval.hpp"
#include "GCSPoly.hpp"
#include "GCSPie.hpp"
#include "GCWindow.hpp"
#include "GEvent.h"
#include "GResource.h"
#include "GCSampleScreen.prc"

GCSampleScreen::GCSampleScreen()
{
    m_usType = GSCREEN;
}

GCSampleScreen::~GCSampleScreen()
{

}

GCClassMP(GCSampleScreen)
GCInheritMP(GCScreen)

GCClassCBMP(GCSampleScreen)
GCEndClassCBMP()
```

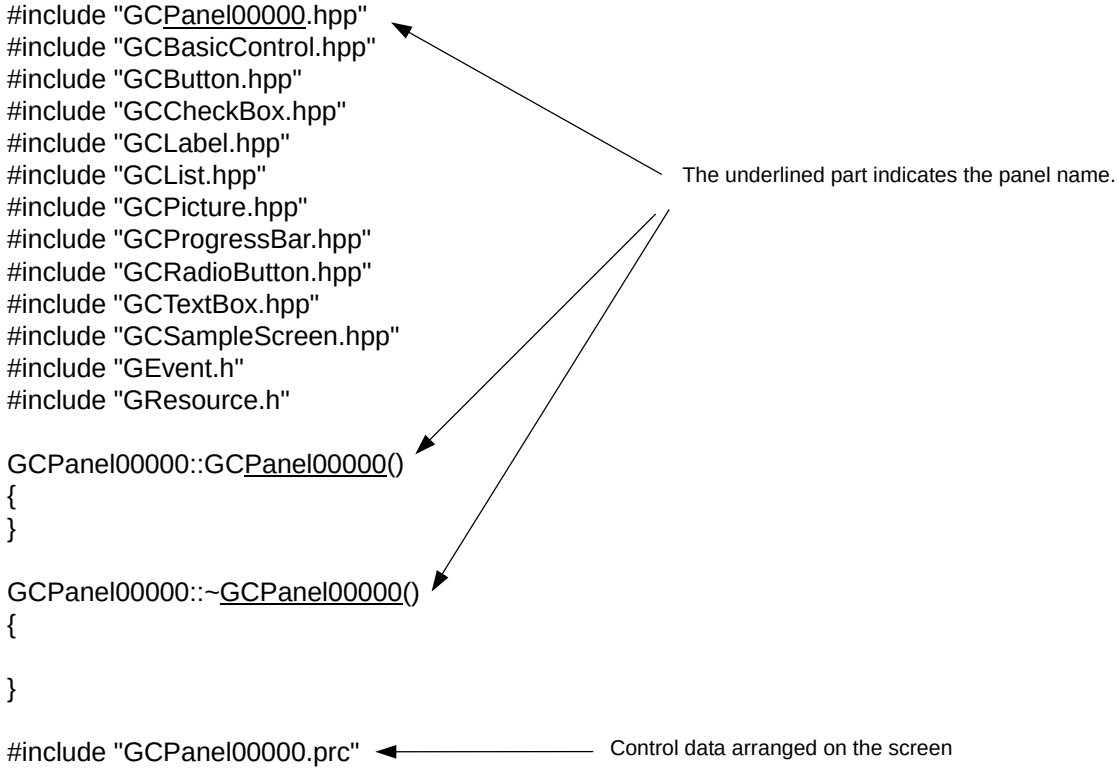
```
#ifdef __cplusplus
extern "C" {
#endif
extern unsigned char cFillPatterns[][8];
extern unsigned char cLinePatterns[][8];
#ifdef __cplusplus
};
#endif


short GCSampleScreen::Create( GCBaseWindow *pParent )
{
    if( GCScreen::Create( pParent ) == FALSE )
                    return FALSE;
     short idx;
     for( idx = 0; idx<NUM_FILL_PATS; idx++ ){
                     GDDefineFillPattern( m_hDraw, idx, (char*)cFillPatterns[idx] );
     }
     for( idx = 0; idx<NUM_DASH_PATS; idx++ ){
                     GDDefinePenDash( m_hDraw, idx+1, (char*)cLinePatterns[idx] );
     }

    //{{INITIAL_PANEL
    ShowPanel( 0 );
    //}}INITIAL_PANEL

     //{{INITIAL_LOCALE
     GRCSetLocale( ID_LOC_LOCALE0 );
     //}}INITIAL_LOCALE

     GESetTimer( this, GSCREEN_BASE_TIMER_ID, GSCREEN_BASE_TIMER_INTERVAL );

     return TRUE;
}
```

Screen generation

Displayed from page 0

13.2.3 GCXXX.cxx (XXX: window/panel name)

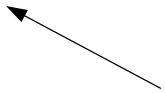
The case of GCPanel00000.cxx whose panel name is "Panel00000" is described below.




```
#include "GCPanel00000.hpp"
#include "GCBasicControl.hpp"
#include "GCButton.hpp"
#include "GCCheckBox.hpp"
#include "GCLabel.hpp"
#include "GCList.hpp"
#include "GCPicture.hpp"
#include "GCProgressBar.hpp"
#include "GCRadioButton.hpp"
#include "GCTextBox.hpp"
#include "GCSampleScreen.hpp"
#include "GEvent.h"
#include "GResource.h"

GCPanel00000::GCPanel00000()
{
}

GCPanel00000::~GCPanel00000()
{

}

#include "GCPanel00000.prc"

short GCPanel00000::CreateChildren( )
{
    GCBaseWindow *pChild;

    unsigned int i;
    for( i = 1; i < sizeof(pProperty)/sizeof(GBaseWindowProperty*); i++ ){
                    pChild = ((GCSampleScreen*)GCWGetScreen())->CreateInstance(
                                                                pProperty[i]->usType,
pProperty[i], this );
                    AddChild( pChild );
                    pChild->Create( this );
    }
    return TRUE;
}
```

Control data arranged on the screen

```
long GCPanel00000::GBUTTON00000OnClick(unsigned short usMessage, long lLParam, long
lUParam)
{
return TRUE;
}
```



Control name with callback function description

```
long GCPanel00000::GBUTTON00000OnCreate(unsigned short usMessage, long lLParam, long
lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnDelete(unsigned short usMessage, long lLParam, long
lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnDraw(unsigned short usMessage, long lLParam, long
lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnKeyPress(unsigned short usMessage, long lLParam,
long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnKeyRelease(unsigned short usMessage, long lLParam,
long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnKillFocus(unsigned short usMessage, long lLParam,
long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnPress(unsigned short usMessage, long lLParam, long
lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnRelease(unsigned short usMessage, long lLParam, long
lUParam)
{
	return TRUE;
}
```

```
long GCPanel00000::GBUTTON00000OnScrollFinish(unsigned short usMessage, long lLParam, long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnSetFocus(unsigned short usMessage, long lLParam, long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnTimer(unsigned short usMessage, long lLParam, long lUParam)
{
	return TRUE;
}

long GCPanel00000::GBUTTON00000OnUser(unsigned short usMessage, long lLParam, long lUParam)
{
	return TRUE;
}
```

13.3 User Code Protection

NC Designer automatically encloses the part to be overwritten by NC Designer in each created file, with tag codes during source code generation.
The source codes added during the next source code generation can be protected by the user's adding source codes other than at the parts enclosed with tag codes.

Lists of tag codes used for each file are shown below.
Screen header file(GCSampleScreen.hpp)

Tag code	Description
//{{OBJECT_TYPE //}}OBJECT_TYPE	Area of definition of object type of panel/window/view frame object created in project is described.
//{{PUBLIC_METHOD //}}PUBLIC_METHOD	Area where the method (function) created in the screen class is described.
//{{PROTECTED_METHOD //}}PROTECTED_METHOD	Area where the method (function) created in the screen class is described.

Screen source file(GCSampleScreen.cxx)

Tag code	Description
//{{INITIAL_PANEL //}}INITIAL_PANEL	Area where the initial panel is specified.
//{{INITIAL_LOCALE //}}INITIAL_LOCALE	Area where the initial locale is specified.

Panel header file(GCXXX.hpp)

Tag code	Description
//{{CONTROL_ID //}}CONTROL_ID	Area where the ID of the control displayed in the panel is described.
//{{PUBLIC_METHOD //}}PUBLIC_METHOD	Area where the definition of the method (function) created in the panel is described.
//{{CALLBACK_METHOD //}}CALLBACK_METHOD	Area where the definition of the callback method (function) created in the panel is described.
//{{PROTECTED_METHOD //}}PROTECTED_METHOD	Area where the definition of the method (function) created in the panel is described.

View frame header file(GCXXXYYY.hpp)

Tag code	Description
//{{PANEL_ID //}} PANEL _ID	Area where the ID of the panel displayed in the view frame is described.
//{{PUBLIC_METHOD //}}PUBLIC_METHOD	Area where the definition of the method (function) created in the view frame is described. The definition of the method provided in the template from the initial state is described.
//{{PROTECTED_METHOD //}}PROTECTED_METHOD	Area where the definition of the method (function) created in the view frame is described.

View frame panel header file(GCXXXYYYPanelZZZ.hpp)

Tag code	Description
//{{CONTROL_ID //}}CONTROL_ID	Area where the ID of the control displayed in the panel is described.
//{{PUBLIC_METHOD //}}PUBLIC_METHOD	Area where the definition of the method (function) created in the panel is described.
//{{CALLBACK_METHOD //}}CALLBACK_METHOD	Area where the definition of the callback method (function) created in the panel is described.
//{{PROTECTED_METHOD //}}PROTECTED_METHOD	Area where the definition of the method (function) created in the panel is described.

NOTE

- Do not modify the part enclosed with tag codes.
- Do not delete any tag code from the file.
- This function is valid if the source code generation method is set at [Postscript].

14. Features and Configuration of GUI Library

This section describes an outline of the GUI library.

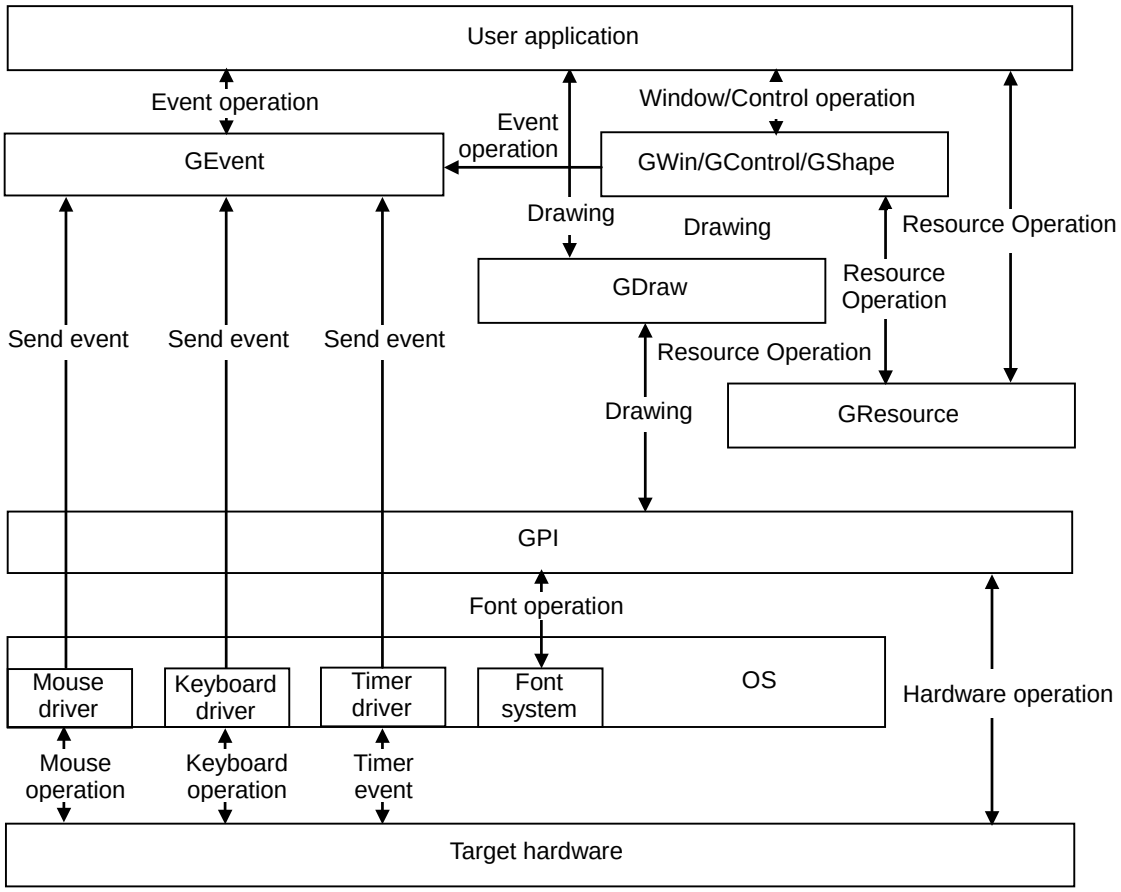
14.1 Features of GUI Library

The graphical user interface (GUI) library is a C++ language library and it strongly supports GUI development. The GUI library provides you with not only drawing functions but also controlling of the mouse, key and other events as well as other functions indispensable for establishment of the GUI such as the window system, so that the GUI can be created without difficulty.

It is also provided with a GPI (graphic platform interface), which is a mechanism for processes depending on hardware, to make porting to each platform easily. When the GUI library is ported to another platform, GPI processes are created according to the target platform.

14.2 Configuration of GUI Library

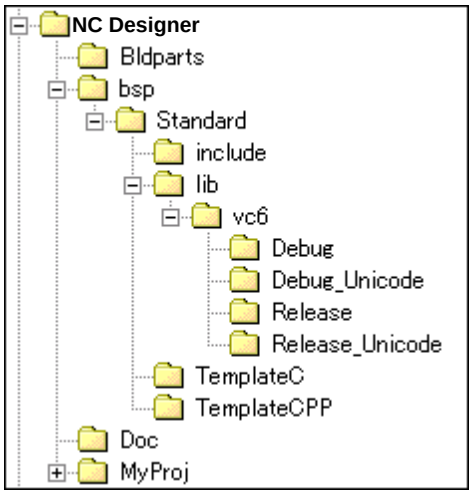
The basic configuration of the GUI library is shown below.



Function name	Description
GDraw	Draws basic figures and characters and specifies coordinates, colors and other drawing environment
GPI	Part dependant on hardware and operating system (drawing to VRAM, font, platform initialization, etc.)
GResource	Handles character strings, solid frames, images and other resources.
GEvent	Controls the mouse, keyboard, timer and other events and window system events such as GWin, GControl and GShape.
GWin/GControl/GShape	Window system

14.3 Folder Configuration

The folder configuration of NC Designer is shown below. (In the case where the installation destination folder name is "NC Designer".)



Each folder is described below.

Folder name		Description
NC Designer		Main folder of NC Designer (Execution files of NC Designer are stored.)
BLDPARTS		Folder storing controls used in NC Designer
bsp		Folder storing BSP
STANDARD		Folder storing Windows version GUI library
Include		Folder storing GUI library header files
lib		Folder storing library files
Vc6	Release-Unicode	Folder storing Unicode-compatible library (for releasing)
MyProj		Folder storing projects

14.4 File Configuration

The file configuration necessary for the development of applications using the GUI library in the Windows environment is described here.

14.4.1 Header File

The header file stored in the "include" folder is shown below.

All header files are included in the "include" folder in addition to Genifa.h.

Header name	Description
Genifa.h	Header file describing definitions of GUI library functions

14.4.2 Library File

The list of library files stored in the Release-Unicode folders inside the "vc6" folder are shown below.

Header name	Description
Genifa.lib	Library for Windows version GDraw, GEvent and GResource
GCWin.lib	Library for Windows version C++ language GWin
gcs.lib	Library for Windows version GCS
resmng.lib	Library for Windows version resource manager

15. Application Execution Method

This section describes the execution method of applications created with NC Designer.

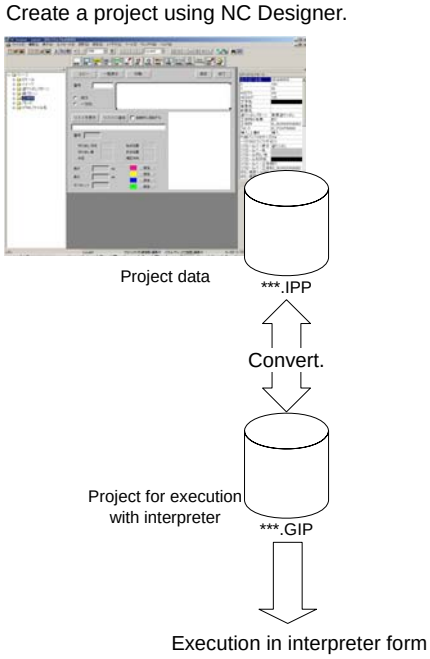
15.1 Application Execution Method

15.1.1 Outline

The execution method of applications created with NC Designer includes two types: interpreter and compilation. Either independent or combinational execution is possible.

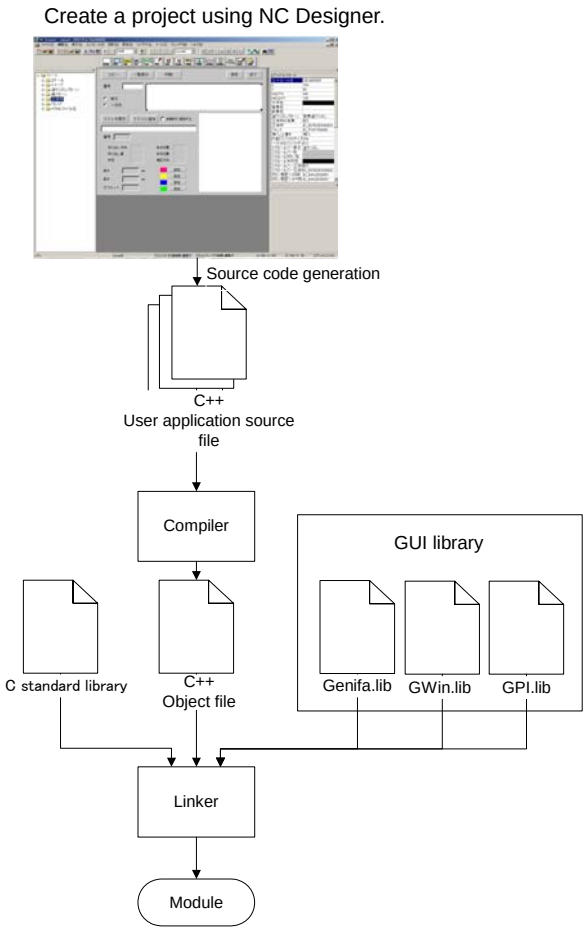
Interpreter Method

With the interpreter method, the project data created with NC Designer is converted into an interpreter project for execution. The feature of the interpreter method is that NC Designer handles all processes from screen establishment to simple control program creation.
Therefore C++ language programming is unnecessary. While the execution speed is slower than that of the compilation method, GUI applications are developed handily.



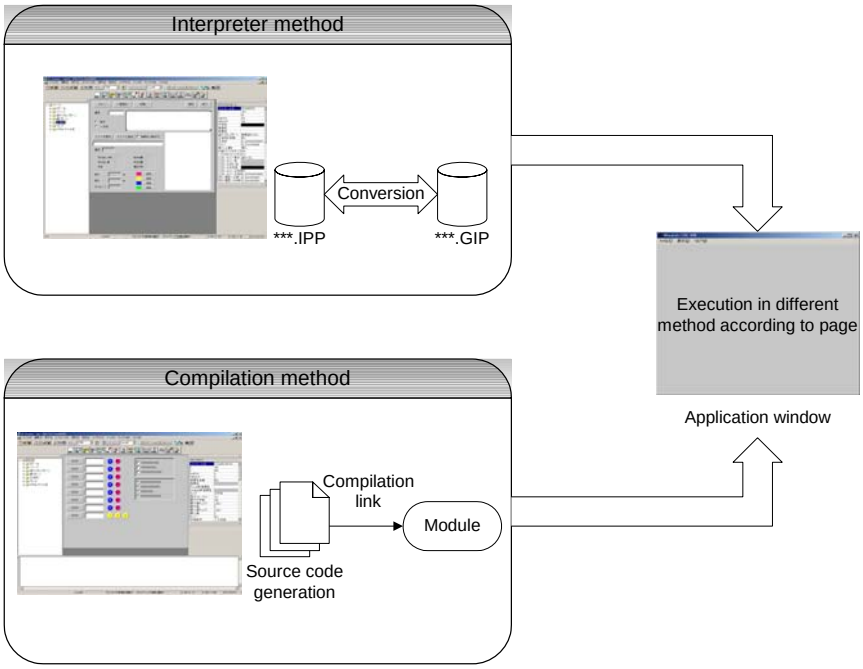
Compilation Method

With the compilation method, a source program is generated from the project data created with NC Designer and it is converted into a module (DLL) for execution. The control program is created in the C++ language and all functions of NC Designer are used to realize various control methods. While C++ language programming is necessary, the execution speed is faster than that of the interpreter method and applications having more complex control functions can be developed.



15.1.2 Independent/Combinational Execution

You can choose the interpreter method, compilation method and combination of both for the execution of the application. Using combination, you can use the advantages of both methods during application development. For example, screens where frequent specification changes are expected are created with the interpreter method, and complex screens are created with the compilation method.



15.2 Interpreter Method

15.2.1 What Is Interpreter Method?

With the interpreter method, the project data created with NC Designer is converted into an interpreter project for execution. The feature of the interpreter method is that NC Designer handles all processes from screen establishment to simple control program creation.
Therefore C++ language programming is unnecessary. While the execution speed is slower than that of the compilation method, GUI applications are developed handily.

15.2.2 Flow of Operation

The procedure for executing the application in the interpreter method is described here.

1. Create a project with NC Designer.

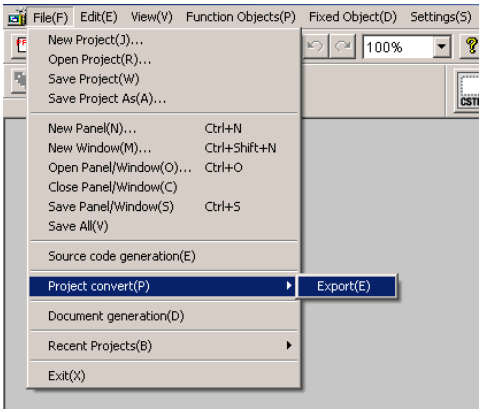
2. To add control programs to the project or controls, describe macros. From the [Settings] menu, select [Panel Macro Edit]/[Project Macro Edit]. Edit the macro in the displayed "Macro Edit" dialog box. For details of the macro editing method, refer to Section 16 "Macro Function."

NOTE

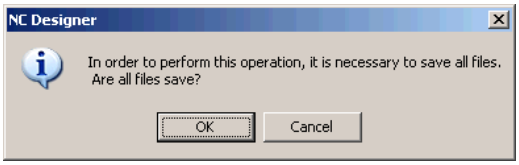
- With the interpreter method, callback events do not function. To add screen switching process to controls, use the macro function.

3. Save the project.

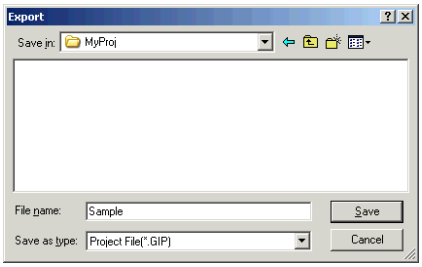
4. Convert the project for interpreter execution. From the [File] menu, select [Project convert] - [Export].



The following confirmation message is displayed. Press the [OK] button.

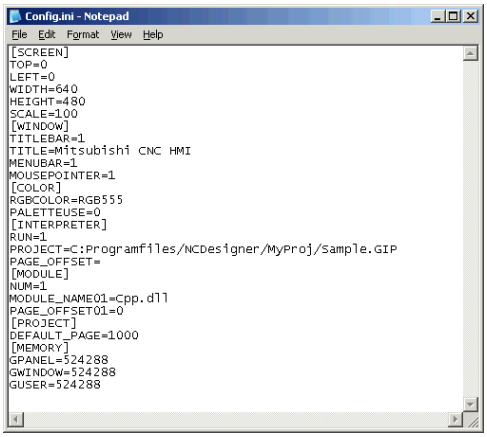


5. The "Export" dialog box is displayed. Enter the destination of the exporting project and the file name. Press the [Save] button.



After exportation is succeeded, a "Project export completed." message is displayed. Exportation is finished.

6. Enter various settings related to execution of the application such as the application execution state and project name to be launched. Use the "Config.ini" file for settings. The Config.ini file is in the folder where NC Designer is installed. Use a text editor to open it.



7. Edit the [INTERPRETER] section in the "Config.ini" file as specified below.

Item	Setting	Description
RUN=	1	Specify the number of projects executed with the interpreter method.
PROJECT=	C:\Program Files\NC Designer \MyProj\Sample.GIP	Specify the name of the interpreter project including the path.
PAGE_OFFSET=	1000	Specify the screen No. offset value.

8. Edit the [PROJECT] section of the "Config.ini" file as specified below.

Item	Setting	Description
DEFAULT_PAGE=	1000	Specify the screen No. displayed first when the project is launched.

After editing, save and close the file.

9. Double click on "melhmi.exe" to launch it and execute the project in the interpreter mode. "melhmi.exe" is an application window for executing a project and is stored in the folder where NC Designer is installed.

15.3 Compilation Method

15.3.1 What Is Compilation Method?

With the compilation method, a source program is generated from the project data created with NC Designer and it is converted into a module (DLL) for execution. The control program is created in the C++ language and all functions of NC Designer are used to realize various control methods. While C++ language programming is necessary, the execution speed is faster than that of the interpreter method and applications having more complex control functions can be developed.

15.3.2 Flow of Operation

The procedure for executing the application in the compilation method is described below.

1. Create a project with NC Designer.

2. Save the project.

3. Generate source codes.

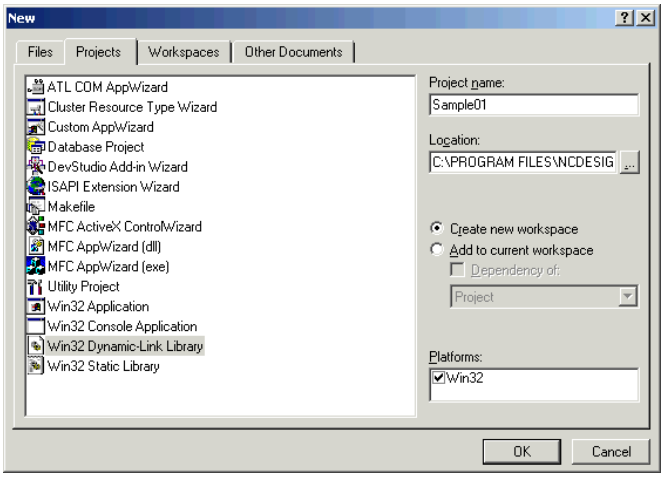
   From the [File] menu, select [Source Code Generation] and follow the displayed "Source code generation wizard" to generate source codes.

   For the source code generation method, refer to Section 13 "Source Code Generation."

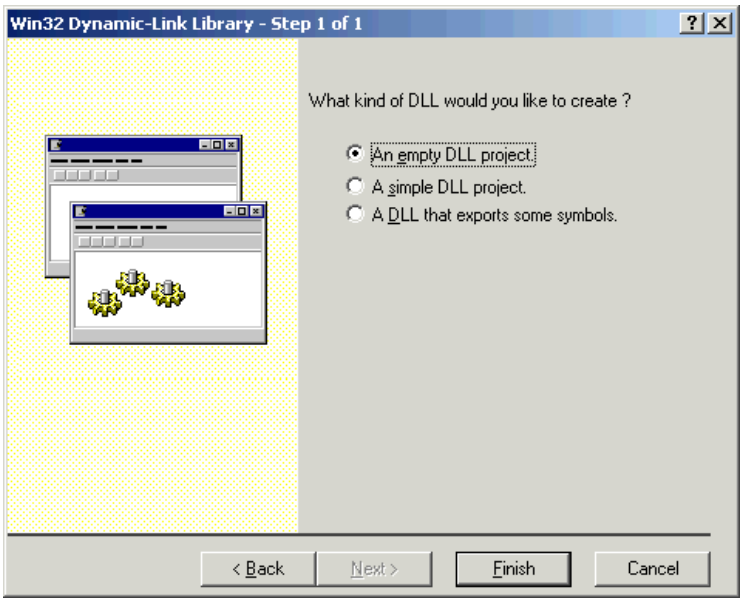
15.3.2.1 Operation Procedure with Visual C++6.0

1. To edit source codes and compile and link, launch Visual C++ Version 6.0 (hereinafter referred to as VC6).

2. Create a VC6 project. From the [File] menu, select [New]. At the "Projects" tab, select "Win32 Dynamic-Link Library" and enter the project name and location. Press the [OK] button.



3. The dialog box shown below is displayed. Select "An empty DLL project" and press the [Finish] button.



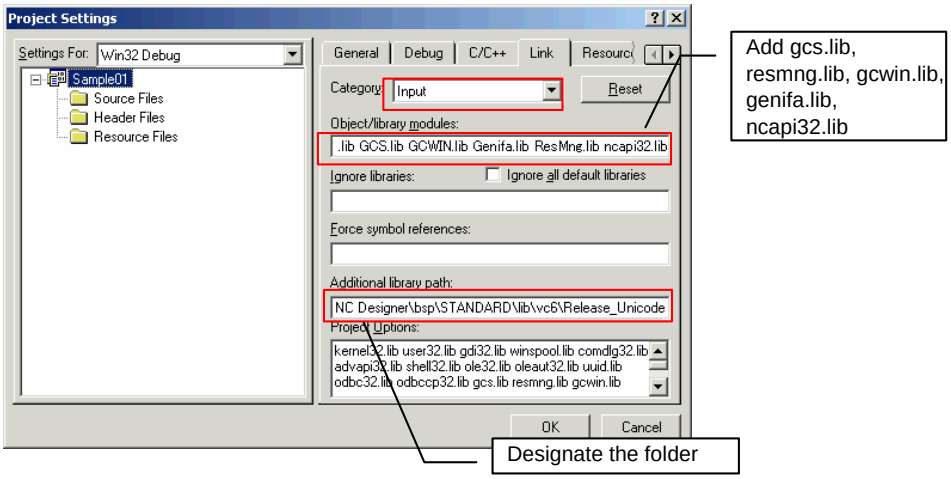
4. Add the source file created with NC Designer to the project. From the [Projects] menu, select [Add To Project] - [Files] and add all files (*.cxx, *.hxx, *.prc, *.c, *.hand *.def) generated with NC Designer.

5. From the [Projects] menu, select [Settings] to display the [Project Settings] dialog box. Display the "Link" tab and add the following file names in the "Object/library modules" setting field in the "Input" category.

gcs.lib, resmng.lib, gcwin.lib, genifa.lib, ncapi32.lib

Similarly, specify "Additional library path" in the "Link" tab. Specify the folder where the VC6 GUI library is stored.
When NC Designer has been installed by default, it will be stored in the following folder. The default installation destination differs depending on the system environment.

- For Windows 2000/XP
  C:\Program Files\NC Designer\bsp\STANDARD\lib\vc6\Release_Unicode
- For Windows Vista/7
  C:\MITSUBISHI CNC\MELSOFT\NC Designer\bsp\STANDARD\lib\vc6\Release_Unicode



6. Next, display the "C++" tab and change the "Preprocessor definitions" setting field in the "Preprocessor" category as shown below.

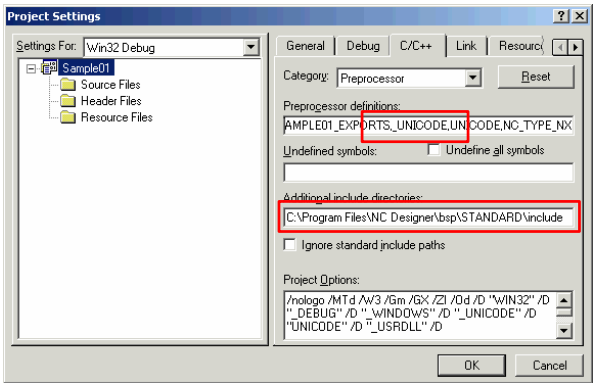
Deleted item	Item to be added
_MBCS	_UNICODE,UNICODE ,NC_TYPE_NX

(Add ",NC_TYPE_NX" when using NC control.)

Specify "Additional include directories."

When NC Designer has been installed by default, it will be stored in the following folder. The default installation destination differs depending on the system environment.

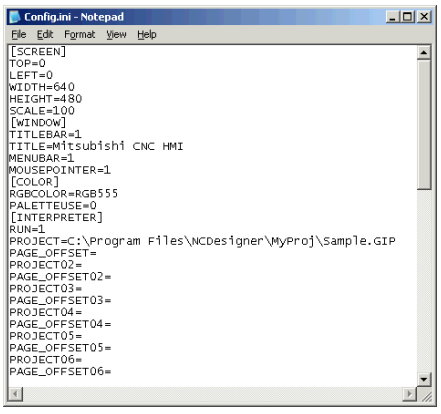
- For Windows 2000/XP
  C:\Program Files\NC Designer\bsp\STANDARD\include
- For Windows Vista/7
  C:\MITSUBISHI CNC\MELSOFT\NC Designer\bsp\STANDARD\include



7. Compile and link.

8. Copy the created library (*.dll) to the folder where "melhmi.exe" is located. "melhmi.exe" is an application window for executing a project and is stored in the folder where NC Designer is installed.

9. Enter various settings related to execution of the application such as the application execution state and project name to be launched. Use the "Config.ini" file for settings. The Config.ini file is in the folder where NC Designer is installed. Use a text editor to open it.



10. Edit the [INTERPRETER] section of the "Config.ini" file as shown below.

Item	Setting	Description
RUN=	0	Specify the number of projects executed in the interpreter method. To execute in the compilation method, specify "0."

11. Edit the [MODULE] section as shown below.

Item	Setting	Description
NUM=	1	Specify the number of projects executed in the compilation method.
MODULE_NAME01=	Sample.dll	Specify the generated module name.
PAGE_OFFSET01=	1000	Specify the screen No. offset value.

12. Edit the [PROJECT] section as shown below.

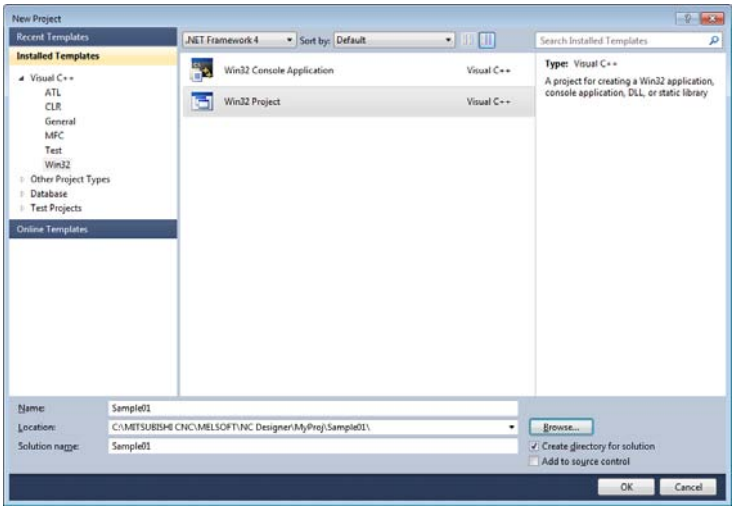
Item	Setting	Description
DEFAULT_PAGE=	1000	Specify the screen No. displayed first when the project is launched.

After setting, save and close the file.

13. Double click on "melhmi.exe" to launch it. The project is executed in the compilation method.

15.3.2.2 Operation Procedure with Visual Studio2010

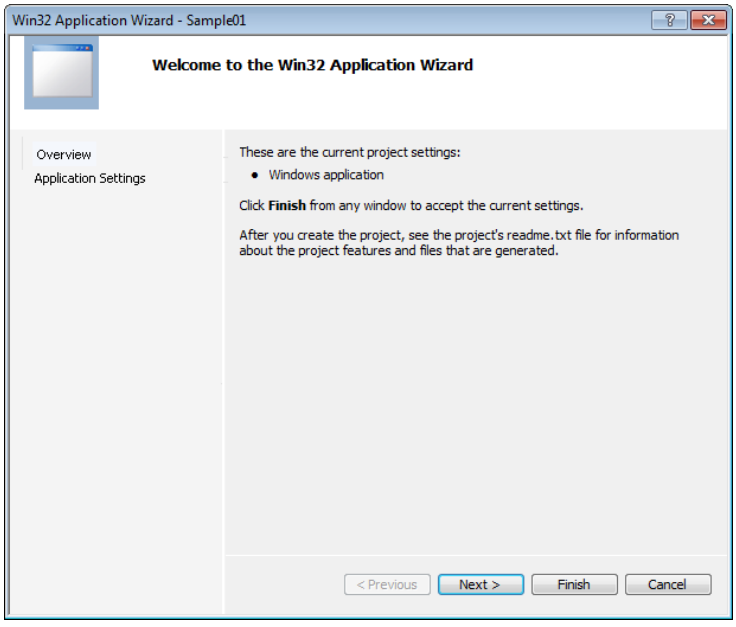
1. Start up Visual Studio2010 (hereinafter called VS2010) to edit the source code, and compile and link.
2. Create a VS2010 project. From the [File] menu, select [New] - [Project...].
   From "Installed Templates" in "New Project" dialog, select "Win32" from "Visual C++", and enter the solution name and Location. Press the [OK] button.



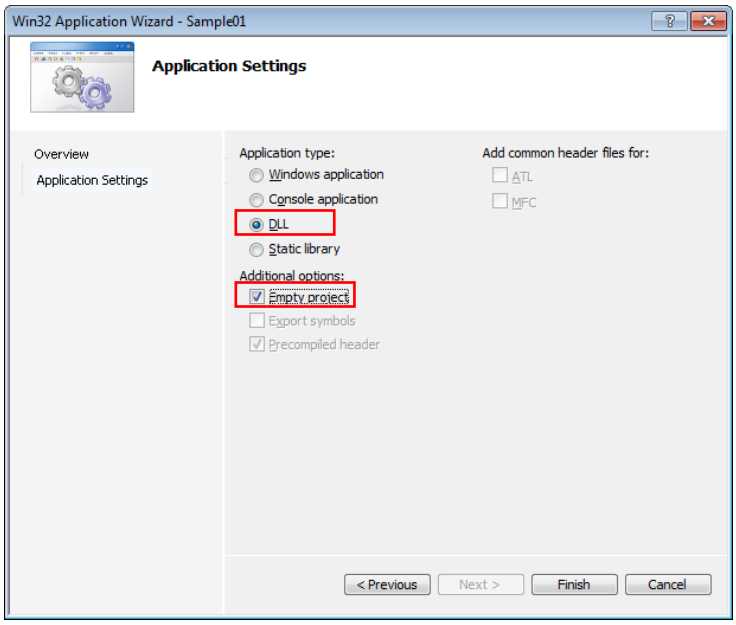
NOTE

- For Visual Studio 2005/2008, a project can be created with the same operation procedure as mentioned above.

3. The dialog box shown below is displayed. Press the [Next] button.

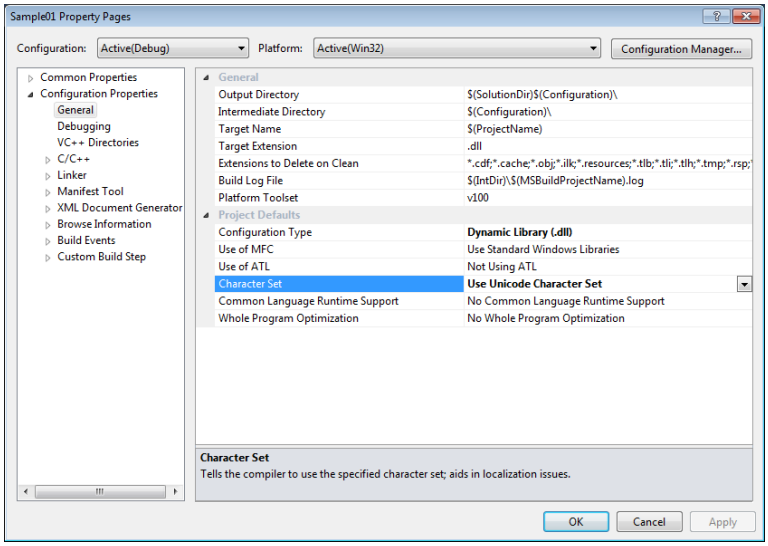


4. The dialog box shown below is displayed. Select ‘‘DLL’’ from ‘‘Application type’’ and ‘‘Empty project’’ from ‘‘Additional options’’. Press the [Finish] button.



5. Add the source file created with NC Designer to the project.
From the [Projects] menu, select [Add Existing Item...] and all files (*.cxx/ *.hxx/ *.prc/ *.c/ *.h/ *.def) generated with NC Designer.

6. From the [Projects] menu, select [Properties] to display "Property Pages" dialog box.
Select "Configuration Properties" - "General ". Select "Use Unicode Character Set" from "Character Set" in "Project Defaults".



Specify each item as shown below.

Category	Item	Details
Configuration Properties - General	Project Defaults - Character Set	Use Unicode Character set
Configuration Properties - C/C++ - General	Additional Include Directories	C:\ MITSUBISHI CNC\MELSOFT\NC Designer \bsp\ STANDARD\include (Note 1)
Configuration Properties -C/C++ - Preprocessor	Preprocessor Definitions	Delete _MBCS. Add _UNICODE, UNICODE and NC_TYPE_NX.
Configuration Properties -C/C++ -Code Generation	Runtime Library	Multi-threaded(/MT)
Configuration Properties -C/C++ - Language	Treat WChar_t As Built in Type	No(/Zc:wchar_t-)
Configuration Properties - Linker - General	Additional Library Directories	C:\ MITSUBISHI CNC\MELSOFT\NC Designer\bsp\ STANDARD\lib\vc6\Release_Unicode (Note 1)
Configuration Properties - Linker - Input	Additional Dependencies	gcs.lib, resmng.lib, gcwin.lib, genifa.lib, ncapi32.lib
	Ignore Specific Default Libraries	Libc.lib
	Module Definition File	Specify the directory where GPROJECT.def is saved.

(Note 1) When NC Designer has been installed by default, it will be stored in the following folder. The default installation destination differs depending on the system environment.

- For Windows 2000/XP
  C:\Program Files\NC Designer\bsp\STANDARD
- For Windows Vista/7
  C: \MITSUBISHI CNC\MELSOFT\NC Designer\bsp\STANDARD

(Note 2) For Visual Studio 2005/2008, add the following setting.

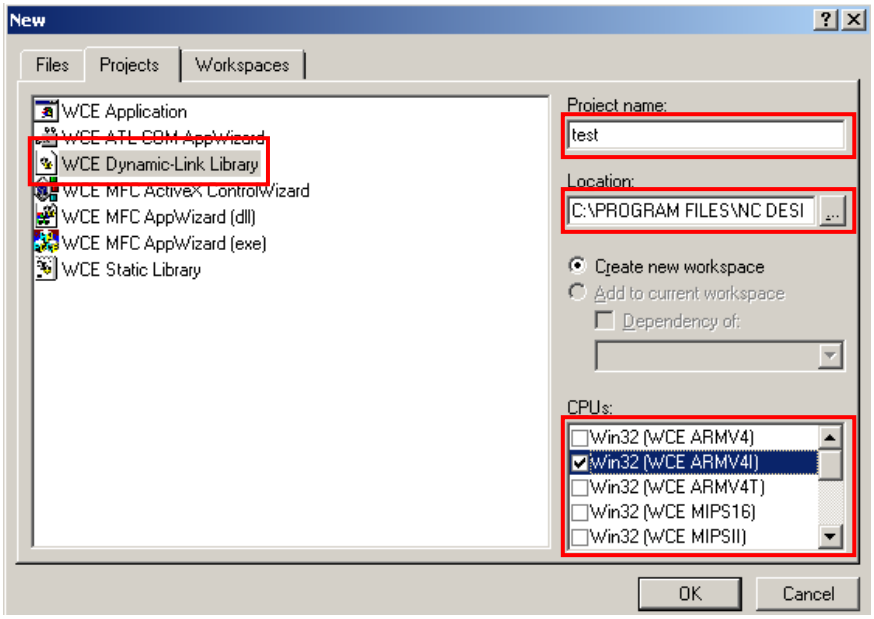
Category	Item	Details
Configuration Properties - C/C++ - General	Detect 64-bit Portability Issues	No

7. Refer to 7 to 13 of "15.3.2.1 Operation Procedure of Visual C++6.0".

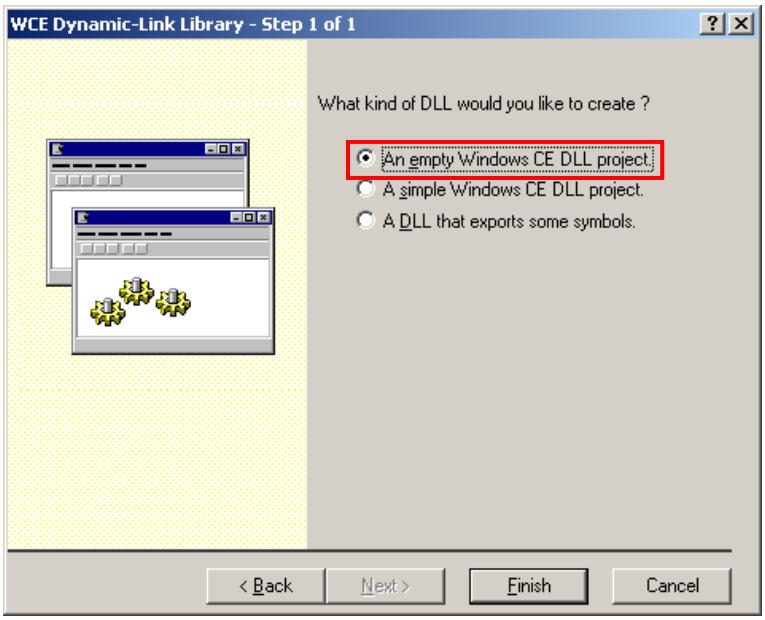
15.3.3 Method of Compilation Environment Establishment in Windows CE

Establishment of compilation environment in Windows CE display

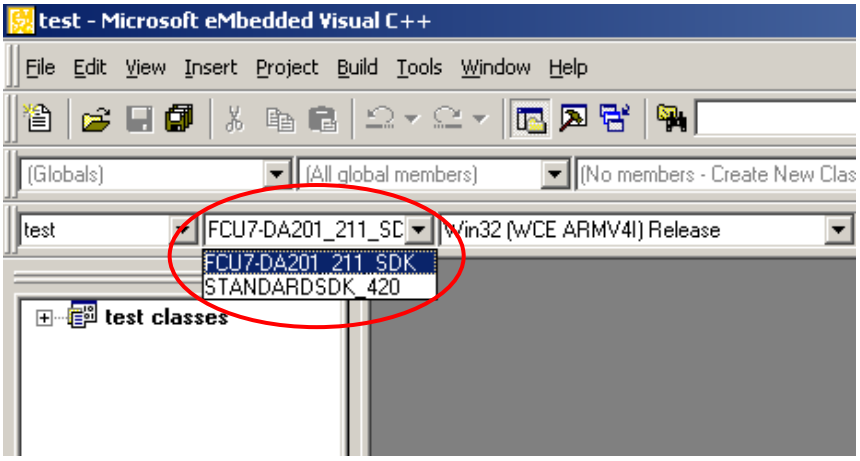
1. Create a project with eMbedded Visual C++ 4.0. From the [File] menu, select [New]. At the "Projects" tab, select "WCE Dynamic-Link Library" and input the project name and stored location. At "CPU", select only "Win32 (WCE ARMV4I)", and remove other checks. After input, press the [OK] button.



2. The dialog box shown below is displayed. Select "An empty DLL project" and press the [Finish] button.



3. From the [Select Active WCE Configuration] of menu bar, select "FCU7-DA201_211_SDK" as shown in the figure below.



4. Add the source file created with NC Designer to the project. When the [Add To Project] - [Files] is selected from the [Project] menu, the following are set because [Insert Files into Project] dialog is displayed. All of the displayed source code file are selected, and press the [OK] button.

5. From the [Project] menu, select [Settings] to display the [Project Settings] dialog box. At the "Link" tab, select "Input" category and set the following.

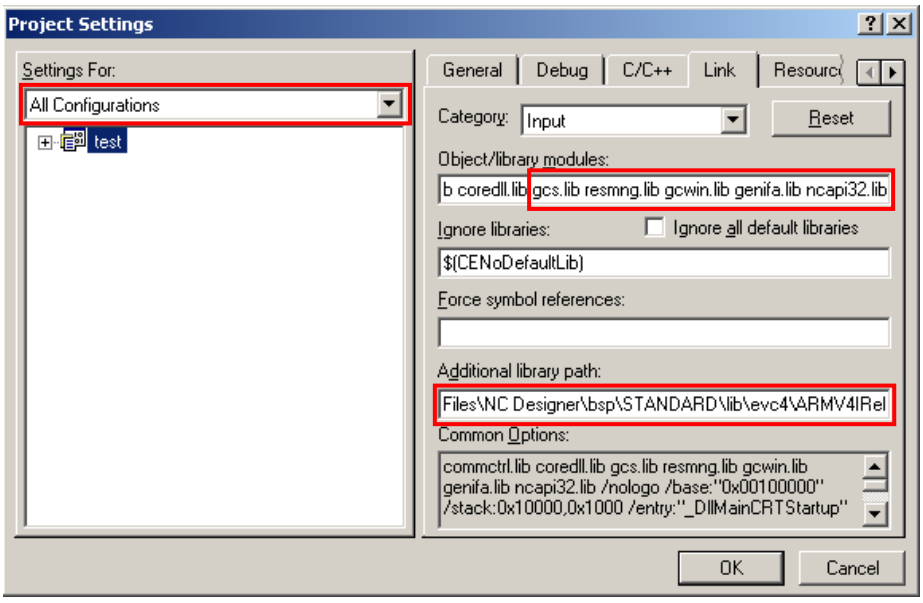
"Settings For" : All Configurations

"Object/library modules"

: gcs.lib resmng.lib gcwin.lib genifa.lib ncapi32.lib

"Additional library path" : If NC Designer has been installed in default folders, the location is following below.

C:\Program Files\NC Designer\bsp\STANDARD\lib\evc4\ARMV4IRel



6. Next, at the "C/C++" tab, select "Preprocessor" category and set the following.

"Settings For" : All Configurations

"Preprocessor definitions" : Add NC_TYPE_NX

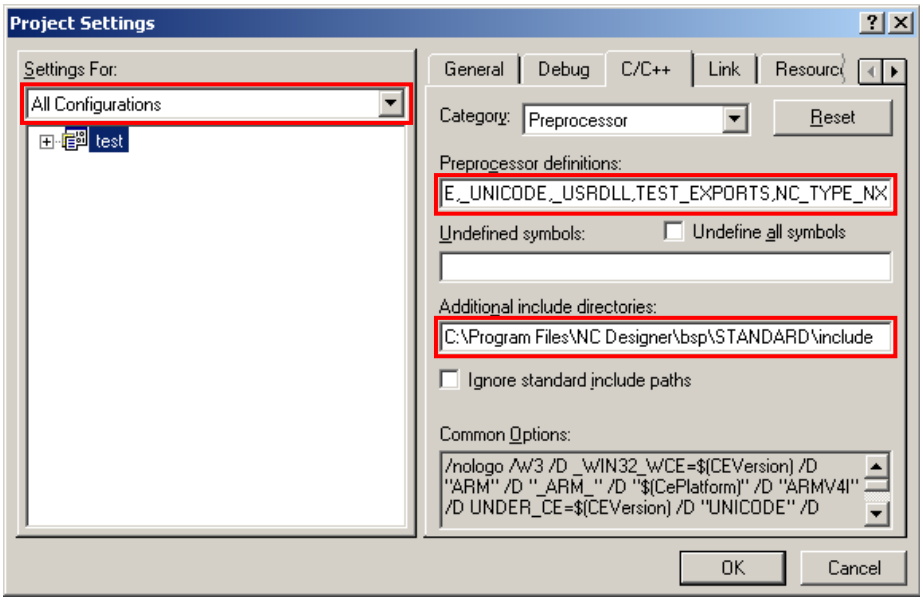
(When the NC control is used, NC_TYPE_NX is added, too.

_MBCS need not be changed to _UNICODE and UNICODE like creating the screen of Windows XP. _UNICODE and UNICODE are added by default.)

"Additional include directories" : If NC Designer has been installed in default folders, the location is following below.

C:\Program Files\NC Designer\bsp\STANDARD\include

After input, press the [OK] button.



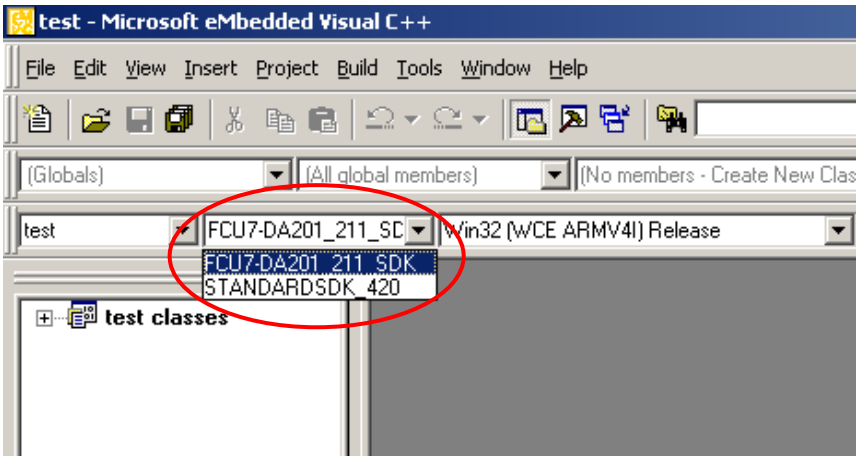
The Release and the Debug can build the source code file by setting the above-mentioned.

(Reference) Preparation of compiler for WindowsCE display unit

Microsoft eMbedded Visual C++4.0 and Service pack (SP1, SP2) can be downloaded from the Microsoft homepage.
Download, and install the following software from the item of the native code development in the homepage.

- eMbedded Visual C++ 4.0
- eMbedded Visual C++ 4.0 Service Pack 1
- eMbedded Visual C++ 4.0 Service Pack 2

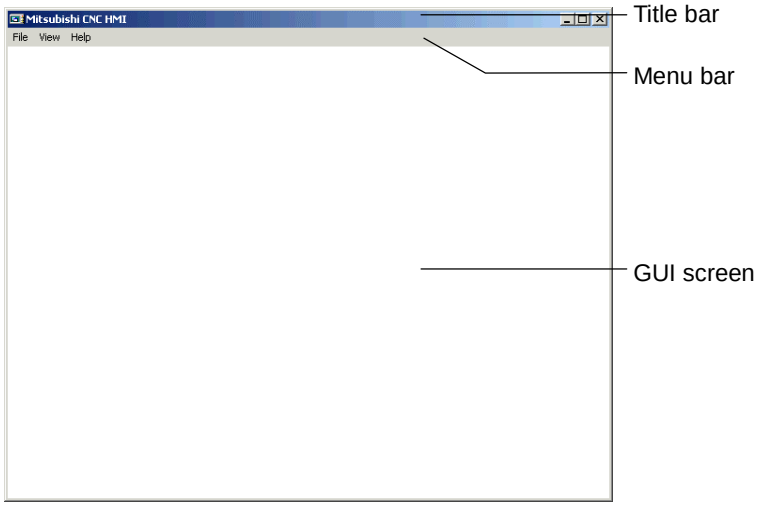
After installed, Windows CE display-specific SDK can be installed by double-clicking "FCU7-DA201_211_SDK.msi" in NC Designer installation CD-ROM.
When the above-mentioned "FCU7-DA201_211_SDK.msi" is installed, "FCU7-DA201_211_SDK" can be selected by "Select Active WCE Configuration" of eMbeddedVisual C++4.0.



15.4 Application Window

15.4.1 What Is Application Window?

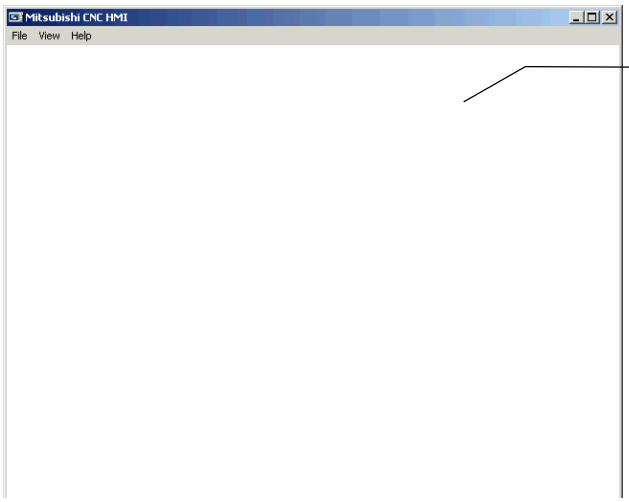
The window displaying the created project is called application window. When executing an application, launch this application window to display panels and windows.



Item	Description
Title bar	The title of the application window is displayed. Specify presence of the title bar and the character string displayed as a title in the Config.ini file. The title bar is displayed for Windows 95, 98, NT, 2000 or XP. It is not displayed in the Windows CE.NET environment.
Menu bar	The menu of the application window is displayed. Presence of the menu bar can be specified in the Config.ini file. The menu bar is displayed for Windows 95, 98, NT, 2000 or XP. It is not displayed in the Windows CE.NET environment.
GUI screen	The panels and windows of the created project are displayed and moved in this area.

15.4.2 Launching the Application Window

1. Open the folder where NC Designer is installed.
   NC Designer is installed in the below folders by default by system environment.
   For Windows 2000/XP
   C:\Program Files\NC Designer
   For Windows Vista/7
   C:\MITSUBISHI CNC\MELSOFT\NC Designer
2. Double click on "melhmi.exe". After it is launched, the window shown below is displayed.



15.4.3 Functions of Application Window

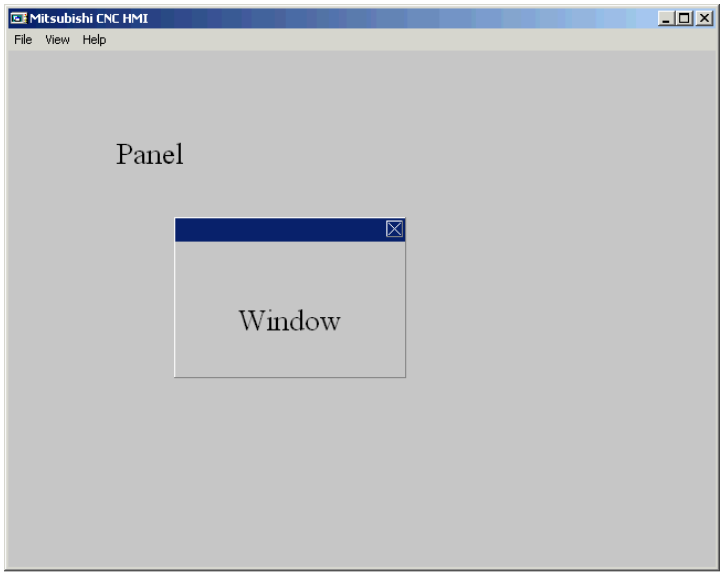
A list of menu items of the application window and application of each item are shown below.

Item		Description
File	Open Project	The project opens. Only the interpreter method project (***.GIP) can be selected.
	Select Screen	Panels/windows open. For details, refer to Section 15.5 "Screen Switching."
	Exit	The application window is closed.
View	Title Bar	The title bar is displayed and hidden alternately.
	Menu Bar	The menu bar is displayed and hidden alternately.
Help	About	The version of NC Designer is displayed.

15.4.4 Screen Configuration

Screen Element

The "page" created with NC Designer consists of a "panel" and "windows." (Refer to Section 2.2 "Specifications of NC Designer.") In the application window, the panel and window can be displayed.



Item	Description
Panel	The panel is displayed in the full screen of the application window. Only one panel is displayed at a time.
Window	The window is displayed in the window state on the application window. Up to 10 windows can be displayed at a time.

Combination of Screen

The panel and window displayed on the application window can be set either in the interpreter or compilation method. Combination of both methods is allowed for the panel and windows, as well.

		Window	
		Interpreter method	**Compilation method**
Panel	**Interpreter method**	○	○
	Compilation method	○	○

15.4.5 Closing the Application Window

To close the application window, perform one of the following operations.

- From the [File] menu, select [Close].
- Click on the  button in the title bar.
- Double click on the  icon at the left end of the title bar.
- Click on the  icon at the left end of the title bar and select [Close] from the displayed control menu box.

- While holding down the [Ctrl] key, press the [F12] key.

15.5 Screen Switching

15.5.1 Outline

The method for switching the screen (panel and window) displayed on the application window is described here.

15.5.2 Screen Switching Method

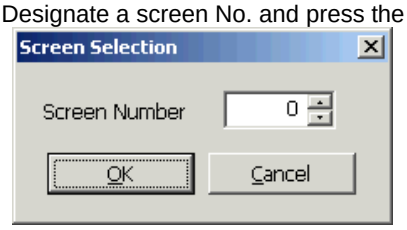
To switch the screen, there are three methods: use the screen switching dialog box of the application window, specify a screen switch event to the control, or use a macro.

Screen Switching Dialog Box

1. Launch the application window.

2. From the [File] menu, select [Open Screen]. The "Screen Selection" dialog box is displayed.

3. Designate a screen No. and press the "OK" button to switch the screen.



Macro

Specify a macro for screen switching to a control to switch the screen.

1. Open a project with NC Designer and create a control for setting screen switching. From the [Settings] menu, select [Panel macro edit] to open the "Macro edit" dialog box.

2. Specify the following macro in the created event. Press the [Event creation] button in the "Macro edit" dialog box and use the displayed "Event creation" dialog box to automatically specify the header and footer.

```
$GButton00000-OnClick
        GCSGEShowPanel(XX);
$End
```

* XX: Specify the new screen No. with the offset value added.

Screen Switching Event (compilation method only)

Specify a screen switching event as a callback process of a control to switch the screen.

1. Open a project with NC Designer and create a control for setting screen switching event. Open properties of the control and specify "Yes" for the callback to which a screen switching event is to be specified.

2. Generate source codes and specify the following event for the callback created in step 1.

```
GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
                                GCSGetScreen(GetGBaseObject()), XX, 0), FALSE);
```

* XX: Specify the new screen No. with the offset value added.

15.5.2.1 Changing From the Custom Screen to 700 Series Standard Screen (F0 Release)

By mounting the following macro processes in OnKeyPress function of the arranged control part on panel, it is possible to change the custom screen to 700 series standard screen by inputting function key.

The page offset No. and the function key No. of each standard screen are as follows.

<The page offset No. of each standard screen>
Monitor screen offset No. : 1000
Setup screen offset No. : 2000
Edit screen offset No. : 3000
Diagnosis screen offset No. : 4000
Maintenance screen offset No. : 5000

<Function key No.>
Function key code for Monitor screen : F1(112) + SHIFT
Function key code for Setup screen : F2(113) + SHIFT
Function key code for Edit screen : F3(114) + SHIFT
Function key code for Diagnosis screen : F4(115) + SHIFT
Function key code for Maintenance screen : F5(116) + SHIFT

Interpreter method

Macro

```
long _lShiftKey;                    'SHIFT key    input status

'The SHIFT key input status is maintained in the 0th bit of LUPARAM.
_lShiftKey = LUPARAM & H1;

if((LLPARAM == 112) && (_lShiftKey == 1))          'Changing the screen to Monitor screen
      GCSGEShowPanel(1000);
elseif((LLPARAM == 113) && (_lShiftKey == 1))      'Changing the screen to Setup screen
GCSGEShowPanel(2000);
elseif((LLPARAM == 114) && (_lShiftKey == 1))      'Changing the screen to Edit screen
      GCSGEShowPanel(3000);
elseif((LLPARAM == 115) && (_lShiftKey == 1))      'Changing the screen to Diagnosis screen
      GCSGEShowPanel(4000);
elseif((LLPARAM == 116) && (_lShiftKey == 1))      'Changing the screen to Maintenance screen
      GCSGEShowPanel(5000);
endif;
```

Compilation method

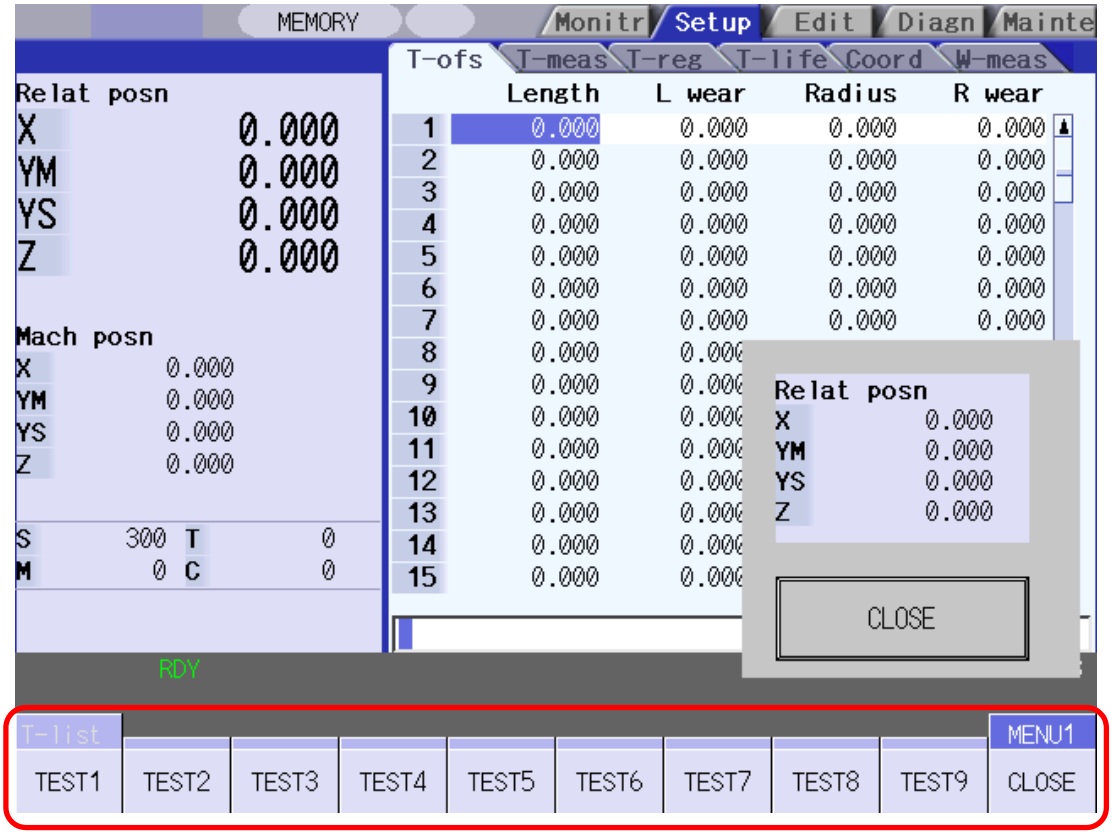
Source code

```
#define KEY_SHIFT		0x01
#define GK_F1			112
#define GK_F2			113
#define GK_F3			114
#define GK_F4			115
#define GK_F5			116


if((lUParam & KEY_SHIFT) == KEY_SHIFT)
{
	if(lLParam == GK_F1)			//Changing the screen to Monitor screen
	{
		GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
			GCSGetScreen(GetGBaseObject()), 1000, 0), FALSE);
	}
	else if(lLParam == GK_F2)		//Changing the screen to Setup screen
	{
		GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
			GCSGetScreen(GetGBaseObject()), 2000, 0), FALSE);
	}
	else if(lLParam == GK_F3)		//Changing the screen to Edit screen
	{
		GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
			GCSGetScreen(GetGBaseObject()), 3000, 0), FALSE);
	}
	else if(lLParam == GK_F4)		//Changing the screen to Diagnosis screen
	{
		GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
			GCSGetScreen(GetGBaseObject()), 4000, 0), FALSE);
	}
	else if(lLParam == GK_F5)		//Changing the screen to Maintenance screen
	{
		GESetEvent(GECreateEventMessage(GM_SHOWPANEL,
			GCSGetScreen(GetGBaseObject()), 5000, 0), FALSE);
	}
}
```

15.5.2.2 Changing the Menu Name While Displaying Custom Screen (Menu Release)

By using the GCSMenuSetMenuButtonLowerName_all function, the menu name when the custom window (menu registered) is displayed can be changed.

<Screen images>



Interpreter method

<GCSMenuSetMenuButtonLowerName_all function format>
1st argument : The page offset No. added custom window
Monitor screen : 1000
Setup screen : 2000
Edit screen : 3000
2nd argument : 0 fixed
3rd argument : 1 fixed
4th argument : Character string displayed in menu

Macro

```
GCSMenuSetMenuButtonLowerName_all(2000,0,1,
" TEST1, TEST2, TEST3, TEST4, TEST5, TEST6, TEST7, TEST8, TEST9, CLOSE");
```

(Note 1) The menu name can be displayed by seven characters a line and two columns, and can be displayed by 14 characters in total.
(Note 2) When eight characters or more are set to the menu name, the menu name is automatically displayed by two lines.

Compilation method

<GCSSetMenuButtonLowerName_all function format (when all menu names are changed)>
1st argument : Menu control part object pointer

It is necessary to acquire the menu control part object by GCSGetChild function beforehand.
When the menu control part object is acquire by GCSGetChild function, the following constant is specified for the 2nd argument of the GCSGetChild function according to the registered screen.

Monitor screen : 58
Setup screen : 39
Edit screen : 16

2nd argument : 1 fixed
3rd argument : Character string displayed in menu pointer (10 pointers)

<GCSSetMenuButtonLowerName_one function format (when one menu name is changed)>
1st argument : Menu control part object pointer
2nd argument : 1 fixed
2nd argument : The menu No. changed menu name
3rd argument : Character string displayed in menu pointer (1 pointer)

Source code

```
char* _psMenuString[10];
long  _lParentType = 0;
GBaseObject* _pParent = GCSGetPanel(GCSGetScreen(GetGBaseObject()));
GBaseObject* _pGCNXMenuSub;

//Set the displayed character string
_psMenuString[0] = (char*)&L"TEST1";
_psMenuString[1] = (char*)&L"TEST2";
_psMenuString[2] = (char*)&L"TEST3";
_psMenuString[3] = (char*)&L"TEST4";
_psMenuString[4] = (char*)&L"TEST5";
_psMenuString[5] = (char*)&L"TEST6";
_psMenuString[6] = (char*)&L"TEST7";
_psMenuString[7] = (char*)&L"TEST8";
_psMenuString[8] = (char*)&L"TEST9";
_psMenuString[9] = (char*)&L" CLOSE ";

//Get the menu control part object
//Following XX values specify the following constant: Monitor = 58, Setup = 39, Edit = 16.
_pGCNXMenuSub = (GBaseObject*)GCSGetChild(_pParent, XX);

//Set the menu name
if( _pGCNXMenuSub != 0 )
{
        GCSSetMenuButtonLowerName_all(_pGCNXMenuSub, 1, _psMenuString);
}
```

· When only one menu name is changed, specifying it as follows.

```
//Set the menu name second from the left
GCSSetMenuButtonLowerName_one(_pGCNXMenuSub, 1, 2, &_psMenuString[1])
```

15.5.2.3 Closing the Custom Screen (Menu Release)

The GCSMenuSendProcessID function is used to close the custom window registered in the menu release.

Interpreter method

<GCSMenuSendProcessID function format>
1st argument : The page offset No. added custom window
Monitor screen : 1000
Setup screen : 2000
Edit screen : 3000
2nd argument : 0 fixed
3rd argument : 0 fixed

Macro

```
GCSMenuSendProcessID(2000,0,0);
```

Compilation method

Issuing the user event by using GESetEvent function and GECreateEventMessage function closes the custom screen displayed by the menu release.
The format of each function is as follows.

<GESetEvent function format>
1st argument : The return value of GECreateEventMessage function
2nd argument : FALSE fixed

<GECreateEventMessage function format>
1st argument : GM_USER fixed
2nd argument : Panel object pointer (The following refer to the getting method.)
3rd argument : USNX_PROCESSID fixed
4th argument : 0 fixed

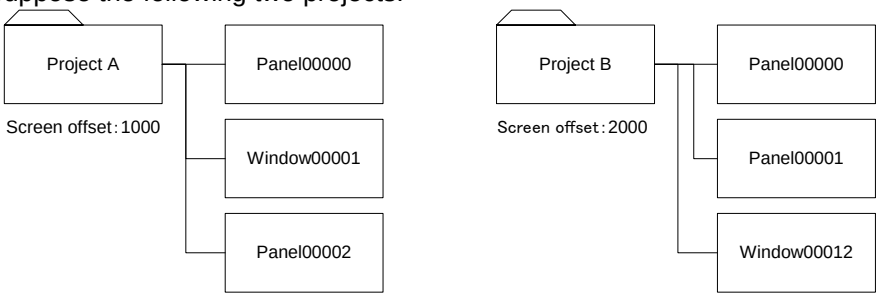
Source code

```
GBaseObject *_gbObj = GCSGetPanel(GCSGetScreen(GetGBaseObject()));
GESetEvent(GECreateEventMessage(GM_USER, _gbObj,
                    USNX_PROCESSID, 0), FALSE);
```

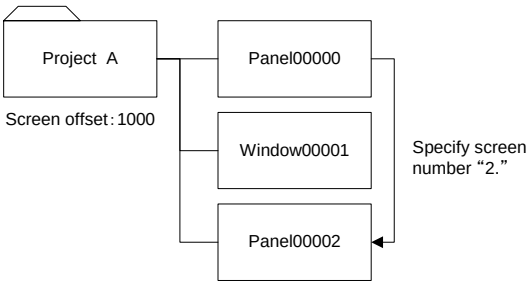
15.5.3 Screen No. Designation Method

The screen No. designated for screen switching varies between screen switching in the same project and that across different projects.

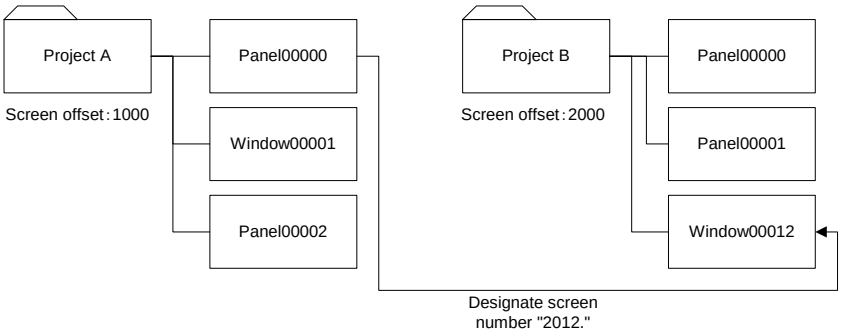
Suppose the following two projects.



To switch the screen in the same project, designate the screen No. specified in NC Designer.
For example, to switch the screen from "Panel00000" to "Panel00002" in project A, specify screen number "2."



To switch the screen across different projects, designate the sum of the screen number and the screen offset value for each project as a screen number.
To switch the screen from project A to "Window 00012" of project B, specify screen No. "2012."



A summary of screen No. examples designated when the screen is switched is shown below.
Project A (screen offset: 1000)

Screen No.	Screen name	Screen switched in same project	Screen switched across different projects
0000	Panel00000	0	1000
0001	Window00001	1	1001
0002	Panel00002	2	1002

Project B (screen offset: 2000)

Screen No.	Screen name	Screen switched in same project	Screen switched across different projects
0000	Panel00000	0	2000
0001	Panel00001	1	2001
0012	Window00012	12	2012

15.5.4 Panel Switching History

NC Designer stores the history of switched panels when screens are switched between panels. The specifications related to panel switching history are shown below.

Item	Specification
Max. records	32
Storage method	Ring buffer method After the maximum number of records (32) is exceeded, the oldest record is deleted and the new record is saved.

NOTE

- Only panel-to-panel switching is recorded. Window display or view frame switching is not recorded.

Panel Switching Using the History

The panel switching history can be used to restore the display panel or advance it. Use a function or macro to operate.

Item	Function/macro to be used
Restore	Function: GCSPrevPage Macro: GCSPrevPage
Advance	Function: GCSNextPage Macro: GCSNextPage

NOTE

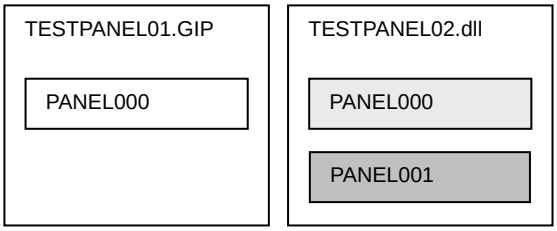
- For the usage of the macro, refer to Section 16 "Macro Function."

15.5.5 Displaying Previously Displayed Custom Screen

If you wish to display a previously displayed custom screen by inputting a function key, you need to define the offset No. (6000 to 7999) of the custom release screen to be held as the previously displayed screen. To define this No., specify the No. using commas as a delimiter in the "PANEL_HOLDXX" key of the [COFFSET] section in the customdef.ini file.
The examples below are not to hold and hold the screen No. using a function (F0) key.

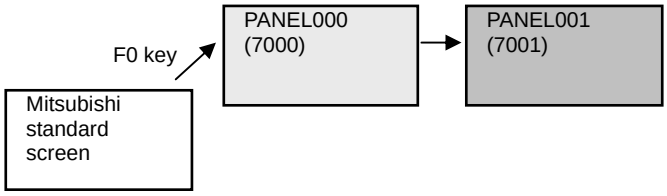
Example)
config.ini

```
[INTERPRETER]
RUN=1
PROJECT01=D:\custom\TESTPANEL01.GIP
PAGE_OFFSET01=6000

[MODULE]
NUM=1
MODULE_NAME01=D:\custom\TESTPANEL02.dll
PAGE_OFFSET01=7000

(The rest is omitted)
```



Example 1) Not hold the screen No.
customdef.ini

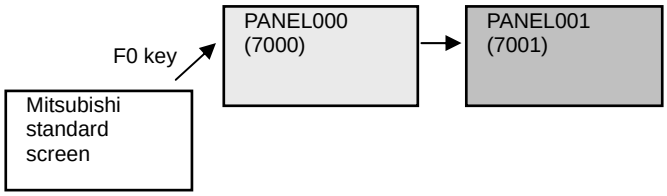
```
[COFFSET]
FUNC_ID01=1
PANEL_OFFSET01=7000

(The rest is omitted)
```



1. When the F0 key is pressed, the screen displays PANEL000.
2. Change the screen from PANEL000 to PANEL001, and then change back to Mitsubishi standard screen.
3. When F0 is pressed again, the screen displays PANEL000.

Example 2) Hold the screen No. within the same project
<Example 2-1>
customdef.ini

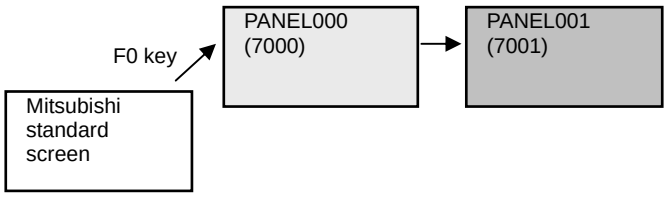
```
[COFFSET]
FUNC_ID01=1
PANEL_OFFSET01=7000
PANEL_HOLD01=7000

(The rest is omitted)
```



1. When the F0 key is pressed, the screen displays PANEL000.
2. Change the screen from PANEL000 to PANEL001, and then change back to Mitsubishi standard screen.
3. When the F0 key is pressed again, the screen displays PANEL000.

<Example 2-2>
customdef.ini

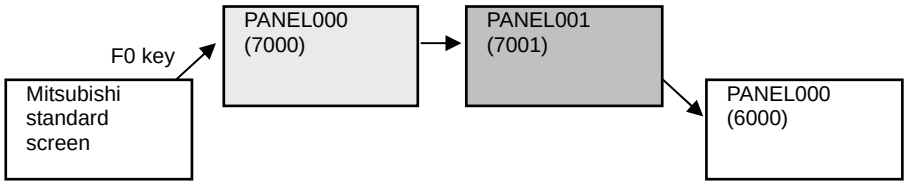
```
[COFFSET]
FUNC_ID01=1
PANEL_OFFSET01=7000
PANEL_HOLD01=7000-7001

(The rest is omitted)
```



1. When the F0 key is pressed, the screen displays PANEL000.
2. Change the screen from PANEL000 to PANEL001, and then change back to Mitsubishi standard screen.
3. When the F0 key is pressed again, the screen displays PANEL001.

Example 3) Hold the screen No. of other project
customdef.ini

```
[COFFSET]
FUNC_ID01=1
PANEL_OFFSET01=7000
PANEL_HOLD01=7000-7001,6000

(The rest is omitted)
```



1. When the F0 key is pressed, the screen displays PANEL000(7000).
2. Change the screen from PANEL000(7000) to PANEL001(7001), from PANEL001(7001) to PANEL000(6000), and then change back to Mitsubishi standard screen.
3. When the F0 key is pressed again, the screen displays PANEL000(6000).

When the following macro process is added to the screen change process, the custom screen to be displayed by inputting a function key can be the one that was previously displayed.

Interpreter Method

<GCSGetLastPanelNumber function format>
1st argument: Function key (0 to 3)

Macro

```
LONG _lPanelNumber = GCSGetLastPanelNumber(0)
GCSGEShowPanel(_lPanelNumber);
```

Compilation Method

<GCSGetLastPanelNumber function format>
1st argument: Module screen object pointer (see below for how to get)
2nd argument: Function key (0 to 3)

Source Code

```
GBaseObject *_pScreenObj = GCSGetModuleScreen();
long _lPanelNumber = GCSGetLastPanelNumber(_pScreenObj, 0);
GESetEvent(GECreateEventMessage(GM_SHOWPANEL, GCSGetScreen(GetGBaseObject()),
_lPanelNumber, 0), FALSE);
```

15.6 Custom Release

15.6.1 Outline

Custom release is an function which allows the user-original window to display as a standard screen or another screen to operation.

Custom release includes, mainly, F0 release and menu release:

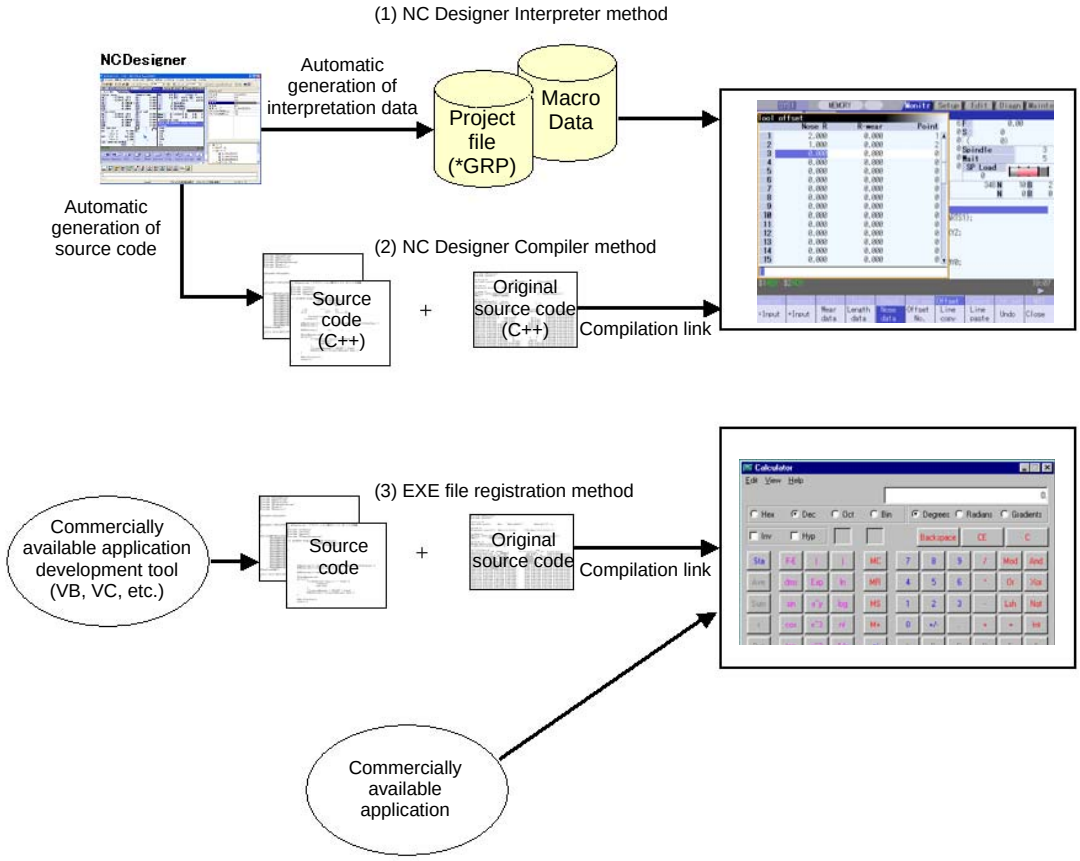
F0 release: Custom release screen (Note 1) can be registered to function keys (F0, SEP, window display, window selection).
When a function key is pressed, the registered custom release screen will be displayed.
This type can be registered with "NC Designer interpreter method", "NC Designer compilation method" and "Executing file registration method" (Note 3).

Menu release: Custom release window (Note 2) can be registered in the main menu of the monitor screen, setup screen and edit screen.
When the registered menu is pressed, the custom release window will be displayed.
This type can be registered with "NC Designer interpreter method", "NC Designer compilation method" and "Executing file registration method" (Note 3).
Depending on the conditions, display/non-display of the custom menu can be changed.
Main menu contents of the monitor, setup and edit screen can be rearranged.

(Note 1) An HMI screen originally created with NC Designer by the user or an execution file prepared by the user.
(Note that an execution file prepared by the user cannot be used in M70/M70V/M700VS/E70.)

(Note 2) An HMI window originally created with NC Designer by the user or an execution file prepared by the user.
(Note that an execution file prepared by the user cannot be used in M70/M70V/M700VS/E70.)

(Note 3) "Executing file registration method" cannot be used in M70/M70V/M700VS/E70.



(1) NC Designer interpreter method (GIP method)
The interpreter data automatically generated with NC Designer can be displayed as an operation screen.
When a simple screen is displayed by using the control that NC Designer provides with, this method is suitable.

(2) NC Designer compilation method (DLL method)
The DLL is created by editing the source code automatically generated by NC Designer and compilation/linking.
The created DLL can be displayed as an operation screen.
When complex processing is executed by using the control that NC Designer provides with, this method is suitable.

(3) Executing file registration method (EXE method)
The execution file (EXE file) originally developed can be displayed as an operation screen.
When an original operation screen is created without using the control that NC Designer provides with, this method is suitable.
(Note that the executing file registration method cannot be used in M70/M70V/M700VS/E70.)

Each feature is shown below.

	NC Designer interpreter method	NC Designer compilation method	Executing file registration method
Creation	◎	○	△
Process speed	△	◎	◎
Flexibility	△	○	◎
Functions	△	○	◎

15.6.2 S/W Configuration

15.6.2.1 Necessary Applications

The following applications are needed for custom release according to the method.

Release method	NC Designer	Visual Studio (Note1)	Application development tool on the market
Interpreter method	○	-	-
Compilation method	○	○	
Executing file registration method	-	△ (Only when the application is developed by using VC++.)	△ (Only when the application is developed by using an application development tool on the market.)
Changed the arrangement of the main menu	-	-	-

○…Necessary -…Not necessary △…Necessary according to the usage

(Note 1) The following application is needed by the display unit used.

<When using M700/M700VW(FCU7-DA3xx/4xx/6xx) >
- Microsoft Visual C++ 6.0 (Service Pack 5 or later)
- Visual Studio 2005/2008/2010

<When using M700(FCU7-DA2xx)>
- Microsoft eMbedded Visual C++ 4.0 (Service Pack 4 applied)
  -> Download from Microsoft Corporation homepage
- SDK for M700(FCU7-DA2xx) (File name : FCU7-DA201_211_SDK.msi)
  -> Provided by installation NC Designer installation CD-ROM.

<When using M70/M70V/M700VS/E70 unit>
- Tornado for MIPS 2.1

15.6.2.2 Necessary Files

The necessary files are as follows for custom release.

File name	Usage	Storage folder (Note1)
Config.ini	This is used when the DLL file and the GIP file are registered as an operation screen.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/
melAppCtrl.ini	This is used when the execution file is registered as an arbitrary key.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/
customdef.ini	This is used when the following cases. - When the custom release window is added by menu release. - When the custom release window is added by F0 release (Excluding executing file registration method). **(Note)** Describe it by UNICODE text.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/
*.jpg	This is a picture file of the icon displayed in the main menu at the menu release.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\img\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\img\ - When using M70/M70V/M700VS/E70 unit /custom/
*.DLL	This is custom screen data file created by compilation method.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/
*.GIP and same name folder	This is custom screen data file created by interpreter method.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/
*.EXE	This is an execution file registered by the executing file registration method.	- When using M700/M700VW (FCU7-DA3xx/4xx/6xx) D:\Custom\ - When using M700 (FCU7-DA2xx) \Memory Card\Custom\ - When using M70/M70V/M700VS/E70 unit /custom/

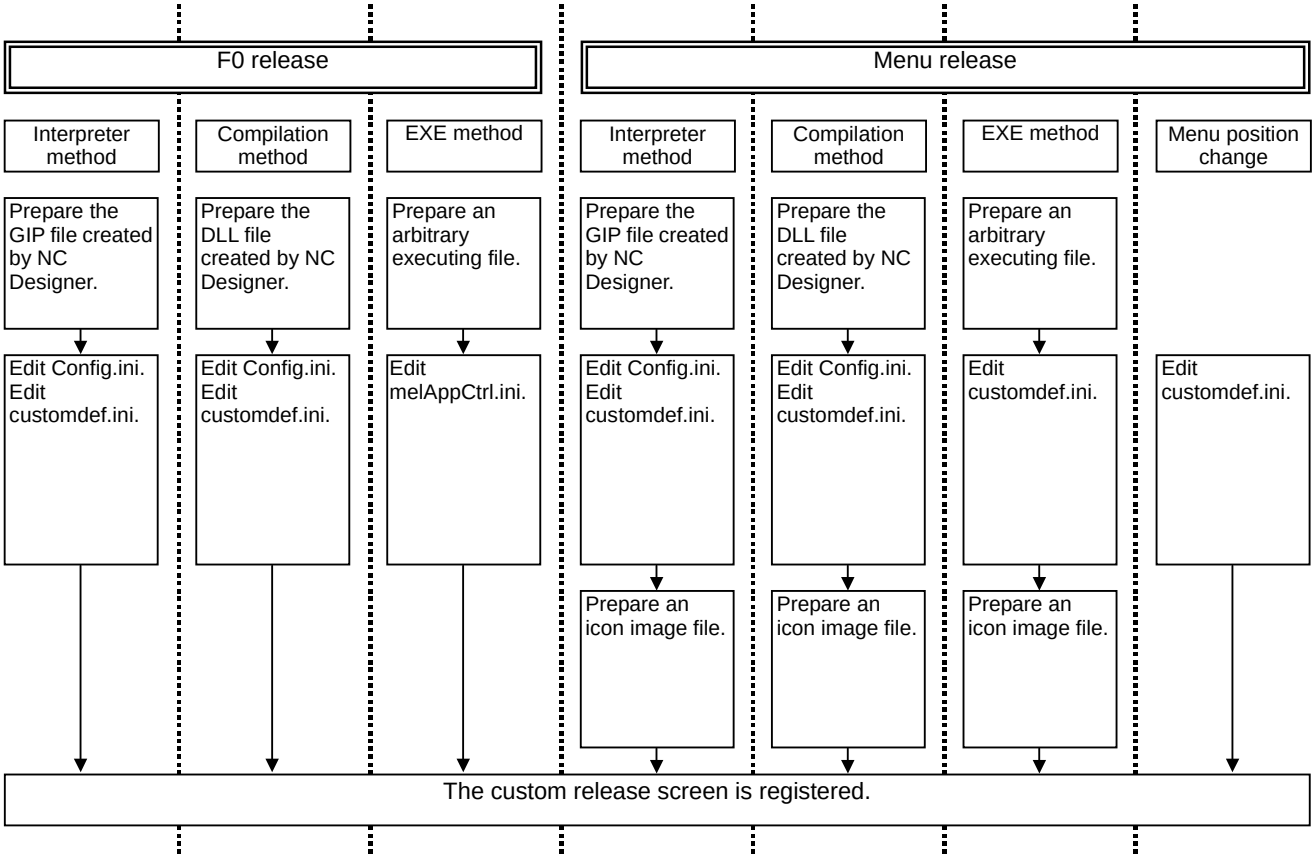
(Note 1) The storage folder is different according to the display unit used.
(Note 2) When FCU7-DA2xx is used as a display unit, the compact flash card is needed in the back of the display unit.
(Note 3) When M70/M70V/M700VS/E70 is used as a display unit, this is stored in a folder by using SETUP INSTALLER. For SETUP INSTALLER, refer to "Appendix.11 Installing Custom Data (M70/M70V/M700VS/E70)".

The necessary file of each method is as follows.

Release method	Config.ini	melAppCtrl.ini	customdef.ini	*.jpg	*.DLL	*.GIP	*.EXE
Interpreter method (F0 release)	○	-	○	-	-	○	-
Compilation method (F0 release)	○	-	○	-	○	-	-
Executing file registration method (F0 release)	-	○	-	-	-	-	○
Interpreter method (Menu release)	○	-	○	○	-	○	-
Compilation method (Menu release)	○	-	○	○	○		-
Executing file registration method (Menu release)	-	-	○	○	-	-	○
Changed the arrangement of the main menu	-	-	○		-	-	-

○…Necessary to prepare or edit -…Not necessary to prepare or edit

15.6.3 Development Procedure of Custom Release S/W



15.6.4 F0 Release

In the F0 release, the screen of the custom release created with NC Designer or the execution file originally prepared can be registered to the function key.

15.6.4.1 Interpreter Method

To register the interpreter method data to the function key, it is necessary to edit Config.ini and customdef.ini. The customdef.ini has to be described by UNICODE text.

15.6.4.1.1 Config.ini

Example of setting

```
[INTERPRETER]
RUN=2                          ;<- Set 2 because PROJECT is registered up to 02.
PROJECT01=PANEL.GIP            ;<- When registering the custom release screen
PAGE_OFFSET01=6500             ;<- The offset No. is from 6000 to 7000.
PROJECT02=WINDOW.GIP           ;<- When registering the custom release window
PAGE_OFFSET02=8000             ;<- The setting range of offset No. is from 8000 to 9700.
```

Edit the following item of the [INTERPRETER] section.

Key name	Details
RUN	● This sets the number of projects executed by the interpreter method. ● The number of projects which can be registered is up to ten. ● Only one project can be registered when the project macro is used. ● Up to 256 screens and windows can be created for one project. Setting range : 0 to 10
PROJECTXX (XX = 01 to 10)	● This sets the GIP file of the startup project by full path. **Ex.)** <When using M700/M700VW (FCU7-DA3xx/4xx/6xx)> PROJECT01=D:\custom\Test.GIP <When using M700 (FCU7-DA2xx)> PROJECT01=\Memory Card\Custom\Test.GIP <When using M70/M70V/M700VS/E70 unit> PROJECT01=Test.GIP **(Note 1)** Number the end of the key name from 01 sequentially. **(Note 2)** If the GIP file which does not exist in PROJECT is set, the screen and the window included in the GIP file registered after set cannot be displayed.
PAGE_OFFSETXX (XX = 01 to 10)	● This is an offset value added to the screen No. in each project. "Screen No. in the project + Offset value" is a screen No. specified when the screen is changed between different projects. ● Use the offset No. within the following ranges when the custom screen is registered to the function key. Setting range : 6000 to 7700 ● The number of function key which can be registered is four as follows. ● SFP ● F0 ● Window display  ● Window selection  ● Use the offset No. within the following ranges when the custom release window is registered to the main menu. Setting range : 8000 to 9700 **(Note 1)** When the screen (window) is created by NC Designer, number it in order of creation. **(Note 2)** Do not set the offset No. outside the above-mentioned setting range. **(Note 3)** Match the No. of the PROJECT key end and the No. of the PAGE_OFFSET key end. **(Note 4)** Leave space about 256 or more about the first offset No. and the second offset No. when two or more offset Nos. are registered. (Example: The first project : 7000, The second project : 7256, etc.)

Edit the following item of the [PROJECT] section when the project macro is used.

Key name	Details
CYCLIC_MACRO	● Set the GMC file of the startup project macro by full path. ● One project macro can be registered. **Ex.)** < When using M700/M700VW (FCU7-DA3xx/4xx/6xx)> CYCLIC_MACRO=D:\custom\Test\Macro\Test.GMC < When using M700 (FCU7-DA2xx)> CYCLIC_MACRO=\Memory Card\Custom\Test\Macro\Test.GMC < When using M70/M70V/M700VS/E70 unit > CYCLIC_MACRO=/Test/Macro/Test.GMC

15.6.4.1.2 customdef.ini

Example of setting

```
; ● When registering the screen (PANEL on NCDesigner) to the function key
; Set 6500 to the offset No. of F0 key
; The previously displayed screen displays screens that the offset No. is from 6500 to 6999.
[COFFSET]
NUM=1
FUNC_ID01=1
PANEL_OFFSET01=6500
PANEL_HOLD01=6500-6999
```

Edit the following item of the [COFFSET] section.

Key name	Details
NUM	This sets the number of registration of F0 release (excluding the executing file registration method). Setting range : 0 to 4 Default : 0
FUNC_IDXX (XX = 01 to 04)	This designates the function key registering the custom release screen. Setting range : 0 to 3 0 : SFP key 1 : F0 key 2 : Screen display key 3 : Screen selection key
PANEL_OFFSETXX (XX = 01 to 04)	This designates the offset No. of the custom release screen registered to the function key. Designate the offset No. registered in Config.ini. Setting range : 6000 to 7999
PANEL_HOLDXX (XX=01 to 04)	Specify the previously displayed custom screen, if you wish to display it again by inputting a function key. Use this key when a custom release screen is made up of more than one screen (panel), and you wish to display the previously displayed screen again. (Note 1) This key is enabled for NC Designer interpreter method or NC Designer compilation method. (Note 2) A window can not be held. (Note 3) Do not include an unnecessary character such as a space in the setting value. Setting range : Maximum number of characters = 64 (Specify the offset No. between 6000 and 7999) Specify the offset No. of the custom screen to be held as a previously displayed screen, using commas as a delimiter. There are two types of methods to specify the offset No., individual designation and range designation. The formats are as follows. Individual designation = (Offset No.),(Offset No.),(Offset No.),… Range designation = (Offset No. – Offset No.),… Default: Not hold Setting example : PANEL_HOLD01=6001,6003,6004-6008,6010,6050

(Note 1) When the NUM key is 1 or more, always set to keys without setting default value.
(Note 2) The character string after " ; " is judged to be a comment.
(Note 3) Do not insert the spaces before and behind " = " between the key and the setting value.

15.6.4.2 Compilation Method

To register the compilation method data in the function key, it is necessary to edit Config.ini and customdef.ini.
The customdef.ini has to be described by UNICODE text.

15.6.4.2.1 Config.ini

Example of setting

```
[MODULE]
NUM =2                          ;<- Set 2 because MODULE_NAME is registered up to 02.
MODULE_NAME01=WINDOWDLL.DLL     ;<- When registering the custom release screen
PAGE_OFFSET01=7000              ;<- The setting range of offset No. is from 6000 to 7700.
MODULE_NAME02=PANELDLL.DLL      ;<- When registering the custom release window
PAGE_OFFSET02=9000              ;<- The setting range of offset No. is from 8000 to 9700.
```

Edit the following item of the [MODULE] section.

Key name	Details
NUM	● This sets the number of projects executed by the compilation method. ● The number of projects which can be registered is up to three. ● Up to 256 screens and windows can be created for one project. Setting range : 0 to 3
MODULE_NAMEXX (XX = 01 to 03)	● For M700/M700VW, this sets the DLL file of the startup project by full path. **Ex.)** <When using M700/M700VW (FCU7-DA3xx/4xx/6xx) > PROJECT01=D:\custom\Test.DLL <When using M700 (FCU7-DA2xx) > PROJECT01=\Memory Card\Custom\Test.DLL ● For M70/M70V/M700VS/E70, this sets the project name of NC Designer. **Ex.)** PROJECT01=Test **(Note 1)** Number the end of the key name from 01 sequentially. **(Note 2)** If the file which does not exist in MODULE_NAME is set, the screen and the window included in the file registered after set cannot be displayed.
PAGE_OFFSETXX (XX = 01 to 03)	● This is an offset value added to the screen No. in each project. "Screen No. in the project + Offset value" is a screen No. specified when the screen is changed between different projects. ● Use the offset No. within the following ranges when the custom screen is registered to the function key. Setting range : 6000 to 7700 ● The number of function key which can be registered is four as follows. ● SFP ● F0 ● Window display  ● Window selection  ● Use the offset No. within the following ranges when the custom release window is registered to the main menu key. Setting range : 8000 to 9700 **(Note 1)** When the screen (window) is created by NC Designer, number it in order of creation. **(Note 2)** Do not set the offset No. outside the above-mentioned setting range. **(Note 3)** Match the No. of the MODULE_NAME key end and the No. of the PAGE_OFFSET key end. **(Note 4)** Leave space about 256 or more about the first offset No. and the second offset No. when two or more offset Nos. are registered. (Example: The first project : 7000, The second project : 7256, etc.)

15.6.4.2.2 customdef.ini

Refer to 15.6.4.1.2.

15.6.4.3 Switching of "Onboard" and "Execution File by F0 Release" by Bit Selection Parameter (#6451 bit0)

By the bit selection parameter "#6451 bit0 (Onboard ON)", whether to start the onboard or the execution file by the F0 release can be selected with the F0 key. When #6451 bit0 (Onboard ON) is 1, the onboard starts. When #6451 bit0 is 0, the executing file by F0 release starts. However, when the executing file is not registered even if #6451 bit0 (Onboard ON) is 0, the input of F0 key is ignored.

(Note 1) Refer to 15.6.4.1 and 15.6.4.2 for registration of the executing file to F0 key.
(Note 2) The setting of the executing file registration method is given to priority when both the interpreter/compilation method and the executing file registration method are registered.
(Note 3) For M700 CE and M70/M700VS/M70V/E70, this function is enabled only when standard HMI is displayed.

Refer to Appendix 9. for the executing file registration method.

15.6.5 Menu Release

In the menu release, the window of the custom release created with NC Designer or the execution file originally prepared can be registered in the main menu of the monitor screen, setup screen and edit screen.
Main menu contents of the monitor, setup and edit screen can be rearranged.

15.6.5.1 Interpreter Method

To register the interpreter method data to the function key, it is necessary to edit Config.ini and customdef.ini and to prepare the image displayed as the icon.
The customdef.ini has to be described by UNICODE text.

15.6.5.1.1 Config.ini

Refer to 15.6.4.1.1.

15.6.5.1.2 customdef.ini

Example of setting

```
; ● When adding the custom release window
; ● Set the custom release window "TEST" to the fourth menu from the left on the first page of the monitor screen.
; ● The panel update processing displaying the window of the custom open is set once every 200ms.
; ● When the window is shut, the instance is held.
; ● The display existence of the menu is acquired from TESTWIN.DLL.
[CMENU]
NUM=2
SCREEN_TYPE01=0
MENU_POS01=3
WINDOW_OFFSET01=8000
MENU_ENG01=TEST
SUMMARY_ENG01=TEST
MENU_JPN01=SHIKEN
SUMMARY_JPN01=SHIKEN
BG_REFRESH_TIME01=200
INSTANCE_HOLD01=1
MENU_STATE_DLL01=TESTWIN.DLL

; ● When adding the executing file
; ● Add the menu "calc" to the fifth menu from the left on the first page of the monitor screen,
; When the menu is pressed, "C:\WINDOWS\SYSTEM32\calc.exe" is executed.
; When the execution file starts, the update cycle of a standard screen is set once every 200ms.
SCREEN_TYPE02=0
MENU_POS02=4
WINDOW_OFFSET02=20000
EXECUTE02=C:\WINDOWS\SYSTEM32\calc.exe,calc,
MENU_ENG02=calc
SUMMARY_ENG02=calc
MENU_JPN02=dentaku
BG_REFRESH_TIME01=200
SUMMARY_JPN02=dentaku
```

Edit the following item of the [CMENU] section.

Key name	Details
NUM	Specify the number of custom release registration. Setting range : 0 to 50 Default : 0
SCREEN_TYPEXX (XX = 01 to 50)	Specify the screen where the menu is added. Setting range : 0 to 2 0: Monitor 1: Setup 2: Edit
MENU_POSXX (XX = 01 to 50)	Specify the menu position to register. Setting range : 0 to 29 0 to 9 : First page 10 to 19 : Second page 20 to 29 : Third page When other menu is registered at the specified menu position, an existing menu becomes invalid.
WINDOW_OFFSETXX (XX = 01 to 50)	● For the interpreter method and the compilation method Specify the displayed window No. The setting value is "the offset No. designated for PAGE_OFFSET of Config.ini + the window No. in the project (0 to 255)". Setting range : 8000 to 9999 ● For executing file registration method Specify the No. corresponding to the image file name displayed as the icon. Setting range : 20000 to 20099

<table>
<tr><th>Key name</th><th>Details</th></tr>
<tr><td>EXECUTEXX
(XX = 01 to 50)</td><td>Designate the started executing file.

The starting status of the executing file can be judged by setting the title bar character string and the class name of the window. As a result, a multiple start of the execution file can be controlled.

If both the title bar character string and the class names of the window are set, it is judged "The execution file is starting" when each requirement is met at the same time.

When the executing file has already started, the focus is set to the corresponding executing file.

The details of arguments are as shown below.

<table>
<tr><th>Argument</th><th>Details</th></tr>
<tr><td>1st argument</td><td>The file name of the starting executing file (including the folder name)
Default : Null character</td></tr>
<tr><td>2nd argument</td><td>The title bar character string of window referred to confirm the starting status of the executing file
Default : Nothing</td></tr>
<tr><td>3rd argument</td><td>The class name of window referred to confirm the starting status of the executing file
Default : Nothing</td></tr>
</table>
Setting range
All : Within 200 characters
File name : Within 100 characters
Window name, class name : Within 50 characters for each

* When two or more executing files corresponding to the condition exist, the executing file found first is operated.
* It is valid only for the executing file registration method.
The method is distinguished with the WINDOW_OFFSET key.</td></tr>
</table>

Key name	Details
MENU_YYYXX (XX = 01 to 50)	Specify character strings for the menu. Describe them within 7 one-byte characters. Set the division of the language to YYY referring to the following. ENG : English　JPN : Japanese DEU : German　FRA : French ITA : Italian　SPA : Spanish CHT : Chinese (traditional)　KOR : Korean POR : Portuguese　DUT : Dutch SWE : Swedish　HUN : Hungarian POL : Polish　CHS : Chinese (simplified) RUS : Russian　TUR : Turkish CZE : Czech When switched to a language to which character strings are not registered, English character strings is displayed by default. Setting range : Within 7 one-byte characters Default : Null character **(Note)** Adjust the character string display position with space.
SUMMARY_YYYXX (XX = 01 to 50)	Specify character strings for the outline column of the menu list. Describe them within 70 one-byte characters. The setting of YYY is similar to "MENU_YYYXX". When switched to a language to which character strings are not registered, English character strings is displayed by default. Setting range : Within 70 one-byte characters Default : Null character **(Note)** Adjust the character string display position with space.

Key name	Details
BG_REFRESH_TIMEXX (XX = 01 to 50)	● For the interpreter method and the compilation method Specify the update cycle of the panel displayed on the background is set while displaying the window of the custom release. When the custom release window is displayed, the update cycle of the panel is changed to the setting value. When closing, it returns to the origin. It is possible to set "Do not update" or until 0 to 10 seconds by each millisecond unit. Setting range : -1 to 10000 Default : 0 -1 : Do not update drawing 0 to 100 : Update at the highest cycle of the panel 100 to 10000 : Update with setting cycle ● For executing file registration method Specify the update cycle of a standard screen when the registered execution file starts. When the executing file is started, the update cycle of a standard screen is changed. When a standard screen moves from the background screen to an active screen, the setting of the update cycle (sleep time) is released. It is possible to set "Do not update", "Do not change" or until 0 to 1 second by each millisecond unit. Setting range : -1 to 1000 Default : 0 -1 : Do not update 0 : Do not change the update cycle 1 to 1000 : Changes to the set update cycle Out of range : Do not change the update cycle
INSTANCE_HOLDXX (XX = 01 to 50)	Specify whether to hold the instance when the window closes. If the instance is held, the window can be displayed with the closing status when the window will be opened next time. Even if the window is displayed on another screen, the window is displayed with the closing status last time. When the window is opened next time, the focus is placed on the focused control before the window was closed or the window. Setting range : 0 to 1 Default : 0 0: Do not hold at window close 1: Hold at window close * It is valid only for the interpreter method and the compilation method. The method is distinguished with the WINDOW_OFFSET key.

Key name	Details
MENU_STATE_DLLXX (XX = 01 to 50)	For M700/M700VW, specify the DLL file defined the function (MCAppGetMenuState()) which checks whether to display the menu by full path. **Ex.)** <When using M700/M700VW (FCU7-DA3xx/4xx/6xx) > MENU_STATE_DLL01=D:\custom\Test_MenuState.DLL <When using M700 (FCU7-DA2xx) > MENU_STATE_DLL01=\Memory Card\Custom\Test_MenuState.DLL Setting range : Number of capital letters of file path = 63byte When the DLL file is set, display (TRUE)/non-display (Excluding TRUE) of the menu is switched by the return value of MCAppGetMenuState() in the DLL. When this item is not set, the menu is unconditionally displayed. Default : Display the menu unconditionally * When the set DLL file does not exist or the MCAppGetMenuState() function is not defined in the set DLL file, the menu is not displayed. When the DLL file path of 63 bytes or more is set, it is judged that the key is invalid, and the menu is displayed. For M70/M70V/M700VS/E70, specify the name with project name of NC Designer following the function (MCAppGetMenuState()) which checks whether to display the menu. **Ex.)** MENU_STATE_DLL01=MCAppGetMenuStateTest

(Note 1) When the NUM key is 1 or more, always set to keys without setting default value.
(Note 2) The character string after " ; " is judged to be a comment.
(Note 3) Do not insert the spaces before and behind " = " between the key and the setting value. (For the menu character string and the outline character string, a space can be inserted by the right side of "=" to adjust the position.)

15.6.5.1.3 Icon Image

File name and size are shown below.
File name : "The values of WINDOW_OFFSETXX_OFF.jpg"
Size : 62 x 40
File format : JPEG
Store the created image file in the following folder.
Storage folder may differ according to the unit being used.
<When using M700/M700VW (FCU7-DA3xx/4xx/6xx) >
D:\Custom\img\
<When using M700(FCU7-DA2xx)>
\Memory Card\Custom\img\
<When using M70/M70V/M700VS/E70 unit >
/custom/

(Note 1) The image might not be correctly displayed when there is defect in the file name, the size, and the file format.
(Note 2) When M70/M70V/M700VS/E70 unit is used as a display unit, the image file is stored in a folder by using SETUP INSTALLER. For SETUP INSTALLER, refer to "Appendix.11 Installing Custom Data (M70/M70V/M700VS/E70)".

15.6.5.2 Compilation Method

To register the compilation method data to the main key, it is necessary to edit Config.ini and customdef.ini and to prepare the image displayed as the icon.
The customdef.ini has to be described by UNICODE text.

15.6.5.2.1 Config.ini

Refer to 15.6.4.2.1.

15.6.5.2.2 customdef.ini

Refer to 15.6.5.1.2.

15.6.5.2.3 Icon Image

Refer to 15.6.5.1.3.

15.6.5.3 Changing the Arrangement of the Main Menu

The main menu of the monitor, setup and edit screen can be permuted in easy-to-use the order.
To change the arrangement of the main menu, the customdef.ini must be edited.
The customdef.ini has to be described by UNICODE text.

15.6.5.3.1 customdef.ini

Example of setting

```
; ● When changing the position where an existing menu is displayed
 [MENU_CHANGE]
NUM=2
; "Edit" is set to the fifth menu of the edit screen.
SCREEN_TYPE01=2
MENU_POS01=4
CHG_SCREEN_ID01=301
; Delete the first menu of the edit screen.
SCREEN_TYPE02=2
MENU_POS02=0
CHG_SCREEN_ID02=0
```

Edit the following item of the [[MENU_CHANGE] section.

Key name	Details
NUM	Specify the number of the changing menu position registration. Setting range : 0 to 90 Default : 0
SCREEN_TYPEXX (XX = 01 to 90)	Specify the screen where the menu is added or changed. Setting range : 0 to 2 0: Monitor 1: Setup 2: Edit
MENU_POSXX (XX = 01 to 90)	Specify the menu position to register. Setting range : 0 to 29 0 to 9 : First page 10 to 19 : Second page 20 to 29 : Third page When other menu is registered at the specified menu position, an existing menu is invalid.
CHG_SCREEN_IDXX (XX = 01 to 90)	Select the ID No. of the menu to register (refer to supplementation 1) at the menu position set by the above-mentioned from the following. When 0 is set, the menu at the position set by the above-mentioned is invalid. Setting range : The setting range changes by SCREEN_TYPEXX as follows. SCREEN_TYPEXX : 0 Screen ID : 0,101 to 121 SCREEN_TYPEXX : 1 Screen ID : 0.201 to 212 SCREEN_TYPEXX : 2 Screen ID : 0.301 to 304

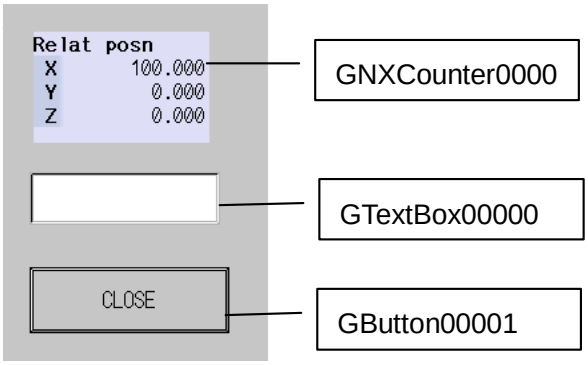
(Note 1) When the NUM key is 1 or more, always set to keys without setting default value.
(Note 2) The character string after " ; " is judged to be a comment.
(Note 3) Do not insert the spaces before and behind " = " between the key and the setting value.

15.6.5.4 Focus while the instance is held

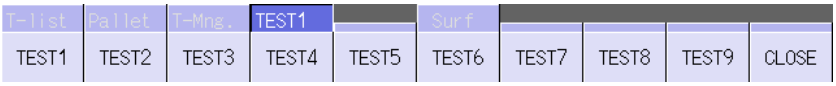
When the instance of the custom release window is held, the focus will stay with the control or the window where the focus was placed when the window was closed last time.
If the initialization while displaying the window is carried out under the specific control, move the focus to that control to execute initialization.

(EXAMPLE)
If the macro is created as follows to perform operations such as the key processing or initialization (character string setting of the menu) with GTextBox0000 control, the focus moves to GButton00001 control when the window is closed by touching GButton00001 control.
As the window is opened again, the key processing and initialization will not be carried out since the focus stays with GButton control.
Thus, move the focus to GTextBox00000 control at GButton00001-OnSetFocus().

・Custom Release Window



・Menu of Custom Release Window



Macro

```
$GTextBox00000-OnSetFocus
        GCSMenuSetMenuButtonLowerName_all(2000,0,1,
                " TEST1, TEST2, TEST3, TEST4, TEST5, TEST6, TEST7, TEST8, TEST9, CLOSE");
$End
$GTextBox00000-OnKeyPress
        long _lShiftKey;    'SHIFT key input status
        'The SHIFT key input status is maintained in the 0th bit of LUPARAM.
        _lShiftKey = LUPARAM & H1;
        if((LLPARAM == 121) && (_lShiftKey == 0))
                GCSMenuSendProcessID(2000,0,0);
        endif;
$End
$GButton00001-OnClick
GCSMenuSendProcessID(2000,0,0);
$End
```

```
$GButton00001-OnSetFocus
        GCSChangeActiveFocus(-1,"GTextBox00000");
$End
```

Return the focus to GTextBox00000

15.6.6 Limitation of Number of Project Registration

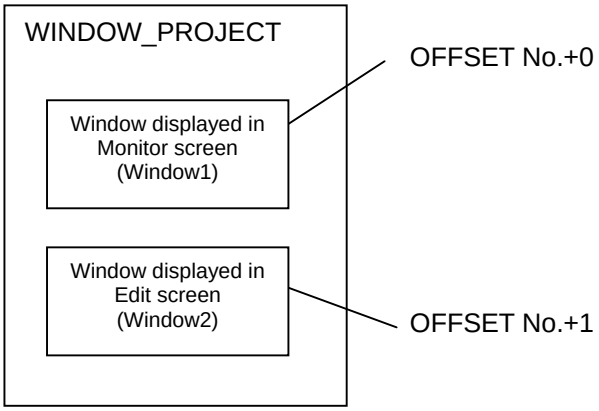
Projects of the interpreter method (GIP file) can be registered up to ten. Projects of the compilation method (DLL file) can be registered up to three.
At this time, correspond as follows so as not to exceed the maximum number.

- Register two or more windows in one project.
- Register two or more screens in one project.

15.6.6.1 Register Two or More Windows in One Project

Two or more windows used by the menu release are registered in one project.
Even if the screen where each window is displayed is different, the windows can be registered.
The example of registering the two windows for menu release in one project is as follows.

Ex.)

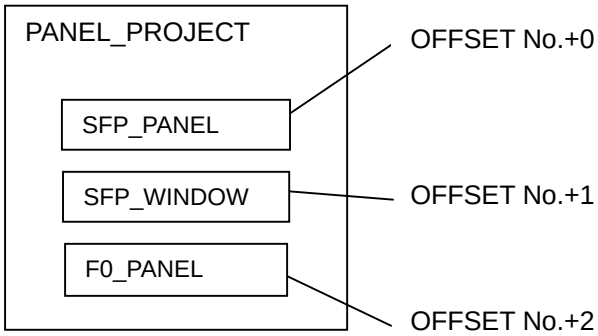


1. Create two windows in one project.
   (The window is number in order of creation: The offset No.+0, the offset No.+1 …)
2. Create the GIP file or the DLL file from the project created by step 1.
3. Register the file name and offset No. (It is assumed 8000 here.) created by step 2 to Config.ini.
4. Register to display the window offset No. 8000 on Moniter screen and the window offset No. 8001 on Edit screen by [CMENU] section of customdef.ini. (Refer to 15.6.5 for detail of the setting method.)
5. Window1 can be registered on the Moniter screen and Window2 can be registered in the Edit screen.

15.6.6.2 Register Two or More Screens in One Project

Two or more screens used by the F0 release are registered in one project.
The example of registering the screen for F0 key and the screen for SFP key in one project is as follows.

Ex.)



1. Create the screen when the F0 key is pressed and the screen when the SFP key is pressed in one project.
   (In the above example, the panel of F0 is registered as the third screen.)
2. Create the GIP file or the DLL file from the project created by step 1.
3. Register the file name and offset No. (It is assumed 6000 here.) created by step 2 to Config.ini.
4. Register to display the offset No. 6000 at SFP key and the offset No. 6002 at F0 key by [COFFSET] section of customdef.ini. (Refer to 15.6.4 for detail of the setting method.)
5. When the SFP key is pressed, SFP_PANEL is displayed, and when the F0 key is pressed, F0_PANEL is displayed.

15.6.7 About the Switch of Display/Non-display of the Menu by the Parameter

In this paragraph, the specification of function (MCAppGetMenuState()) which checks display/non-display of the menu is explained.

When display/non-display of the menu is switched by parameter, it is necessary to create the function MCAppGetMenuState() in the DLL designated by the [CMENU] section - "MENU_STATE_DLL" key of customdef.ini file and to judge non-display/display.
When the DLL file is specified in the "MENU_STATE_DLL" key, the specified DLL file is called, and display/non-display of the menu is switched by the return value. When the DLL file set in key does not exist or the MCAppGetMenuState() function is not defined in the DLL file, the menu is not displayed. When "MENU_STATE_DLL" is not set, the menu is unconditionally displayed.

	Function name	MCAppGetMenuState()		
Process	Returns whether to display the menu at the specified position.			
Argument	Type	Data name	I/O	Explanation
	const long	_lScreenType	I	Screen type (0 to 2) 0: Monitor 1: Setup 2: Edit
	const long	_lMenuPos	I	Menu position (0 to 29) 0 to 9 : 1st page 10 to 19 : 2nd page 20 to 29 : 3rd page
Return value	long (TRUE : Display/Not TRUE : Non-display)			
Details	Judge the display/non-display of the menu, and set TRUE : Display/Not TRUE : Non-display to the return value.			

Function model

```
//*********************************************************************************************
//
//          <Function name>             MCAppGetMenuState
//          <Function>                  Returns whether to display the menu at the specified position.
//
//                    [Argument]
//                              const long_lScreenType      (i) Screen type (0 to 2)
//                              const long_lMenuPos         (i) Menu position (0 to 29)
//                    [Return value]
//                              long        TRUE            : Menu display
//                                          Not TRUE        : Menu non-display
//
//
//*********************************************************************************************
long MCAppGetMenuState( const long _lScreenType, const long _lMenuPos )
{
          return TRUE;
}
```

15.6.8 Parameter

The list of the parameter is described in this paragraph.

No.	Name	Details	Setting range
#6451 bit0	Onboard on	Switch the onboard ON/OFF. 1 : Onboard ON 0 : Onboard OFF	0/1

15.6.9 Limitations

Common

- In the interpreter method, the key code flows out only to an active control.
- When the page offset No. of the interpreter method and the compilation method overlaps in Config.ini, the page offset No. of the compilation method is given to priority.
  When the page offset No. overlaps in the interpreter method or the compilation method, the page * offset No. previously set is given to priority.
- When two or more settings overlap to same function key or menu in customdef.ini, The setting described later is active.
- When two or more settings overlap to same key code in melAppCtrl, either setting is active.
- When the executing file registered by the executing file registration method is not displayed by full-screen, the standard screen is displayed forward to touch the standard screen which operates on the back ground. (The registered execution file is hidden behind the standard screen.)
- When using M700 CE and M70/M700VS/M70V/E70, switching of onboard / the custom application by bit selection parameter is enabled only when standard HMI is displayed.
- When using M70/M700VS/M70V/E70, the upper bound of the maximum resource data (.res) size is 2MB.

F0 release

- When the custom release screen or the execution file is registered in the function key where the process exists, the existing process is invalid.
- Two or more screens cannot be registered in one function key by F0 release other than the executing file registration method. Register in another key when two or more screens are displayed.
- When the custom release screen is created by NC Designer, two is standard about number of windows which can be opened at the same time on panel.

Menu release

- The custom release window cannot be added to the Diagnosis screen and the Maintenance screen.
- The arrangement of the main menu cannot be changed on the Diagnosis screen and the Maintenance screen.
- When the menu is added or deleted in the position where the main menu exists, the existing menu is overwritten.
- The main menu of other screens cannot be set by changing the arrangement of the main menu.
  (Ex. :The main menu of Edit screen cannot be set to Monitor screen.)
- Neither an existing main menu name nor the icon image are changed.
- When the menu of manual operation MST or the counter set is deleted by changing the main menu, the function to display pop-up with the address key is invalid.
- Two or more custom release windows cannot be displayed at the same time.
- Display the 3D check screen and the custom release window at the position where both do not overlap or where 3D screen is completely hidden in the custom release window when these are displayed at the same time.
- When an illegal file path is set in the executing file registration method, the menu is registered, but there is no reaction even if the menu is pressed.
- When the panel renewal is stopped to set BG_REFRESH_TIME to -1, the data displayed in the panel is not guaranteed. Take measures to display the registered window by full-screen, etc.
- When the display/non-display of the menu added by conditions is switched, the setting is not active until restarting the standard screen even if the corresponding conditions are changed.
- When "INSTANCE_HOLD" is set to "1", only instance of the custom window defined by WINDOW_OFFSET in customdef.ini is stored.

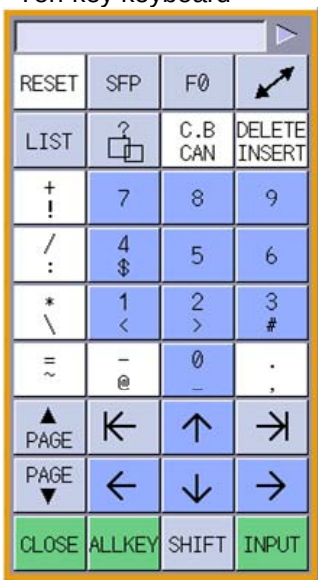
15.7 M70 S/W Keyboard

15.7.1 Outline

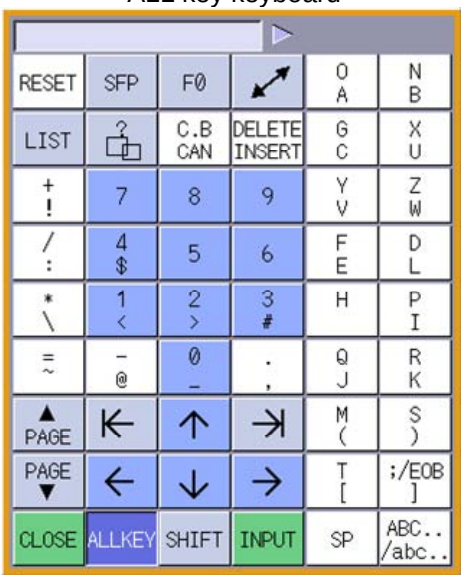
M70 custom screen can display the following S/W keyboard window.
This S/W keyboard window can be laid out on the left end, middle or right end of the screen. When the [ALLKEY] button is pressed, the keyboard is switched between the ten-key and ALL key display.
This section describes how to use the S/W keyboard window on M70 custom screen.

・M70 software keyboard

<Ten-key keyboard>



< ALL key keyboard>



15.7.2 Function Specifications

Defining the following items is required to use the S/W keyboard, as the M70 S/W keyboard window has been created as a window of NC Designer.

・Monitor screen's offset No. 1000
・S/W keyboard window's page No. 27

By specifying the screen No. as "1027", each of the S/W keyboard functions can be activated.

<Interpreter method>

	Description	Function name	Function No.	Setting
1	Open the S/W keyboard	GCSCreateGWindow()	-	-
2	Close the S/W keyboard	GCSCloseGWindow()	-	-
3	Set the S/W keyboard display position	GCSUser()	4193	0: Ten-key keyboard on the left end 1: Ten-key keyboard on the middle 2: Ten-key keyboard on the right end 3: ALL key keyboard on the left end 4: ALL key keyboard on the middle 5: ALL key keyboard on the right end
4	Set the status of entry area (Enabled/Disabled/ Password entry)	GCSUser()	4189	0: Disabled 1: Enabled 2: Password entry
5	Get the status of entry area (Enabled/Disabled/ Password entry)	GCSUser()	4194	Entry area's status (0: Disabled 1: Enabled 2: Password entry)
6	Clear the entry area	GCSUser()	4190	0 (fixed)
7	Set a character string to the entry area	GCSUser()	4196	A character string to enter
8	Get a character string from the entry area	GCSUser()	4197	A character string in the entry area of S/W keyboard window
9	Display the S/W keyboard in the foreground	GCSUser()	4195	0 (fixed)

<Compilation method>

	Description	Function name	Event message	Setting
1	Open the S/W keyboard	GCSCreateGWindow()		
2	Close the S/W keyboard	GCSDeleteChild()		
3	Set the S/W keyboard display position	GCSUser()	USNX_CHANGESWKEYPOS	0: Ten-key keyboard on the left end 1: Ten-key keyboard on the middle 2: Ten-key keyboard on the right end 3: ALL key keyboard on the left end 4: ALL key keyboard on the middle 5: ALL key keyboard on the right end
4	Set the status of entry area (Enabled/Disabled/ Password entry)	GCSUser()	USNX_SETSWKEYINPUTACTIVE	0: Disabled 1: Enabled 2: Password entry
5	Get the status of entry area (Enabled/Disabled/ Password entry)	GCSUser()	USNX_GETSWKEYINPUTACTIVE	Entry area's status (0: Disabled 1: Enabled 2: Password entry)
6	Clear the entry area	GCSUser()	USNX_CLEARSWKEYINPUTDATA	0 (fixed)
7	Set a character string to the entry area	GCSUser()	USNX_SETSWKEYINPUTDATA	A character string to enter
8	Get a character string from the entry area	GCSUser()	USNX_GETSWKEYINPUTDATA	A character string in the entry area of S/W keyboard window
9	Display the S/W keyboard in the foreground	GCSUser()	USNX_MOVELASTWINDOW	0 (fixed)

15.7.3 M70 Programming Method

15.7.3.1 Open S/W Keyboard Window

To open the S/W keyboard window, you need to set the entry area of the S/W keyboard window to either enabled, disabled or password mode. For how to set, refer to "7 Set the Status of Entry Area of S/W Keyboard Window ". The default S/W keyboard is the ten-key type and is displayed on the left end. For how to set the keyboard, refer to "Set S/W Keyboard Position".

Example

Interpreter Method

Display the S/W keyboard window when the button control (GButton00000) is pressed
(Set the entry area of the S/W keyboard window to disabled)

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 0);
   GCSCreateGWindow (1027);              'Open the S/W keyboard window
   GCSUser (1027, 4189, mem);            'Set the entry area to disabled
   GMEMDelete(mem);
$End
```

Compilation Method

Display the S/W keyboard window using the GCSCreateGWindow function.

```
// Open the S/W keyboard window
long _lStatus = GCSCreateGWindow( GCSGetScreen(GetGBaseObject()),1027 );
if ( _lStatus == TRUE )
{
        // Get the S/W keyboard window object
        GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),
                (unsigned short)(1027 + 1) );
        GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
        // Set the S/W keyboard entry area to disabled
        GCSUser( _ gpWinPanelObj, USNX_SETSWKEYINPUTACTIVE, 0 );
}
```

15.7.3.2 Close S/W Keyboard Window

Example

Interpreter Method

Close the S/W keyboard window when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GCSCloseGWindow (1027);                ‘Close S/W keyboard window
$End
```

Compilation Method

Close the S/W keyboard window using the GCSDeleteChild function.

```
GBaseObject *_gcFrameObj;
_gcFrameObj = GCSGetFrame( GCSGetScreen(GetGBaseObject()) );

GBaseObject *_pFrmChild;
_pFrmChild = GCSGetChild( _gcFrameObj, (unsigned short)(1027 + 1) );
if (_pFrmChild != NULL)
{
        // Close the S/W keyboard window
        GCSDeleteChild( _gcFrameObj, _pFrmChild );
}
```

15.7.3.3 Set S/W Keyboard Position

15.7.3.3.1 Set the initial display position of the S/W keyboard window

Example

Interpreter Method

Display the ten-key S/W keyboard window in the screen's middle when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 1);
   GCSCreateGWindow (1027);              'Open the S/W keyboard window
   GCSUser (1027, 4193, mem);            'Display the ten-key S/W keyboard window
                                                                  in the screen's middle.
   GCSUser (1027, 4189, 0);              'Set the entry area to disabled
GMEMDelete(mem);
$End
```

Compilation Method

When the GCSUser function (Function: USNX_CHANGESWKEYPOS) is used after the GCSCreateGWindow function, the S/W keyboard window can be opened at the specified display position.

```
// Open the S/W keyboard window
long _lStatus = GCSCreateGWindow( GCSGetScreen(GetGBaseObject()),1027 );
if ( _lStatus == TRUE )
{
        // Get the S/W keyboard window object
        GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),
                                                                  (unsigned short)(1027 + 1) );
        GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
        // Change the display position of the S/W keyboard
        GCSUser( _ gpWinPanelObj, USNX_CHANGESWKEYPOS, 1 );
        // Set the S/W keyboard entry area to disabled
        GCSUser( _ gpWinPanelObj, USNX_SETSWKEYINPUTACTIVE, 0 );
}
```

15.7.3.3.2 Change the S/W keyboard window position arbitrarily while the window is being displayed

Example

Interpreter Method

Display the ten-key S/W keyboard window in the screen's middle when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 1);
   GCSUser (1027, 4193, mem);  'Display the ten-key S/W keyboard window in the screen's middle
   GMEMDelete(mem);
$End
```

Compilation Method

Change the display position of the S/W keyboard window using the GCSUser function (Function: USNX_CHANGESWKEYPOS).

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),
                                                                    (unsigned short)(1000 + 27 + 1) );
if( gpWindowObj != NULL )
{
     GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
     // Change the display position of the S/W keyboard
     GCSUser( _ gpWinPanelObj, USNX_CHANGESWKEYPOS, 1 );
}
```

15.7.3.4 Set the Status of Entry Area of S/W Keyboard Window

The status of the entry area of the S/W keyboard window can be set to disabled, enabled or password mode.

Disabled : When a key is input from the S/W keyboard window, the key is passed to the control where the focus is located.

Enabled : When a key is input from the S/W keyboard window, the entered key is displayed in the entry area of the S/W keyboard window.
(Note) If you wish to display an entered key in the entry area of the S/W keyboard window, refer to "3.4.1 Display the entered characters on the entry area of S/W keyboard window ".

Password display : When a key is entered from the S/W keyboard window, "*" is displayed in the entry area of the S/W keyboard window.

Example

Interpreter Method

Set the entry status of the S/W keyboard window to enabled when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 1);
   GCSUser (1027, 4189, mem);              'Set the entry area to enabled
   GMEMDelete(mem);
$End
```

Compilation Method

Set the status of the entry area of the S/W keyboard window using the GCSUser function (Function: USNX_ SETSWKEYINPUTACTIVE).

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),    (unsigned short)(1027 + 1) );
if( gpWindowObj != NULL )
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      // Set the S/W keyboard entry area to enabled or disabled
      long _lStatus = 1;
      GCSUser( _ gpWinPanelObj, USNX_SETSWKEYINPUTACTIVE, (long)& _lStatus);
}
```

15.7.3.4.1 Display the entered characters on the entry area of S/W keyboard window

If the entry area of the S/W keyboard window is enabled, an entered key is not displayed in the entry area even when a key is entered from the S/W keyboard window.
However, by adding GCSKeyPress() to the OnKeyPress function of a control placed on a panel, a character string is displayed in the entry area of the S/W keyboard window.

Example

Interpreter Method

```
$GTextBox00000-OnKeyPress
   GCSKeyPress(1027, LLPARAM, LUPARAM)          'Pass a key to the S/W keyboard window
endif
```

* Use the virtual key code (LLPARAM) and key state (LUPARAM) given to KeyPress as the arguments of the GCSKeyPress function.

Compilation Method

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),    (unsigned short)(1027 + 1) );
// If the S/W keyboard window object has been gotten (if the S/W keyboard window is displayed)
if(gpWindowObj != NULL)
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      // Pass a key to the S/W keyboard window
      GCSKeyPress ( _ gpWinPanelObj, lLParam, lUParam);
}
```

15.7.3.4.2 Process when the [INPUT] key is pressed

If the entry area of the S/W keyboard window is enabled, a value in the entry area is not automatically set in the control placed on a panel even when the [INPUT] key on the keyboard window is pressed. To make this enabled, it is required to get the value using the OnKeyPress function of the control.
By adding the below process to a control placed on a panel, a value in the entry area of the S/W keyboard window can be set.

Example

Interpreter Method

```
$GTextBox00000-OnKeyPress
 if(LLPARAM == 13)
  GMEM mem;
  STRING strStat;
  mem = GMEMCreate("GETINPUT", 140);
  GCSUser (1027, 4197, mem);              'Get a character string from the entry area
                                                                of the S/W keyboard window
  strStat = GMEMGetString(mem,0);
  GCSTextboxSetString(-1,"GTextBox00000",strStat);
  GMEMSetLong(mem, 0, 0);
  GCSUser (1027, 4190, mem);              'Clear the entry area of the S/W keyboard window
  GMEMDelete(mem);
  GCSChangeActiveFocus(-1,"GTextBox00000");     'Move the focus to GTextBox00000
 endif
$End
```

Compilation Method

```
if(lLParam == GK_RETURN)
{
      char _szData[256];
      memset( _szData, 0 , sizeof(_szData) );

      // Get the S/W keyboard window object
      GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),
                                                                (unsigned short)(1027 + 1) );
      // If the S/W keyboard window object has been gotten (if the S/W keyboard window is displayed)
      if(gpWindowObj != NULL)
      {
            GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
            // Get a value from the entry area of the S/W keyboard
            GCSUser( _ gpWinPanelObj, USNX_ USNX_GETSWKEYINPUTDATA, (long) _szData);
      }
}
```

15.7.3.5 Get the Entry Area Status of S/W Keyboard Window

Example

Interpreter Method

Get the status of the entry area of the S/W keyboard window in Stat when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   LONG Stat;
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GCSUser (1027, 4194, mem);        'Get the status of the entry area of the S/W keyboard window
   Stat = GMEMGetLong(mem, 0);
   GMEMDelete(mem);
$End
```

Compilation Method

Get the status of the entry area of the S/W keyboard window using the GCSUser function (Function: USNX_ GETSWKEYINPUTACTIVE).

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),    (unsigned short)(1027 + 1) );
if( gpWindowObj != NULL )
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      Long _lStatus = 0;
      // Get the status (enabled or disabled) of the S/W keyboard entry area
      GCSUser( _ gpWinPanelObj, USNX_GETSWKEYINPUTACTIVE, (long)&_lStatus );
}
```

15.7.3.6 Clear S/W Keyboard Entry Area

Example

Interpreter Method

Clear the entry area of the S/W keyboard window when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 0);
   GCSUser (1027, 4190, mem);              'Clear the entry area of the S/W keyboard window
   GMEMDelete(mem);
$End
```

Compilation Method

Clear the entry area of the S/W keyboard window using the GCSUser function (Function: USNX_ CLEARSWKEYINPUTDATA).

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),
(unsigned short)(1027 + 1) );
if( gpWindowObj != NULL )
{
GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
// Clear the entry area of the S/W keyboard
GCSUser( _ gpWinPanelObj, USNX_CLEARSWKEYINPUTDATA, 0 );
}
```

15.7.3.7 Set Character String in S/W Keyboard Entry Area

Example

Interpreter Method

Set a character string in the entry area of the S/W keyboard window when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 70);
   GMEMSetString(mem, 0, "abcdef");
   GCSUser (1027, 4196, mem);                ‘Set a character string in the S/W keyboard window
   mem = GMEMDelete("TESTMEM ");
   GMEMDelete(mem);
$End
```

Compilation Method

Set a character string in the entry area of the S/W keyboard window using the GCSUser function (Function: USNX_ SETSWKEYINPUTDATA).

```
// Character string to be set
char _szData[256];
memset( _szData, 0 , sizeof(_szData) );
_szData = "ABCDE"

// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),    (unsigned short)(1027 + 1) );
if(gpWindowObj != NULL)
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      // Set a character string in the S/W keyboard entry area
      GCSUser( _ gpWinPanelObj, USNX_SETSWKEYINPUTDATA, (long)_szData);
}
```

15.7.3.8 Get Character String from S/W Keyboard Entry Area

Example

Interpreter Method

Get a character string from the entry area of the S/W keyboard window when the button control (GButton00000) is pressed.

```
$GButton00000-OnClick
   GMEM mem;
   STRING Data;
   mem = GMEMCreate("TESTMEM", 140);
   GCSUser (1027, 4197, mem);          'Get a character string from the entry area of the S/W keyboard window
   Data = GMEMGetString( mem, 0);
   GMEMDelete(mem);
$End
```

Compilation Method

Get a character string displayed on the entry area of the S/W keyboard window using the GCSUser (Function: USNX_ GETSWKEYINPUTDATA).

```
char _szData[256];
memset( _szData, 0 , sizeof(_szData) );

// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),   (unsigned short)(1027 + 1) );
// If the S/W keyboard window object has been gotten (if the S/W keyboard window is displayed)
if(gpWindowObj != NULL)
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      // Get a value from the entry area of the S/W keyboard
      GCSUser( _ gpWinPanelObj, USNX_ USNX_GETSWKEYINPUTDATA, (long) _szData);
}
```

15.7.3.9 Display S/W Keyboard Window in the Foreground

If the display position of the S/W keyboard window is overlapped with that of the other window, the other window is displayed in the foreground when the other window is displayed or touched. In this case, the S/W keyboard window is displayed in the background of the other window. However, the S/W keyboard window is again displayed in the front by adding the following process.

Example

Interpreter Method

Display the S/W keyboard window in the foreground when the button control (GButton00000) is pressed

```
$GButton00000-OnClick
   GMEM mem;
   mem = GMEMCreate("TESTMEM", 4);
   GMEMSetLong(mem, 0, 0);
   GCSUser (1027, 4195, mem);          'Display the S/W keyboard window in the foreground
   GMEMDelete(mem);
$End
```

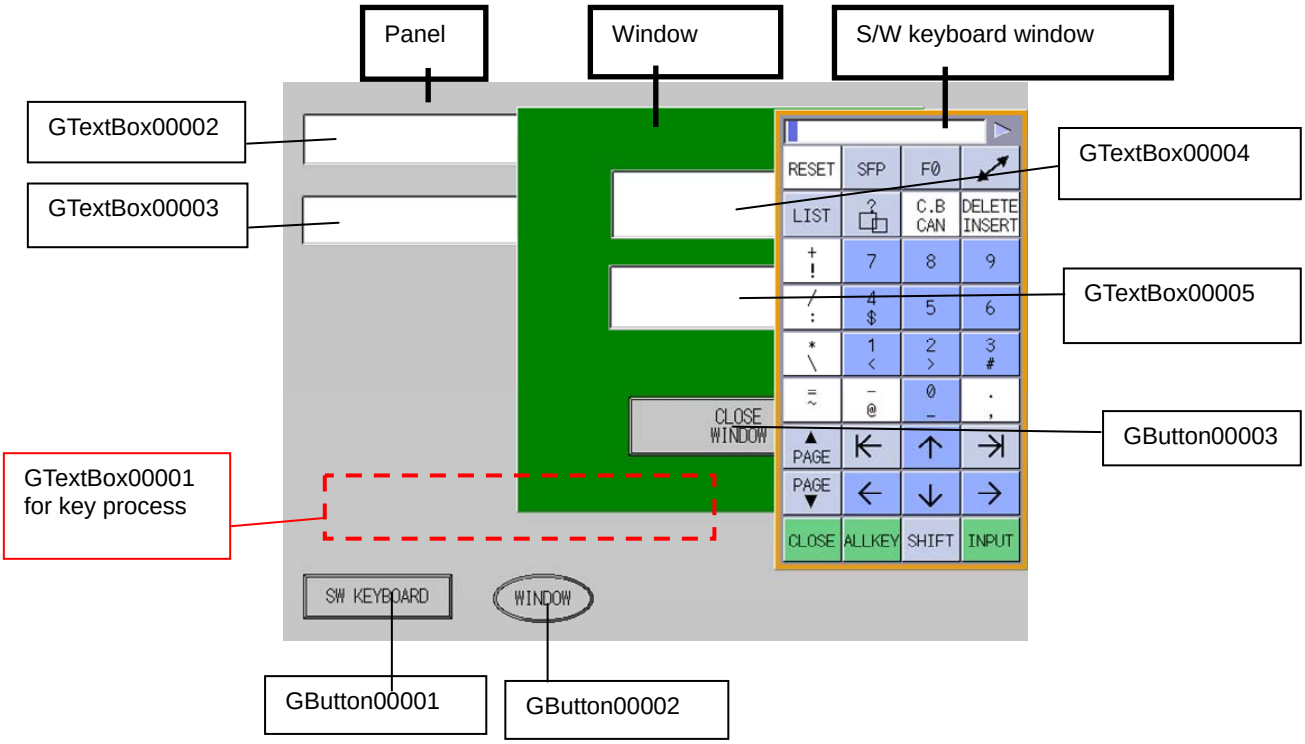
Compilation Method

Display the S/W keyboard window in the foreground using the GCSUser function (Function: USNX_MOVELASTWINDOW).

```
// Get the S/W keyboard window object
GBaseObject *_gpWindowObj = GCSGetWindow( (GCSGetScreen(GetGBaseObject())),    (unsigned short)(1027 + 1) );
// If the S/W keyboard window object has been gotten (if the S/W keyboard window is displayed)
if(gpWindowObj != NULL)
{
      GBaseObject *_gpWinPanelObj = GCSGetPanel( _gpWindowObj );
      // Display S/W keyboard in the foreground when the keyboard is not displayed in the foreground
      GCSUser( _ gpWinPanelObj, USNX_MOVELASTWINDOW, 0);
}
```

15.7.4 Example

Screen Configuration



Action

(1) When GButton00001 on the panel is pressed, the S/W keyboard window is displayed.

(2) The S/W keyboard window is displayed always in the foreground.

(3) When GButton00002 on the panel is pressed, the window is displayed.

(4) When GButton00003 on the window is pressed, the window is closed.

(5) When a key is entered on the S/W keyboard window while the window is not displayed, the entered character is displayed on the entry areas of both GTextBox00002 and the S/W keyboard window.

(6) When the [INPUT] key is pressed while the window is not displayed, the character string in the entry area of the S/W keyboard window is displayed on GTextBox00003.

(7) When a key is entered on the S/W keyboard window while the window is displayed, the entered character is displayed on the entry areas of both GTextBox00004 and the S/W keyboard window.

(8) When the [INPUT] key is pressed while the window is displayed, the character string in the entry area of the S/W keyboard window is displayed on GTextBox00005.

Source Code

When a window is opened using F0 release, the focus moves to the window. When a key on the S/W keyboard window is touched, the focus moves to the S/W keyboard window, thus the key is not passed to the window.
To enter the key touched on the S/W keyboard window into a control on a window, create the TextBox control on a panel at first, on which a key process to a control on a window is carried out. The focus should be on the TextBox control created on the panel when the window is displayed. Get the key touched on the S/W keyboard window using the TextBox control for key process, and get the character string on the TextBox control using a control of window.

< Open the S/W keyboard window>
・The control GButton00001 on a panel
When a key on the S/W keyboard window is touched, the entered character is displayed on GTextBox00002, so the focus moves to GTextBox00002.

```
$GButton00001-OnClick
        GCSCreateGWindow(1027);                         'Display the S/W keyboard window
        GMEM mem;
        mem = GMEMCreate("TESTMEM", 4);
        GMEMSetLong(mem,0,0);
        GCSUser(1027,4190,mem);                 'Clear the entry area
        GMEMSetLong(mem,0,1);
        GCSUser(1027,4189,mem);                 'Set the entry area status to enabled
        GMEMDelete(mem);
        GCSChangeActiveFocus(-1,"GTextBox00002");       'Move the focus to GTextBox00002
$End
```

<Display S/W keyboard in the foreground>
When a window is touched while the S/W keyboard window is being displayed, the focus moves to the touched control, so the window is displayed in the foreground of the S/W keyboard window.
Therefore, display the S/W keyboard window in the forefront when the focus moves to each control of window.

```
$GTextBox00005-OnSetFocus
        GMEM mem;
        mem = GMEMCreate("TESTMEM", 4);
        GMEMSetLong(mem,0,0);
        GCSUser(1027,4195,mem);         'Display the S/W keyboard window in the foreground
        GMEMDelete(mem);
$End

$GTextBox00004-OnSetFocus
        GMEM mem;
        mem = GMEMCreate("TESTMEM", 4);
        GMEMSetLong(mem,0,0);
        GCSUser(1027,4195,mem);         'Display the S/W keyboard window in the foreground
        GMEMDelete(mem);
$End
```

<Open a window>
・The control GButton00002 on a panel
When a window is opened, the opened window is displayed in the foreground. Therefore, display the S/W keyboard window again in the foreground.
Move the focus to the TextBox control for key process.

```
$GButton00002-OnClick
        GCSCreateGWindow(1);
        GMEM mem;
        mem = GMEMCreate("TESTMEM", 4);
        GMEMSetLong(mem,0,0);
        GCSUser(1027,4195,mem);             'Display the S/W keyboard window in the foreground
        GCSUser(1027,4190,mem);                     'Clear the entry area
        GMEMDelete(mem);
        GCSChangeActiveFocus(-1,"GTextBox00001");           'Move the focus to GTextBox00001
$End
```

<Close a window>
・GButton00003 control on a window
After the window has been closed, the focus is moved to GTextBox00002.

```
$GButton00003-OnClick
        GCSCloseGWindow(1);
        GCSChangeActiveFocus(0,"GTextBox00002"); 'Move the focus to GTextBox00002
$End
```

<Key process to a control on a panel>
When the [INPUT] key is pressed, a value in the entry area of the S/W keyboard window is set in the GTextBox00003 control on a window.
The entered key is displayed on the entry area of the S/W keyboard window, thus the key is passed to the S/W keyboard window using GCSKeyPress.

```
$GTextBox00002-OnKeyPress
        if(LLPARAM == 13)
        STRING strStat;
        GMEM mem;
        mem = GMEMCreate("TESTMEM", 140);
        GCSUser(1027, 4197, mem);           'Get the character string from the entry area
        strStat = GMEMGetString(mem, 0);
        GCSTextboxSetString(-1,"GTextBox00003",strStat);      'Set a character string in
                                                                    GTextBox00003
       GMEMDelete(mem);
        endif;
       GCSKeyPress(1027,LLPARAM, LUPARAM);    'Pass a key to the S/W keyboard window
$End
```

<Enter a key in a window>
・The control GTextBox00001 on a panel (for key process)
A process to be carried out when a key is input to GTextBox00004 is added to the TextBox control for key process.
When the [INPUT] key is pressed, a value in the entry area of the S/W keyboard window is set in the GTextBox00005 control on a window.
In order to display the entered key on the entry area of the S/W keyboard window, the key is passed to the S/W keyboard window using GCSKeyPress.

```
$GTextBox00001-OnKeyPress
        if(LLPARAM == 13)
          STRING strStat;
          GMEM mem;
          mem = GMEMCreate("TESTMEM", 140);
          GCSUser(1027, 4197, mem);          'Get the character string from the entry area
          strStat = GMEMGetString(mem, 0);
          GCSTextboxSetString(1,"GTextBox00005",strStat);      'Set a character string
                                                                    in GTextBox00005
         GMEMDelete(mem);
        endif;
           GCSKeyPress(1027,LLPARAM, LUPARAM);     'Pass a key to the S/W keyboard window
$End
```

・The control TextBox00004 on a window
Get an entered character string from the TextBox control for key process by use of Timer of TextBox00004.
After getting the character string, move the focus back to the TextBox control for key process.

```
$GTextBox00004-OnTimer
       STRING strStat;
       GCSTextboxGetString(0,"GTextBox00001",strStat);      'Get a character string
                                                                 to GTextBox00001
       GCSTextboxSetString(-1,"GTextBox00004",strStat);     'Set a character string
                                                                 to GTextBox00004
       GCSChangeActiveFocus(0,"GTextBox00001");             'Move the focus to GTextBox00001
$End
```

16. Macro Function

This section describes the macro function.

16.1 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 Designer 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.

16.2 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.

16.2.1 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.

16.2.2 Screen Macro Execution Condition

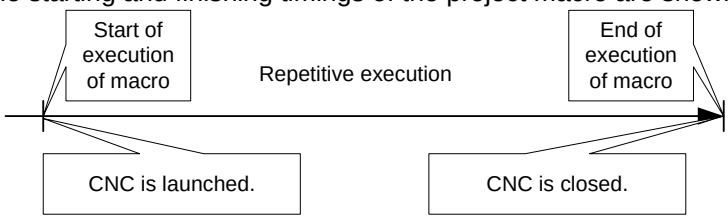
The screen macro is specified for controls existing on the screen. Execution of a screen macro corresponds to the callback function of the control. That is, the event held by each control is the execution condition of the macro.

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.

16.2.3 Macro Execution Timing

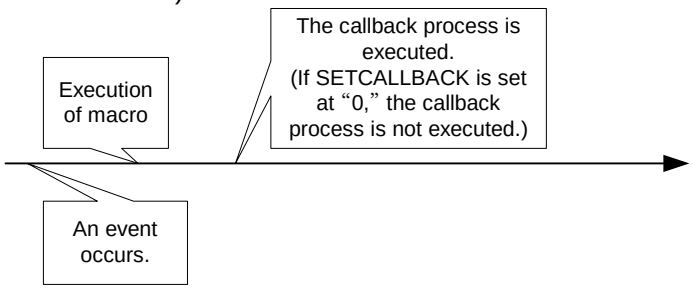
Project Macro

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).

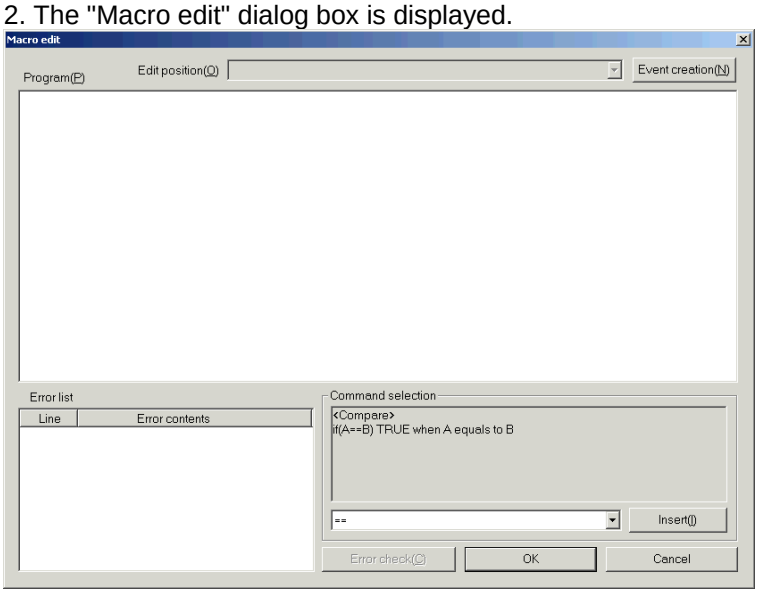


16.3 Macro Editing

16.3.1 Macro Editing

The procedure for registering or editing the project macro or screen macro is described here.

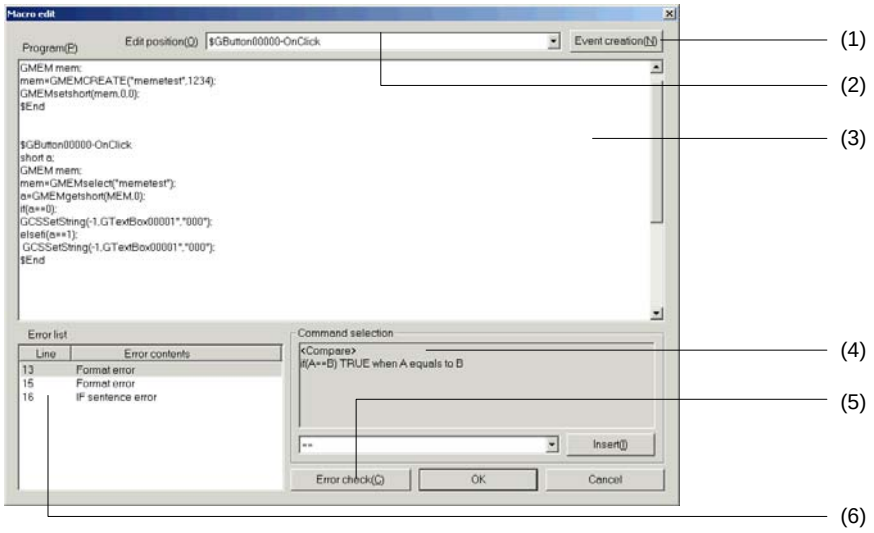
1. From the [Settings] menu, select [Project macro edit]/[Panel macro edit].
2. The "Macro edit" dialog box is displayed.



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

16.3.2 Macro Editing Dialog Box

The "Macro Edit" dialog box is described.



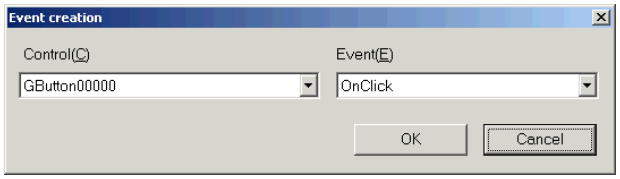
No.	Item	Description
(1)	Event creation	The "Event Creation" dialog box for adding an event in the program editing area is displayed. This part may not be operated when the project macro is edited.
(2)	Edit position	A list of macro events is displayed. Select a desired event from the combo box to move the cursor to the editing position of the selected event. The events are displayed in the alphabetical order in the combo box.
(3)	Program	Area for describing the macro program
(4)	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.
(5)	Error check	Syntax errors in the program description are checked for.
(6)	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.

16.3.3 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.

16.4 Macro Programming

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

16.4.1 Macro Editing Area

The maximum lines that can be edited in the macro editing area and the maximum number of characters per line are specified below.

Item	Description
Max. number of lines	10000
Max. number of characters per line	100 one-byte characters

16.4.2 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.

16.4.3 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.

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～#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～#100199 #200100～#200199 #300100～#300199 #400100～#400199 #500100～#500199 #600100～#600199 #700100～#700199 #800100～#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 No. * 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～#999	Used as variables for common variables "#400 to #999" shared among all part systems. These variables might become common variables for each part system depending on the setting of "#1052 MemVAl" and "#1303 V1comN". In this case, these become variables to handle common variables #400 to #900 for 1st part system.
#100400～#100999 #200400～#200999 #300400～#300999 #400400～#400999	Used as variables for common variable "#400 to #999" shared among all part systems. If the first one digit is not same as the variable No. but the last three digits are same, both of them can handle the same common variable among part systems. (Example: Both #100500 and #200500 handle the common variable #500 shared among the part systems.) These variables might become common variables for each part system depending on the setting of "#1052 MemVal" and "#1303 V1comN". In this case, common variable for all part systems can be handled. Common variables #400 to #999 for all part systems can be handled by "# + part system No. * 100000 + variable No.". If a part system No. 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 Designer 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.

- Refer to "13.4 Variable Commands" in "IB-1500072 MITSUBISHI CNC M700/M70 Series Programming Manual (Machining Center System)" or "13.8 Variable Commands" in "IB-1500057 MITSUBISHI CNC M700/M70 Series Programming Manual(Lathe System)" for related parameters.

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"

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

Type conversion

The followings are the available type conversion for a macro program. By converting the type, numbers read as a character string can be handled as a numerical value of integer type or floating point type.

Type conversion command	Example	Description
ATOL	B = ATOL(A)	Convert a character string type variable A to 32bit integer type and substitute it into B.
ATOF	B = ATOF(A)	Convert a character string type variable A to 64bit floating decimal point type and substitute it into B.

(Note 1) If a variable A includes a character other than numbers, a decimal point or a positive/negative sign, up to the number before the character will be valid.

EX.)

A="1a2";	B=ATOL(A);	The value B is 1
A="a12";	B=ATOL(A);	The value B is 0
A="-10";	B=ATOL(A);	The value B is -10
A="--10";	B=ATOL(A);	The value B is 0
A=" 10";	B=ATOL(A);	The value B is 10
A="12 3";	B=ATOL(A);	The value B is 12
A="12.3";	B=ATOF(A);	The value B is 12.3
A="1.2E1";	B=ATOF(A);	The value B is 12

(Note 2) A variable which includes a decimal point or exponent display cannot be converted with ATOL. They will be handled as same as other characters.

(Note 3) If a variable A is out of the range (-2147483648 to 2147483647)which cannot be handled with 32bit integer type or out of the range (valid real digit number 15) which cannot be handle with 64bit floating decimal point, an incorrect value will be substituted.

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 eight 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.

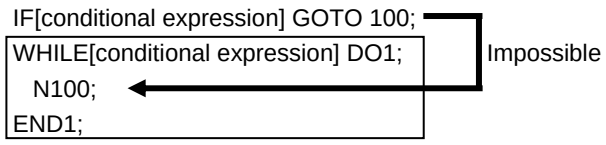
```
$GTextBox00000-OnKeyPress
  IF(#1 == #2) GOTO 100;
  #3 = #3 + 1;
  N99;
  #10 = #5
  N100;
  #1 =#10;
$End
```

(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
```

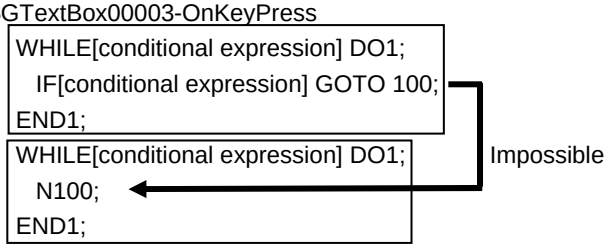
(3) It is not possible to branch into WHILE[conditional expression] D0i with GOTO statement.

$GTextBox00002-OnKeyPress



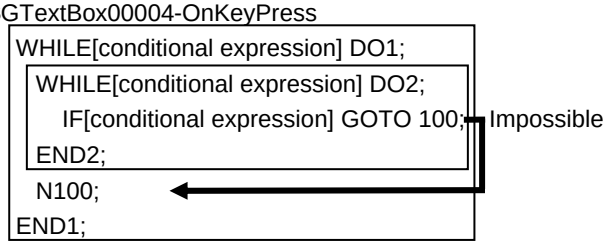
$End

$GTextBox00003-OnKeyPress



$End

$GTextBox00004-OnKeyPress

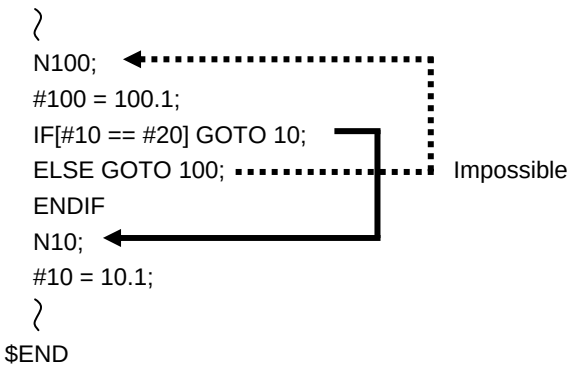


$End

(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

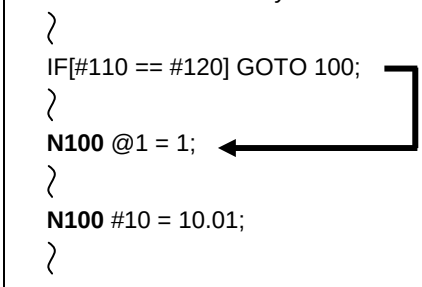


$END

(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



$END

16.5 Function Details

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

16.5.1 Normal Command

SetCallback	Select call-back processing execution
Syntax	SetCallBack(nIsExec);
Argument	(i) LONG nIsExec : Whether or not to execute call-back processing (Specify -1 for self screen.)
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(gmemName, nSize);
Argument	(i) STRING gmemName : 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(gmemName);
Argument	(i) STRING gmemName : 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 gmemName.
Example	Gets the global memory named "ABC". GMEM mem; mem = GMEMSelect("ABC");

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

GMEMSetChar	Set 8bit integer value to global memory
Syntax	GMEMSetChar(gmemName, nOffset, nData);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position (i) CHAR nData : setting value
Return value	None
Details	Sets 8bit integer value in the nOffset byte data from the beginning of the global memory with the specified name.
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(gmemName, nOffset);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position
Return value	8bit integer value
Details	Gets 8bit integer value in the nOffset byte data from the beginning of the global memory with the specified name.
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(gmemName, nOffset, nData);
Argument	(i) STRING gmemName : global memory (i) LONG nOffset : offset position (i) SHORT nDatavalue : setting
Return value	None
Details	Sets 16bit integer value in the nOffset byte data from the beginning of the global memory with the specified name.
Example	Sets 16bit integer (1000) in the "ABC" global memory's 4th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetShort(mem, 4, 1000);

GMEMGetShort	Get 16bit integer value from global memory
Syntax	GMEMGetShort(gmemName, nOffset);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position
Return value	16bit integer value
Details	Gets 16bit integer value in the nOffset byte data from the beginning of the global memory with the specified name.
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(gmemName, nOffset, nData);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position (i) LONG nData : setting value
Return value	None
Details	Sets 32bit integer value in the nOffset byte data from the beginning of the global memory with the specified name.
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(gmemName, nOffset);
Argument	(i) STRING gmemName : global memory name (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 global memory with the specified name.
Example	Gets 32bit integer in the "ABC" global memory's 4th byte position from the top. LONG Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetLong(mem, 4);

GMEMSetDouble	Set 32bit floating point value in global memory
Syntax	GMEMSetDouble(gmemName, nOffset, fData);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position (i) DOUBLE fData : setting value
Return value	None
Details	Sets 32bit floating point value in the nOffset byte data from the beginning of the global memory with the specified name.
Example	Sets 32bit floating point value (0.1234) in the "ABC" global memory's 4th byte position from the top. GMEM mem; mem = GMEMSelect("ABC"); GMEMSetDouble(mem, 4, 0.1234);

GMEMGetDouble	Get 32bit floating poing value from global memory
Syntax	GMEMGetDouble(gmemName, nOffset);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position
Return value	32bit floating point value
Details	Gets 32bit floating point value in the nOffset byte data from the beginning of the global memory with the specified name.
Example	Gets 32bit floating point value (0.1234) in the "ABC" global memory's 4th byte position from the top. DOUBLE Data; GMEM mem; mem = GMEMSelect("ABC"); Data = GMEMGetDouble(mem, 4);

GMEMSetString	Set character string in global memory
Syntax	GMEMSetString(gmemName, nOffset, strData);
Argument	(i) STRING gmemName : 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 data from the beginning of the global memory with the specified name.
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(gmemName, nOffset);
Argument	(i) STRING gmemName : global memory name (i) LONG nOffset : offset position
Return value	Character string
Details	Gets the character string in the nOffset byte data from the beginning of the global memory with the specified name.
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. 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 screen No. 10

```
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(10, "GNXMenu00000", 1, nMenuLoop,
      strMenu) ;
   nMenuLoop = nMenuLoop + 1 ;
NEXT
```

16.5.2 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 screen No.10 to 1. GCSSetBlinkEnableStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetBlinkEnableStatus(10, "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 screen No.10 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 screen No.10 in Stat. LONG Stat; Stat = GCSGetBlinkStatus(10, "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 screen No.10 to 1. GCSSetBlinkType(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetBlinkType(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSSetCaptionScrollEnable(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollEnable(10, "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 screen No.10. GCSStartCaptionScroll(10, "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 screen No.10. GCSStopCaptionScroll(10, "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 screen No.10. GCSPauseCaptionScroll(10, "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 screen No.10. GCSRestartCaptionScroll(10, "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 screen No.10 in Stat as to whether or not it goes out of the bounds. LONG Stat; Stat = GCSIsCaptionOutOfBounds(10, "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 screen No.10 to 10. GCSSetCaptionScrollDelayTime(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollDelayTime(10, "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 screen No.10 to 10. GCSSetCaptionScrollRefreshTime(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollRefreshTime(10, "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 screen No.10 to 10. GCSSetCaptionScrollMovementValue(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollMovementValue(10, "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 screen No.10 to 1. GCSSetCaptionScrollStartPosition(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollStartPosition(10, "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 screen No.10 to 100. GCSSetCaptionScrollPosition(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSGetCaptionScrollPosition(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSIsCaptionScrollFinish(10, "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 screen No.10 panel. GCSGEShowPanel(10);

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
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.
Example	Show the screen No.20 window. GCSCreateGWindow(20);

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 No.
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 No. (No. 1 -)
Return value	None
Details	Sets the part system No. After the part system No. is set, the control area is registered as a redraw area.
Example	Sets the GNXCounter00000 part system No. in the screen No. 10 to 1. GCSSetSystemNumber(10, "GNXCounter00000", 1) ;

GCSGetSystemNumber	Get part system No.
Syntax	GCSGetSystemNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Part system No.
Details	Gets the part system No.
Example	Gets the GNXCounter00000 part system No. in the screen No. 10 in Stat. LONG Stat; Stat = GCSGetSystemNumber(10, "GNXCounter00000") ;

16.5.3 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 screen No.10 to the momentary mode. GCSButtonSetAction(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetAction(10, "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 screen No.10 to be displayed in a rectangle. GCSButtonSetDisplay(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetDisplay(10, "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 screen No.10 to 1. GCSButtonSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetBorderID(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSButtonGetOnDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSButtonGetOffDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSButtonGetFocusDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSButtonGetDisableDesign(10, "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 screen No.10 to 1. GCSButtonSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat=GCSButtonGetFontID(10,"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 screen No.10 to 1. GCSButtonSetStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSButtonSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetFocusEffect(10, "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 screen No.10 to 1. GCSButtonSetStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSButtonGetStatus(10, "GButton00000");

16.5.4 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 screen No.10 to 1. GCSCheckboxSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetBorderID(10, "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 screen No.10 to 1. GCSCheckboxSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetFontID(10, "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 screen No.10 to 1. GCSCheckboxSetStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSCheckboxSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetFocusEffect(10, "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 screen No.10 to 1. GCSCheckboxSetStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetStatus(10, "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 screen No.10 to 10. GCSCheckboxSetBoxSize(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetBoxSize(10, "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 screen No.10 to white (0xffffff). GCSCheckboxSetBoxColor(10,"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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetBoxColor(10, "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 screen No.10 to white (0xffffff). GCSCheckboxSetFocusColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSCheckboxGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSCheckboxSetDisableCaptionColor(10, "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 screen No.10 in Stat. LONG Stat; Stat=GCSCheckboxGetDisableCaptionColor(10, "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 screen No.10 to white (0xffffff). GCSCheckboxSetDisableBoxColor(10, "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 screen No.10 in Stat. LONG Stat; Stat=GCSCheckboxGetDisableBoxColor(10, "GCheckBox00000");

16.5.5 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 screen No.10 to 1. GCSEditSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetBorderID(10, "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 screen No.10 to 1. GCSEditSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetFontID(10, "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 screen No.10. LONG Stat; Stat = GCSEditAddString(10, "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 screen No.10 to 0xffffff. GCSEditSetTextColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetTextColor(10, "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 screen No.10 to 0xffffff. GCSEditSetBackColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetBackColor(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	mem = GMEMCreate("TESTMEM", 12);
	GCSEditGetBrush(10, "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 screen No.10. GCSEditDeleteString(10, "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 screen No.10. GCSEditAddLine(10, "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 screen No.10. GCSEditInsertLine(10, "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 screen No.10. GCSEditDeleteLine(10, "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(10, "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 screen No.10 to CR+LF. STRING strStat; GCSEditSetLineFeedCode(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetLineFeedCode(10, "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 screen No.10 to line No.2 and index No.3. GCSEditSetCursor(10, "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 screen No.10 as follows:
	nLine : Line No.
	nIndex : Index No.
	LONG nLine;
	LONG nIndex;
	mem = GMEMCreate("TESTMEM", 8);
	GCSEditGetCursor(10, "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 screen No.10 to 0xffffff. GCSEditSetForeColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetForeColor(10, "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 screen No.10 to 2. GCSEditSetFillPattern(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetFillPattern(10, "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 screen No.10 to "overwrite" mode. GCSEditSetInsertMode(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetInsertMode(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	mem = GMEMCreate("TESTMEM", 12);
	GCSEditGetLineBrush(10, "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 screen No.10 to 0xffffff. GCSEditSetLineTextColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetLineTextColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetLength(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 under the following conditions: character string for search : "ABC" replacement character string : "12345" Gets the number of replacements in Stat. LONG Stat; Stat =GCSEditReplaceAll(10, "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 screen No.10 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(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSEditGetLineCount(10, "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 screen No.10, then gets the result in Stat. LONG Stat; Stat = GCSEditRemoveString(10, "GEdit00000", 4);

16.5.6 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 screen No.10 to 1. GCSHtmlbrowserSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetBorderID(10, "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 screen No.10 to 1. GCSHtmlbrowserSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetFontID(10, "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 screen No.10 to 20. GCSHtmlbrowserSetScrollBarWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetScrollBarWidth(10, "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 screen No.10 to white (0xffffff). GCSHtmlbrowserSetTextColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetTextColor(10, "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 screen No.10 to white (0xffffff). GCSHtmlbrowserSetBackColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetBackColor(10, "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 screen No.10 to white (0xffffff). GCSHtmlbrowserSetLinkColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSHtmlbrowserGetLinkColor(10, "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 screen No.10 in "index.htm". GCSHtmlbrowserSetHtmlFileName(10, "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 screen No.10 in strStat. STRING strStat; strStat = GCSHtmlbrowserGetHtmlFileName(10, "GHtmlBrowser00000");

16.5.7 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 screen No.10 to 1. GCSLabelSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSLabelGetFontID(10, "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 screen No.10 to 1. GCSLabelSetStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSLabelGetStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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);

16.5.8 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 screen No.10 to 1. GCSListSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetBorderID(10, "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 screen No.10 to 1. GCSListSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetFontID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to white (0xffffff). GCSListSetFocusColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSListSetScrollButtonColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetScrollButtonColor(10, "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 screen No.10 to white (0xffffff). GCSListSetScrollBarColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetScrollBarColor(10, "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 screen No.10 to white (0xffffff). GCSListSetNormalColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetNormalColor(10, "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 screen No.10 to white (0xffffff). GCSListSetDisableColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetDisableColor(10, "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 screen No.10 to 20. GCSListSetScrollBarWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetScrollBarWidth(10, "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 screen No.10 to white (0xffffff) GCSListSetSelectBarColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetSelectBarColor(10, "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 screen No.10 to 20. GCSListSetMaxListLines(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetMaxListLines(10, "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 screen No.10, and gets the result in Stat. LONG Stat; GCSListAddString(10, "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 screen No.10, and gets the result in Stat. LONG Stat; Stat = GCSListInsertString(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetLineCount(10, "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 screen No.10, and gets the result in Stat. LONG Stat; Stat = GCSListRemoveString(10, "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 screen No.10 in strStat. STRING strStat; GCSListGetListString(10, "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 screen No.10. GCSListRemoveAllStrings(10, "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(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetCurrentSelect(10, "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 screen No.10. GCSListSetVisibleScrollBarStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetVisibleScrollBarStatus(10, "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 screen No.10. GCSListSetEnableScrollBarStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetEnableScrollBarStatus(10, "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 screen No.10. GCSListSetTopLine(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSListGetTopLine(10, "GList00000");

16.5.9 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 screen No.10 to 0. GCSPictureSetDisplay(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSPictureGetDisplay(10, "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 screen No.10 to 1. GCSPictureSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSPictureGetBorderID(10, "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 screen No.10 to 1. GCSPictureSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSPictureGetFontID(10, "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 screen No.10 to 1. GCSPictureSetStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSPictureGetStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 3. GCSPictureSetStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSPictureGetStatus(10, "GPicture00000");

16.5.10 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 screen No.10 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(10,
"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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSProgressbarGetBackGroundDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows: nPat : Fill pattern nFCol : Foreground color nBCol : Background color nImg : Image resource ID

```
mem = GMEMCreate("TESTMEM", 16);
GCSProgressbarGetBarDesign(10, "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 screen No.10. GCSProgressbarSetRange(10, "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 screen No.10 as follows:
	nMin : Minimum value
	nMax : Maximum value
	SHORT nMin;
	SHORT nMax;
	mem = GMEMCreate("TESTMEM", 4);
	GCSProgressbarGetRange(10, "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 screen No.10 to 0. GCSProgressbarSetDirection(10, "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 screen No.10 in Stat. Stat = GCSProgressbarGetDirection(10, "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 screen No.10. GCSProgressbarSetValue(10, "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 screen No.10 in Stat. GCSProgressbarGetValue(10, "GProgressBar00000", 10);

16.5.11 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 screen No.10 to 1. GCSRadiobuttonSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetBorderID(10, "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 screen No.10 to 1. GCSRadiobuttonSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetFontID(10, "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 screen No.10 to 1. GCSRadiobuttonSetStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSRadiobuttonSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetFocusEffect(10, "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 screen No.10 to 1. GCSRadiobuttonSetStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetStatus(10, "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 screen No.10 to 10. GCSRadiobuttonSetBoxSize(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetBoxSize(10, "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 screen No.10 to white (0xffffff). GCSRadiobuttonSetBoxColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetBoxColor(10, "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 screen No.10 to white (0xffffff). GCSRadiobuttonSetFocusColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSRadiobuttonSetDisableCaptionColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetDisableCaptionColor(10, "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 screen No.10 to white (0xffffff). GCSRadiobuttonSetDisableBoxColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetDisableBoxColor(10, "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 screen No.10 to 1. GCSRadiobuttonSetRadioGroup(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSRadiobuttonGetRadioGroup(10, "GRadioButton00000");

16.5.12 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 screen No.10 to 1. GCSScrollbarexSetDisplay(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetDisplay(10, "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 screen No.10 to 1. GCSScrollbarexSetDirection(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetDirection(10, "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 screen No.10 to 10. GCSScrollbarexSetPageSize(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetPageSize(10, "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 screen No.10 to white (0xffffff). GCSScrollbarexSetScrollArrowColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetScrollArrowColor(10, "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 screen No.10 when the button is ON to 1. GCSScrollbarexSetPrevBtnOnImgID(10, "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 screen No.10 when the button is OFF to 1. GCSScrollbarexSetPrevBtnOffImgID(10, "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 screen No.10 when the button is ON in Stat. LONG Stat; Stat = GCSScrollbarexGetPrevBtnOnImgID(10, "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 screen No.10 when the button is OFF in Stat. LONG Stat; Stat = GCSScrollbarexGetPrevBtnOffImgID(10, "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 screen No.10 when the button is ON to 1. GCSScrollbarexSetNextBtnOnImgID(10, "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 screen No.10 when the button is OFF to 1. GCSScrollbarexSetNextBtnOffImgID(10, "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 screen No.10 when the button is ON in Stat. LONG Stat; Stat = GCSScrollbarexGetNextBtnOnImgID(10, "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 screen No.10 when the button is OFF in Stat. LONG Stat; Stat = GCSScrollbarexGetNextBtnOffImgID(10, "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 screen No.10 to white (0xffffff). GCSScrollbarexSetBackGroundColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetBackGroundColor(10, "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 screen No.10 to 1. GCSScrollbarexSetOutLineBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetOutLineBorderID(10, "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 screen No.10 to 5. GCSScrollbarexSetPinchWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetPinchWidth(10, "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 screen No.10 to white (0xffffff). GCSScrollbarexSetPinchColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetPinchColor(10, "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 screen No.10 to 1. GCSScrollbarexSetBarEnableStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSScrollbarexGetBarEnableStatus(10, "GScrollBarEx00000") ;

16.5.13 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 screen No.10 to 1. GCSTextboxSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetBorderID(10, "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 screen No.10 to 1. GCSTextboxSetFontID(10, "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 screen No.10 in Stat. Stat = GCSTextboxGetFontID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSTextboxSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetFocusEffect(10, "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 screen No.10 to white (0xffffff). GCSTextboxSetFocusColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSTextboxSetNormalColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetNormalColor(10, "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 screen No.10 to white (0xffffff). GCSTextboxSetDisableColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetDisableColor(10, "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
Return value	None
Details	Sets the character string type. One of the following values is set for ucType: 0: character string 1: 16bit integer value 3: 32biti integer 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 screen No.10 to 1. GCSTextboxSetTextType(10, "GTextBox00000", 1) ;

(Note) When the character string type is changed, change the character string display format resource ID to the suitable one.

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	0: character string 1: 16bit integer value 3: 32bit integer value
Details	Gets the display character string type.
Example	Gets the GTextBox00000 display character string type in the screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetTextType(10, "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 ID
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 screen No.10 to 1. GCSTextboxSetFormatID(10, "GTextBox00000", 1) ;

(Note) Set the suitable character string display format resource ID for the character string type.

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 in the screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetFormatID(10, "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 screen No.10 to 1. GCSTextboxSetPasswordStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetPasswordStatus(10, "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 screen No.10 to 1. GCSTextboxSetCommaStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetCommaStatus(10, "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 screen No.10 to 1. GCSTextboxSetRefuseInputNumberStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputNumberStatus(10, "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 screen No.10 to 1. GCSTextboxSetRefuseInputSmallLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputSmallLetterStatus(10, "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 screen No.10 to 1. GCSTextboxSetRefuseInputCapitalLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputCapitalLetterStatus(10, "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 screen No.10 to 1. GCSTextboxSetRefuseInputSymbolLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputSymbolLetterStatus(10, "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 screen No.10 to 1. GCSTextboxSetRefuseInputFullLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetRefuseInputFullLetterStatus(10, "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 screen No.10 to 1. GCSTextboxSetCheckMaxStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetCheckMaxStatus(10, "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 screen No.10 to 1. GCSTextboxSetCheckMinStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetCheckMinStatus(10, "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 screen No.10 to 100. GCSTextboxSetMaxGValue(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetMaxGValue(10, "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 screen No.10 to 1. GCSTextboxSetMinGValue(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetMinGValue(10, "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 screen No.10. GCSTextboxSetString(10, "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 : global memory to store 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 screen No.10 in Stat. STRING strStat; GCSTextboxGetString(10, "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 value
2: less than minimum value
3: greater than maximum value

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.

Example

Sets the LONG integer value 100 as a value to be displayed in the GTextBox00000 in the screen No.10.

```
GCSTextboxSetTextType(10, "GTextBox00000", 3) ;
mem = GMEMCreate("TESTMEM", 4) ;
GMEMSetLong(mem, 0, 100) ;
GCSTextboxSetGValue(10, "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 2: less than minimum value 3: greater than maximum value
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.
Example	Gets the numerical value being displayed in GtextBox00000 in the screen No.10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSTextboxGetGValue(10, "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 screen No.10 to -2. GCSTextboxSetBackGroundPattern(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSTextboxGetBackGroundPattern(10, "GTextBox00000") ;

16.5.14 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 screen No.10 to the momentary mode. GCSNCPLCButtonSetAction(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetAction (10, "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 screen No.10 to be displayed in a rectangle. GCSNCPLCButtonSetDisplay(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetDisplay(10, "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 screen No.10 to 1. GCSNCPLCButtonSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetBorderID(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOffDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnOnDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetOnOffDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetFocusDesign(10, "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 screen No.10 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(10, "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 screen No.10 as follows:
	nPat : Fill pattern
	nFCol : Foreground color
	nBCol : Background color
	nImg : Image resource ID
	mem = GMEMCreate("TESTMEM", 16);
	GCSNCPLCButtonGetDisableDesign(10, "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 screen No.10 to 1. GCSNCPLCButtonSetFontID(10, "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 screen No.10 in Stat. LONG Stat; Stat=GCSNCPLCButtonGetFontID(10,"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 screen No.10 to 1. GCSNCPLCButtonSetOnStringID(10, " 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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnStringID(10, " 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 screen No.10 to 1. GCSNCPLCButtonSetOffStringID(10, " 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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOffStringID(10, " 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 screen No.10 to 1. GCSNCPLCButtonSetOnOnStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnOnStringID(10, "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 screen No.10 to 1. GCSNCPLCButtonSetOnOffStringID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetOnOffStringID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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);

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 screen No.10 to 1. GCSNCPLCButtonSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetFocusEffect(10, "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 screen No.10 to 1. GCSNCPLCButtonSetStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCButtonGetStatus(10, "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.

Example

Sets PLC device 1 operation to the reading in GNCPLCButton00000 in the screen No.10.

```
LONG Stat;
Stat = GCSNCPLCButtonSetDeviceType(10,
"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.

Example

Gets PLC device 1 operation in GNCPLCButton00000 in the screen No.10 as nVal : LONG integer value.

```
GMEM mem;
LONG nVal;
mem = GMEMCreate("TESTMEM", 4) ;
GCSNCPLCButtonGetDeviceType(10, "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.
Example	Sets "X0" to PLC device of PLC device 1 in GNCPLCButton00000 in the screen No.10. LONG Stat; Stat = GCSNCPLCButtonSetDevice(10, "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 : global memory to store PLC device character string
Return value	Getting result 0: Abnormal 1: Normal
Details	Gets the PLC device.
Example	Sets PLC device of PLC device 1 to strStat in GNCPLCButton00000 in the screen No.10. STRING strStat; GCSNCPLCButtonGetDevice(10, "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.
Example	Sets 1 to PLC device value of PLC device 1 in GNCPLCButton00000 in the screen No.10. LONG Stat; Stat = GCSNCPLCButtonSetDeviceVal(10, "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.
Example	Gets PLC device value of PLC device 1 in GNCPLCButton00000 in the screen No.10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSNCPLCButtonGetDeviceVal(10, " GNCPLCButton00000", 0, mem); nVal = GMEMGetLong(mem, 0) ; GMEMDelete(mem);

16.5.15 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 screen No.10 to 1. GCSNCPLCTextboxSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetBorderID(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetFontID(10, "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 screen No.10 in Stat. Stat = GCSNCPLCTextboxGetFontID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetFocusEffect(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetFocusEffect(10, "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 screen No.10 to white (0xffffff). GCSNCPLCTextboxSetFocusColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSNCPLCTextboxSetNormalColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetNormalColor(10, "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 screen No.10 to white (0xffffff). GCSNCPLCTextboxSetDisableColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetDisableColor(10, "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 : Signed decimal integer GPLCTEXT_TYPE_UDEC : Unsigned decimal integer GPLCTEXT_TYPE_HEX : Hexadecimal integer GPLCTEXT_TYPE_FLOAT : Real number 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 screen No.10 to 1. GCSNCPLCTextboxSetTextType(10, "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 : Signed decimal integer GPLCTEXT_TYPE_UDEC : Unsigned decimal integer GPLCTEXT_TYPE_HEX : Hexadecimal integer GPLCTEXT_TYPE_FLOAT : Real number
Details	Gets the display character string type.
Example	Gets the GNCPLCTextBox00000 display character string type in the screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetTextType(10, "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. 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 screen No.10 to 2 byte. GCSNCPLCTextboxSetTextSize(10, "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.
Example	Gets the GNCPLCTextBox00000 PLC device size in the screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetTextSize(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetPasswordStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetPasswordStatus(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetCommaStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCommaStatus(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetCheckMaxStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCheckMaxStatus(10, "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 screen No.10 to 1. GCSNCPLCTextboxSetCheckMinStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetCheckMinStatus(10, "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 screen No.10 to 100. GCSNCPLCTextboxSetMaxGValue(10, "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 screen No.10 in nMax.
	LONG nMax;
	mem = GMEMCreate("TESETMEM", 4);
	GCSNCPLCTextboxGetMaxGValue(10,
	"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 screen No.10 to 1. GCSNCPLCTextboxSetMinGValue(10, "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 screen No.10 in nMix.
	LONG nMix;
	mem = GMEMCreate("TESETMEM", 4);
	GCSNCPLCTextboxGetMinGValue(10, "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 failed 1: setting succeeded
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.
Example	Sets the character string "ABCD" in GNCPLCTextBox00000 in the screen No.10. GCSNCPLCTextboxSetString(10, "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 : global memory to store display character string
Return value	0: acquisition failed 1: acquisition succeeded
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 screen No.10 in Stat. STRING strStat; GCSNCPLCTextboxGetString(10, "GNCPLCTextBox00000", strStat) ;

GCSNCPLCTextboxSetGValue	Set display 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 2: less than minimum value 3: greater than maximum value
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 value displayed in a text box is set, the control area is registered as a redraw area.
Example	Sets the LONG integer value 100 as a value to be displayed in the GNCPLCTextBox00000 in the screen No.10. GCSNCPLCTextboxSetGValue(10, "GNCPLCTextBox00000", 3) ; mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSNCPLCTextboxSetGValue(10, "GNCPLCTextBox00000", mem) ; GMEMDelete(mem);

GCSNCPLCTextboxGetGValue	Get display 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 2: less than minimum value 3: greater than maximum value
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.
Example	Gets the numerical value being displayed in GNCPLCTextBox00000 in the screen No.10 as nVal : LONG integer value. 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 screen No.10 to -2. GCSNCPLCTextboxSetBackGroundPattern(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetBackGroundPattern(10, "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 screen No.10 to 2. GCSNCPLCTextboxSetIntegerWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetIntegerWidth(10, "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 screen No.10 to 2. GCSNCPLCTextboxSetDecimalWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetDecimalWidth(10, "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 screen No.10 as DOUBLE integer value : 2.5. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetDouble(mem, 0, 2.5) ; GCSNCPLCTextboxSetScale(10, " 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 screen No.10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSNCPLCTextboxGetScale(10, "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 screen No.10 to 100. GCSNCPLCTextboxSetOffset(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetOffset(10, "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.
Example	Sets "X0" to GNCPLCTextBox00000 PLC device in the screen No.10. LONG Stat; Stat = GCSNCPLCTextboxSetDevice(10, "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 : global memory to store PLC device character string
Return value	Getting result 0: Abnormal 1: Normal
Details	Gets the PLC device.
Example	Gets the GNCPLCTextBox00000 PLC device in the screen No.10 in strStat. STRING strStat; GCSNCPLCTextboxGetDevice(10, "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 screen No.10 to ON. GCSNCPLCTextboxSetZeroSuppressStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCPLCTextboxGetZeroSuppressStatus(10, "GNCPLCTextBox00000");

16.5.16 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 screen No.10 to 1. GCSNCDataTextboxSetBorderID(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetBorderID(10, "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 screen No. 10 to 1. GCSNCDataTextboxSetFontID(10, "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 screen No. 10 in Stat. Stat = GCSNCDataTextboxGetFontID(10, "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 screen No.10 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(10, "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 screen No.10 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(10, "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 screen No. 10 to 1. GCSNCDataTextboxSetFocusEffect(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetFocusEffect(10, "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 screen No. 10 to white (0xffffff). GCSNCDataTextboxSetFocusColor(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetFocusColor(10, "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 screen No.10 to white (0xffffff). GCSNCDataTextboxSetNormalColor(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetNormalColor(10, "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 screen No.10 to white (0xffffff). GCSNCDataTextboxSetDisableColor(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDisableColor(10, "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 screen No.10 to 1. GCSNCDataTextboxSetTextType(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetTextType(10, "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 screen No.10 to 1. GCSNCDataTextboxSetCommaStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetCommaStatus(10, "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 screen No.10 to 1. GCSNCDataTextboxSetRefuseInputNumberStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputNumberStatus(10, "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 screen No.10 to 1. GCSNCDataTextboxSetRefuseInputSmallLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputSmallLetterStatus(10, "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 screen No.10 to 1. GCSNCDataTextboxSetRefuseInputCapitalLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputCapitalLetterStatus(10, "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 screen No.10 to 1. GCSNCDataTextboxSetRefuseInputSymbolLetterStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetRefuseInputSymbolLetterStatus(10, "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 screen No.10 to 1. GCSNCDataTextboxSetZeroSuppressStatus(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetZeroSuppressStatus(10, "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 screen No. 10. GCSNCDataTextboxSetString(10, "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 : global memory to store 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 screen No. 10 in strStat. STRING strStat; GCSNCDataTextboxGetString(10, "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 screen No.10. GCSNCDataTextboxSetStringBuffer(10, "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 screen No.10 to strString. mem = GMEMCreate("TESTMEM", 4) ; STRING strString ; strString = GCSNCDataTextboxGetStringBuffer(10, "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 screen No. 10. GCSNCDataTextboxSetTextType(10, "GNCDataTextBox00000", 2) ; mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSNCDataTextboxSetGNCValue(10, "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 screen No. 10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSNCDataTextboxGetGNCValue(10, "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(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetBackGroundPattern(10, "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 screen No.10 to 2. GCSNCDataTextboxSetIntegerWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetIntegerWidth(10, "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 screen No.10 to 2. GCSNCDataTextboxSetDecimalWidth(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDecimalWidth(10, "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 screen No.10 as DOUBLE real number value : 2.54. mem = GMEMCreate("TESTMEM", 8) ; GMEMSetDouble(mem, 0, 2.54) ; GCSNCDataTextboxSetScale(10, " 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 screen No.10 as nVal : DOUBLE real number value. GMEM mem; DOUBLE nVal; mem = GMEMCreate("TESTMEM", 8) ; GCSNCDataTextboxGetScale(10, "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 screen No.10 to 100. GCSNCDataTextboxSetOffset(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetOffset(10, "GNCDataTextBox00000");

GCSNCDataTextboxSetSystemNumber	Set part system No.
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 No.
Return value	None
Details	Sets the part system No. After the part system No. is set, the control area is registered as a redraw area.
Example	Sets the GNCDataTextBox00000 part system No. in the screen No. 10 to 1. GCSNCDataTextboxSetSystemNumber(10, "GNCDataTextBox00000", 1) ;

GCSNCDataTextboxGetSystemNumber	Get part system No.
Syntax	GCSNCDataTextboxGetSystemNumber(nWindowNo, strName);
Argument	(i)LONG nWindowNo : screen No. (Specify -1 for self screen.) (i)STRING strName : control name
Return value	Part system No.
Details	Gets the part system No.
Example	Gets the GNCDataTextBox00000 part system No. in the screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSystemNumber(10, "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 screen No.10 to 0 (foreground). GCSNCDataTextboxSetGround(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetGround(10, "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 16 : Axis No.
Example	Sets the GNCDataTextBox00000 axis No. in the screen No. 10 to 1. GCSNCDataTextboxSetAxisInfo(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetAxisInfo(10, "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 screen No. 10 to 1. GCSNCDataTextboxSetSection(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSection(10, "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 screen No. 10 to 1. GCSNCDataTextboxSetSubSection(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetSubSection(10, "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(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSNCDataTextboxGetDataType(10, "GNCDataTextBox00000") ;

16.5.17 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 screen No. 10. GCSMenuRefresh(10, "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 screen No. 10 to enabled and redraws the menu. GCSMenuSetRefreshValidFlag(10, "GNXMenu00000", 1) ; GCSMenuRefresh(10, "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.
Example	Sets the names as shown below for the GNXMenu00000 menu buttons in the screen No. 10. 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(10, "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.
Example	Sets the names as shown below for the GNXMenu00000 upper row menu buttons in the screen No. 10. 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(10, "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.
Example	Sets the names as shown below for the GNXMenu00000 lower row menu buttons in the screen No. 10. 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(10, "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.
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 screen No. 10

```
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(10, "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.
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 screen No. 10

```
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(10, "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.

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 screen No. 10

```
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(10, "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 screen No.10. GCSMenuSetMenuButtonState_one(10, "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 screen No. 10. GCSMenuSetMenuButtonUpperState_one(10, "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 screen No. 10. GCSMenuSetMenuButtonLowerState_one(10, "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 screen No.10. LONG Stat; Stat = GCSMenuSetMenuButtonState_one(10, "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 screen No.10. LONG Stat; Stat = GCSMenuGetMenuButtonUpperState_one(10, "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 screen No.10. LONG Stat; Stat = GCSMenuGetMenuButtonLowerState_one(10, "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 screen No.10. LONG Stat; Stat = GCSMenuSetMenuButtonOffIcon_one(10, "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 screen No.10. LONG Stat; Stat = GCSMenuSetMenuButtonOnIcon_one(10, "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) ;

16.5.18 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
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 screen No.10. LONG Stat; Stat = GCSIOCheckUpDateFile(10, "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
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 (M700) 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 and an external device.
Example	Transfers the common variable file "COMMON.VAR" in the "COMMON" directory of "Drive D" using GNXFileInOut00000 in the screen No.10. LONG Stat; Stat = GCSIOFileTransfer(10, "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 (M700) 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 screen No. 10. LONG Stat; Stat = GCSIODeleteFile(10, "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 (M700) 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 screen No. 10. LONG Stat; Stat = GCSIOCreateDirectory(10, "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 (M700) 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 screen No. 10. LONG Stat; Stat = GCSIOCreateDirectory(10, "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 screen No.10. LONG Stat; Stat = GCSInterrupt(10, "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 screen No.10. GCSIOInitializeMessage(10, "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 (M700) 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 screen No.10. LONG Stat; Stat = GCSGetLastError(10, "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 screen No.10. LONG Stat; Stat = GCSIOGetTransferStatus(10, "GNXFileInOut00000");

16.5.19 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 screen No. 10 so that if a message is split, the residual message is not displayed at the next drawing update. GCSAlarmMessageSetDispMessageType(10," GNXAlarmMessage00000", 0);

16.5.20 MonitorStatus

GCSMonStatusSetDispSysNumber	Set part system No. 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 No. to display (0 to 4)
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 screen No. 10 to the part system No. 2. LONG Stat; Stat = GCSMonStatusSetDispSysNumber(10, "GNXMonitorStatus00000", 2);

GCSMonStatusGetDispSysNumber	Get part system No. 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 No. of part system being displayed (0 to 4)
Details	Gets the value of the property "Part system designation".
Example	Gets the part system displayed with GNXMonitorStatus00000 in the screen No. 10 in MonStat. LONG MonStat; MonStat = GCSMonStatusGetDispSysNumber(10, "GNXMonitorStatus00000");

16.5.21 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 screen No. 10. LONG lStatus; lStatus = GCSCounterAxisChange(10, "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 15: Tip workpiece coordinate position 17: Tool axis movement 18: Tip machining position 19: Relative position After the counter type is set, the control area is registered as a redraw area.
Example	Sets the GNXCounter00000 counter type in the screen No. 10 to 1. LONG lStatus; lStatus = GCSCounterSetCounterType(10, "GNXCounter00000", 1) ;

16.5.22 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 After the time type is set, the control area is registered as a redraw area.
Example	Sets the GNXCycleTime00000 time type (upper) in the screen No. 10 to 1. LONG lStatus; lStatus = GCSCycleTimeSetTimeType(10, "GNXCycleTime00000", 0, 1) ;

16.5.23 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 screen No. 10 to 1. GCSLoadMeterSetDispType(10, "GNXLoadMeter00000", 1) ;

16.5.24 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 screen No. 10 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(10, "GNXSPCommand00000", mem, 2) ; GMEMDelete(mem);

16.5.25 NCTable

GCSNCTableSetCellForeColor	Set foreground color of each cell

Syntax	GCSNCTableSetCellForeColor(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 foreground color (RGB) (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
Details	Sets the foreground color of the designated cell. After the foreground color is set, the control area is registered as a redraw area.
Example	Sets blue (0x0000ff) to the foreground color of the cell in GNCTable00000 line No.3, row No.2 in the screen No. 10. LONG Stat; Stat = GCSNCTableSetCellForeColor(10, "GNCTable00000", H0000FF, 3, 2);

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 screen No. 10. LONG Stat; Stat = GCSNCTableSetCellBackColor(10, "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 screen No. 10. LONG Stat; Stat = GCSNCTableSetCellString(10, "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 screen No. 10 to "pString". STRING pString; LONG Stat; Stat = GCSNCTableGetCellString(10, "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 screen No. 10. LONG Stat; Stat = GCSNCTableSetSubCursorNum(10, "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 screen No.10 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(10, "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 screen No.10 to "30" with Pixel mode. LONG Stat; Stat = GCSNCTableSetCellWidth(10, "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 screen No.10 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(10, "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 screen No.10 as the title. LONG Stat; Stat = GCSNCTableSetLineTitleString(10, "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 screen No.10. STRING pString; LONG Stat; Stat =GCSNCTableGetLineTitleString(10, "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 screen No.10 as a title. LONG Stat; Stat = GCSNCTableSetRowTitleString(10, "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 screen No.10. STRING pString; LONG Stat; Stat =GCSNCTableGetRowTitleString(10, "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 screen No.10. LONG Stat; Stat = GCSNCTableSetLineTitleStringPosition(10, "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 screen No.10.

```
LONG Stat;
Stat = GCSNCTableSetRowTitleStringPosition(10,
"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 screen No.10. LONG Stat; Stat = GCSNCTableSetCellStringPosition(10, "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 screen No.10 as the last position before sub cursor moves. LONG Stat; Stat = GCSNCTableSetBeforeSubCursorNum(10, "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 screen No.10 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(10, "GNCTable00000", mem);
	nLine = GMEMGetShort(mem, 0);
	nRow = GMEMGetShort(mem, 2);
	GMEMDelete(mem);

GCSNCTableSetLineTitleForeColor	Set foreground color of line title

Syntax	GCSNCTableSetLineTitleForeColor(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 foreground 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 foreground color of the designated line title. After the color is set, the control area is registered as a redraw area.
Example	Set blue (0x0000ff) to the foreground color of the line title in GNCTable00000 line No.3 in the screen No.10. LONG Stat; Stat = GCSNCTableSetLineTitleForeColor(10, "GNCTable00000", H0000FF, 3);

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 screen No.10. LONG Stat; Stat = GCSNCTableSetLineTitleBackColor(10, "GNCTable00000", H008000, 3);

GCSNCTableSetRowTitleForeColor	Set foreground color of row title

Syntax	GCSNCTableSetRowTitleForeColor(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 foreground 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 foreground color of the designated row title. After the color is set, the control area is registered as a redraw area.
Example	Set blue (0x0000ff) to the foreground color of the row title for GNCTable00000 line No.2 in the screen No.10. LONG Stat; Stat = GCSNCTableSetRowTitleForeColor(10, "GNCTable00000", H0000FF, 2);

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 screen No.10. LONG Stat; Stat = GCSNCTableSetRowTitleBackColor(10, "GNCTable00000", H008000, 2);

GCSNCTableSetSubCursorForeColor	Set foreground color of sub cursor

Syntax	GCSNCTableSetSubCursorForeColor(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(RGB)
Return value	(Long) Setting result 0x0 :Normal
Details	Sets foreground color of the sub cursor. After the color is set, the control area is registered as a redraw area.
Example	Set blue (0x0000ff) to the foreground color of the sub cursor in GNCTable00000 in the screen No.10. LONG Stat; Stat = GCSNCTableSetSubCursorForeColor(10, "GNCTable00000", H0000FF);

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 screen No.10. LONG Stat; Stat = GCSNCTableSetSubCursorBackColor (10, "GNCTable00000", H008000);

16.5.26 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 screen No.10 to 1. GCSInputBoxSetPasswordStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetPasswordStatus(10, "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 screen No.10. GCSInputBoxSetString(10, "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 : global memory to store 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 screen No.10 in strStat. STRING strStat; GCSInputBoxGetString(10, "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 screen No.10. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSInputBoxSetGValue(10, "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 screen No.10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSInputBoxGetGValue(10, "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 screen No. 10 to 1. GCSInputBoxSetFocusEffect(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSInputBoxGetFocusEffect(10, "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 screen No. 10 to -2. GCSInputBoxSetBackGroundPattern(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetBackGroundPattern(10, "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 screen No.10 to "1". GCSInputBoxSetCalcFlag(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetCalcFlag(10, "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 screen No.10 (1). GCSInputBoxSetInputValue(10, "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 screen No.10 to 1. GCSInputBoxSetEchoback(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetEchoback(10, "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 screen No.10 to "1". GCSInputBoxSetAbsIncType(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetAbsIncType(10, "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 screen No.10 in Stat. GCSInputBoxSetRefrectControl(10, "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 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 screen No.10 to Stat. LONG Stat; Stat= GCSInputBoxGetRefrectControlTextType(10," 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 screen No.10 in Stat. LONG Stat; Stat = GCSInputBoxGetInputMode(10, "GInputBox00000") ;
Compliment	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 screen No.10.

```
LONG _lCtrlid;
_lCtrlid = GCSInputBoxGetSubCursorControl
(10,"GInputBox00000") ;
```

Compliment

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
```

16.5.27 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 screen No.10 to "1". GCSSoftKeySetPasswordStatus(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSSoftKeyGetPasswordStatus(10, "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 screen No.10. GCSSoftKeySetString(10, "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 : global memory to store 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 screen No.10 in strStat. STRING strStat; GCSSoftKeyGetString(10, "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 screen No.10. mem = GMEMCreate("TESTMEM", 4) ; GMEMSetLong(mem, 0, 100) ; GCSSoftKeySetGValue(10, "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 screen No.10 as nVal : LONG integer value. GMEM mem; LONG nVal; mem = GMEMCreate("TESTMEM", 4) ; GCSSoftKeyGetGValue(10, "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 screen No. 10 to "1". GCSSoftKeySetFocusEffect(10, "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 screen No. 10 in Stat. LONG Stat; Stat = GCSSoftKeyGetFocusEffect(10, "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 screen No.10 to "1". GCSSoftKeySetCalcFlag(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSSoftKeyGetCalcFlag(10, "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 screen No.10 (1). GCSSoftKeySetInputValue(10, "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 screen No.10 to "1". GCSSoftKeySetAbsIncType(10, "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 screen No.10 in Stat. LONG Stat; Stat = GCSSoftKeyGetAbsIncType(10, "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 screen No.10 in Stat. GCSSoftKeySetRefrectControl(10, "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 screen No.10 to Stat. LONG Stat; Stat= GCSSoftKeyGetRefrectControlTextType(10," 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 screen No.10 in Stat. LONG Stat; Stat = GCSSoftKeyGetInputMode(10, "GSoftKey00000") ;
Compliment	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 screen No.10. GCSSoftKeySetTitleString(10, "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 screen No.10.

```
LONG _lCtrlid;
_lCtrlid = GCSSoftKeyGetSubCursorControl (10,"
GSoftKey00000") ;
```

Compliment

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
```

16.6 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 No.

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 No.

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 No. 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 : global memory to store 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 No. to 3. Stat = GCSNCDataGetString(gControl, 127, 2, strData) ; GMEMDelete(gControl);

16.7 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.
(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 ;.
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.
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;)

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 " ".

17. GWin (Window Control)

This section describes GWin, which is a window control object.

17.1 Outline

NC Designer provides you with a GUI library. Services in the GUI library related to the window system are called GWin. Controls such as buttons and pictures operating on GWin are called GControl, and figures such as rectangles and lines are called GShape.

Class Configuration

GWin Class

Class name	Description
GCBaseWindow	Parent class of all GWin and GCControl classes. This class holds the coordinate position and style, which are common among all classes.
GCPanel	Class for controlling the controls, figures, view frames and other objects.
GCFrame	Class for controlling the window.
GCSVFrame	Class deriving from GCFrame. This class controls the view frame.
GCScreen	Only one class in the system. This class controls the re-drawing area of panels and windows, and distributes events.
GCWindow	Class for controlling the window.
GCWDC	Class for controlling the drawing attributes of the window.

GControl Control Class

Class name	Description
GCControl	This class is the parent of all control objects. It issues the callback process message.
GCBasicControl	Basic control.
GCButton	Button control for momentary and alternate actions. When the button is pressed, the control executes an event through the callback process.
GCPicture	Display control having up to 32 statuses. The control switches the appearance according to the current status.
GCLabel	Display control for displaying the character string.
GCTextBox	Control for entering/displaying character strings and values.
GCList	Control for displaying a list.
GCCheckBox	Control for controlling the check box.
GCRadioButton	Control for controlling the radio button.
GCProgressBar	Control for controlling the progress bar.
GCHtmlBrowser	Control for controlling the HTML browser.
GCScrollBarEx	Control for controlling the scroll bar.
GCEdit	Control for controlling the edit control.
GCNCControl	This class is the parent of NC control. This calls the callback process.

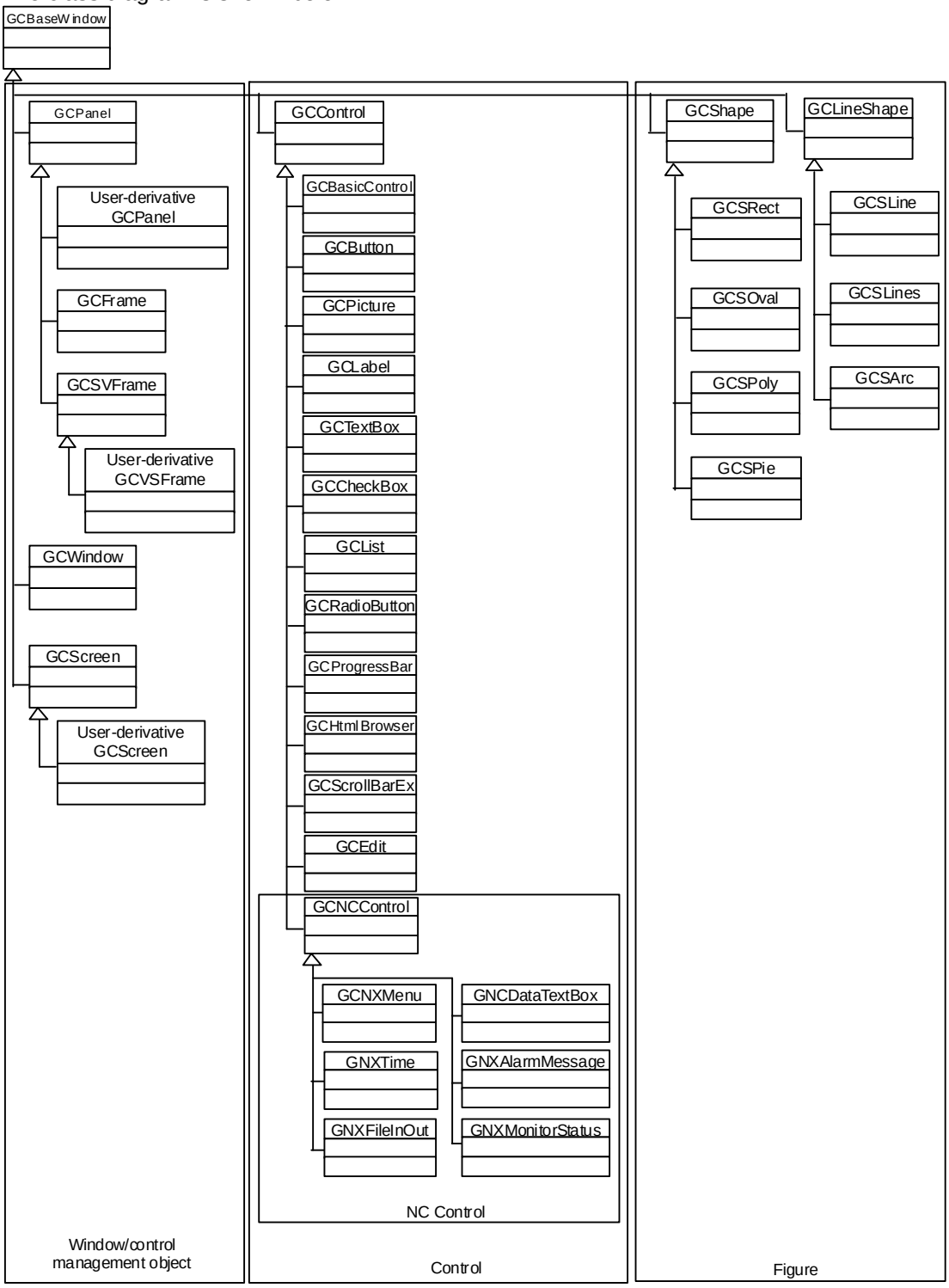
GNCControl NC control class

Class name	Description
GCNCControl	This is an NC control that can be created on a panel. This calls the callback process.
GCNXMenu	Control for displaying the menu button name and icon.
GNCDataTextBox	Control for the internal data of NC to display or enter in the designated rectangle.
GNXMonitorStatus	Control for displaying the NC operation status.
GNXAlarmMessage	Control for displaying the alarm No. and alarm message character string.
GNXFileInOut	Control for inputting/outputting NC data between the NC memory and an external device.
GNXTime	Control for displaying the time.

GShape Figure Class

Class name	Description
GCShape	Parent class of figures. It holds attributes of the perimeter line and those of the brush.
GCSRect	Rectangle figure.
GCSOval	Circle and oval figures.
GCSPoly	Polygon figure.
GCSPie	Sector figure.
GCLineShape	Parent class of line figures. It holds attributes of the line.
GCSLine	Line figure.
GCSLines	Connected line figure.
GCSArc	Arc figure.

The class diagram is shown below.



Class Definition

The classes belonging to GWin, GControl and GShape hold definitions indicating the class type. The range is described below.

Range	Object
0x0000 to 0x00FF	Basic object
0x0100 to 0x5FFF	Window object
0x6000 to 0x7FFF	User window object
0x8000 to 0xBFFF	Control object
0xC000 to 0xCFFF	Figure object
0xD000 to 0xDFFF	Line figure object
0xE000 to 0xFFFF	User control object

The existing class definitions are shown below.

Type	Value	Type	Value
GOBJECT	0x0000	GLINESHAPE	0xD000
GBASEWINDOW	0x0001	GSLINE	0xD001
		GSLINES	0xD002
GPANEL	0x0100	GSARC	0xD003
GSCREEN	0x0101		
GFRAME	0x0102	GNCTABLE	0xEF03
GSVFRAME	0x0103	GNCPLCBUTTON	0xEF06
GWINDOW	0x0104	GNCPLCTEXTBOX	0xEF07
		GNCDATATEXTBOX	0xEF08
GCONTROL	0x8000	GNCPLCEXBUTTON	0xEF09
GBASICCONTROL	0x8001	GNCPLCMESSAGE	0xEF0A
GBUTTON	0x8002		
GPICTURE	0x8003	GNXMENU	0xE002
GLABEL	0x8004	GNXCOUNTER	0xE003
GTEXTBOX	0x8005	GNXGMODAL	0xE004
GLIST	0x8006	GNXGMODALSIMPLE	0xE005
GCHECKBOX	0x8007	GNXCYCLETIME	0xE006
GRADIOBUTTON	0x8008	GNXFEEDRATE	0xE008
GSCROLLBAR	0x8009	GNXMSTB	0xE009
GPROGRESSBAR	0x800A	GNXSPCOMMAND	0xE00A
GHTMLBROWSER	0x800D	GNXLOADMETER	0xE00D
GSCROLLBAREX	0x800E	GNXPRGBUFF	0xE00E
GEDIT	0x800F	GNXONB	0xE00F
		GNXGMODAL_L	0xE032
GSHAPE	0xC000	GNXFILEINOUT	0xE039
GSRECT	0xC001	GNXMONITORSTATUS	0xE03A
GSOVAL	0xC002	GNXTIME	0xE03B
GSPIE	0xC003	GNXALARMMESSAGE	0xE03C
GSPOLY	0xC004		
GSRRECT *	0xC005		

*1 Class including a definition only

Importing Class Attributes

Each class is given an Import structure to acquire the attributes of the object generated during source generation by NC Designer. The name of the structure is the class definition name + property (example: in case of GCBaseWindow class, GBase WindowProperty) in principle.
During importation, the acquisition process proceeds according to the import properties of the corresponding object.

17.2 Macro

Macro Function

Among macros used in the GUI library, those used independently are described below.

Macro function name	Function
GDefPropertyClass()	Property structure declaration part
GDefPropertyEnd()	Property structure termination part
GCClassMP()	Message process declaration part
GCDefMsg()	Message process function definition
GCInheritMP()	Argument to parent class

Macro Definition

Definition of each macro is shown below.

```
/* Property structure attribute macro        */
#define GDefPropertyClass( GClassName, GSuperClass )\
      typedef struct GClassName GClassName;\                  /* Declaration of property structure          */
      struct GClassName {\
      GSuperClass gscBase;\                                   /* Store the parent class.                    */
                                                              /* Describe the structure members in this
                                                              area.                                      */
#define GDefPropertyEnd\
      };

/* Message process macro    */
#define GCClassMP( GClassName )\                              /* The message process function is declared*/
      long GClassName::MessageProcedure( unsigned short usMessage, long lLParam, long lUParam )\
      {\
            switch( usMessage ){\                             /* The message is divided into cases.         */

#define GCDefMsg( message, method )\                          /* Process function is specified for the
                                                              designated message.                        */
            case message:\                                    /* Message is added when necessary.           */
                  return method( usMessage, lLParam, lUParam );\
                  break;\

#define GCInheritMP( GCSuperClass )\
            default:\                                         /* Handed to message process function in
                                                              parent class                               */
            return GCSuperClass::MessageProcedure( usMessage, lLParam, lUParam );\
            break;\
            }\
            return -1;\
      }\
```

For example, to add a message process to panel named GCPanel00000, add the following description in GCPanel00000.cxx.

```
GCClassMP( GCPanel00000 )                                     /* Definition of panel class process function */
            GCDefMsg( GM_MYNEWMSG, NewMsgFnc )                /* Definition of message and process function
                                                                                                         */
GCInheritMP( GCPanel )                                        /* Message is handed to GCPanel class         */
```

17.3 GCBaseWindow (window control)

GCBaseWindow is the basic class of GWin and GCControl. GCBaseWindow holds the definitions of functions common among panels, frames, windows, screens, controls and figures.

Shape

Object shape variable ulStyle uses low-order 16 bits for GBaseWindow and high-order 16 bits for each object.
The data of low-order 16 bits is described below.

Definition	Corresponding bit	Description
GW_STYLE_VISIBLE	Bit 0	Indicates show/hide attribute of the object. 0: Hide 1: Show
GW_STYLE_ENABLE	Bit 1	Indicates whether the object is controlled by an input event or not. 0: Uncontrollable 1: Controllable
GW_STYLE_CONSUMABLE	Bit 2	Indicates whether the input event of the object is consumed or not. 0: Do not consume. 1: Consume.
GW_STYLE_CAPTION	Bit 3	Indicates the show/hide attribute of the object nameplate. 0: Hide 1: Show
GW_STYLE_FOCUS	Bit 4	Indicates whether the focus is located at the object or not. 0: No focus 1: With focus

Importing Structure

The import/export structures of GCBaseWindow are shown below.

```
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;
```

List of Functions

The functions of GCBaseWindow are listed below.

Function name	Function
GCSGetType	Get object type
GCSGetID	Get object ID
GCSGetXPosition	Get object horizontal position
GCSGetYPosition	Get object vertical position
GCSGetPosition	Get object position in point structure
GCSGetWidth	Get object width
GCSGetHeight	Get object height
GCSGetSize	Get object size
GCSSetBounds	Set relative position within parent object
GCSGetBounds	Get relative position within parent object
GCSCalcClientRect	Get coordinate position of client area
GCSGetClientRect	Get object client area
GCSClientToScreen	Convert client coordinates to screen coordinates
GCSGetStyle	Batch style acquisition
GCSSetVisibleStatus	Set display/non-display status
GCSGetVisibleStatus	Get display/non-display status
GCSSetEnableStatus	Set controllable or uncontrollable status
GCSGetEnableStatus	Get controllable or uncontrollable status
GCSSetConsumableStatus	Set input event consumable or not consumable status
GCSGetConsumableStatus	Get input event consumable or not consumable status
GCSSetCaptionStatus	Set caption display status
GCSGetCaptionStatus	Get caption display status
GCSSetFocusStatus	Set focus status
GCSGetFocusStatus	Get focus status
GCSSetStyleStatus	Set style
GCSGetStyleStatus	Get style
GCSSetParent	Set parent object
GCSGetParent	Get parent object
GCSImportProperty	Import property settings
GCSCreate	Create process
GCSDelete	Delete process
GCSPaint	Drawing process
GCSOnDraw	Client area drawing process
GCSAddRefreshRect	Client area redraw area registration
GCSLbuttonPress	Process for pressing left mouse button
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSLbuttonRelease	Process for releasing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSKeyRelease	Process for releasing key
GCSSetFocus	Focus setting process
GCSKillFocus	Focus removing process
GCSTimer	Timer process
GCSUser	User process
GCSSetFocusInformation	Set focus movement information
GCSGetFocusInformation	Get focus movement information
GCSChar	Character input process
GCSClose	Issue object close process request
GCSDeleteChild	Child object deletion process

17.4 GCPanel (panel)

GCPanel is the parent class of the classes of which control objects in GWin. It is provided with a basic mechanism of the class that controls the window or manages the controls.

Importing Structure

The import structure of GCPanel is shown below.

```
GDefPropertyClass( GPanelProperty, GBaseWindowProperty )
    GDesign gdBackGroundDesign;                    /* Background design          */
GDefPropertyEnd
```

Parent Class

The parent class is GCBaseWindow.

List of Functions

The functions of GCPanel are listed below.

Function name	Function
GCSSetBackGroundDesign	Set background design
GCSGetBackGroundDesign	Get background design
GCSImportProperty	Import property settings
GCSCreate	Create process
GCSDelete	Delete process
GCSAddChild	Add child object
GCSDeleteAllChildren	Delete all child objects
GCSGetChild	Search child object
GCSCreateChildren	Process for creating child object
GCSPaint	Drawing process
GCSOnDraw	Client area drawing process
GCSDrawBackGround	Background drawing process
GCSDrawChildren	Child object drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSGetChildList	Get child object list
GCSKeyPress	Process for pressing key
GCSSetFocus	Process for focus setting
GCSDeleteChild	Child object deletion process
GCSSetActiveChild	Set active object
GCSGetActiveChild	Get active object (only direct child object)
GCSEndModal	Process for ending modal window

17.5 GCFrame (frame)

GCFrame is the parent class of the classes that control the window and panel in GCWin. It is provided with a basic mechanism necessary for controlling windows.

Importing Structure

No import member is provided for GCFrame.

Parent Class

The parent class of GCFrame is GCPanel.

List of Functions

The functions of GCFrame are listed below.

Function name	Function
GCSOnLButtonPress	Client area process for pressing left mouse button

17.6 GCSVFrame (view frame)

GCSVFrame considers registered child objects as a single screen and enables to display only one screen among multiple screens it controls.

Importing Structure

No import member is provided for GCSVFrame.

Parent Class

The parent class of GCSVFrame is GCFrame.

List of Functions

The functions of GCSVFrame are listed below.

Function name	Function
GCSCreate	Create process
GCSAddChild	Add child object
GCSShowPanel	Display panel switching process
GCSDrawChildren	Child object drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button

17.7 GCScreen (screen)

GCScreen considers registered child objects as a single screen and enables to display only one screen among multiple screens it controls.

Importing Structure

The import structure of GCScreen is shown below.

```
GDefPropertyClass( GScreenProperty, GBaseWindowProperty )
        GDesign          gdBackGroundDesign;   /* Background              */
        unsigned short   usSystemFontID;       /* System font ID          */
        unsigned short   usSystemBorderID;     /* Solid frame resource    */
        GColor           gcSystemForeColor;    /* System foreground color */
        GColor           gcSystemBackColor;    /* System background color */
        GColor           gcSystemActiveColor;  /* System active color     */
        GColor           gcSystemInactiveColor;/* System inactive color   */
        unsigned short   usSystemPalette;      /* System palette          */
GDefPropertyEnd
```

Parent Class

The parent class of GCScreen is GCBaseWindow.

List of Functions

The functions of GCScreen are listed below.

Function name	Function
GCSCreateInstance	Create instance process
GCSShowPanel	Display panel switching process
GCSCreateGWindow	Create window process
GCSModalGWindow	Create modal window process
GCSImportProperty	Import property settings
GCSCreate	Create process
GCSAttachPanel	Attach panel object
GCSGetPanel	Get attached GCPanel object
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSPaint	Drawing process
GCSOnDraw	Client area drawing process
GCSDrawFrame	Related panel drawing process
GCSDrawPanel	Related GCPanel object drawing process
GCSDrawBackGround	Background drawing process
GCSDispatchMessage	Event dispatch process
GCSAddRefreshRect	Client area redraw area registration
GCSSetBackGroundDesign	Set background design
GCSGetBackGroundDesign	Get background design
GCSSetSystemFontID	Set system font resource ID
GCSGetSystemFontID	Get system font resource ID
GCSSetSystemBorderID	Set system 3D border resource ID
GCSGetSystemBorderID	Get system 3D border resource ID
GCSSetSystemForeColor	Set system foreground color
GCSGetSystemForeColor	Get system foreground color
GCSSetSystemBackColor	Set system background color
GCSGetSystemBackColor	Get system background color
GCSSetSystemActiveColor	Set system active color
GCSGetSystemActiveColor	Get system active color
GCSSetSystemInactiveColor	Set system inactive color
GCSGetSystemInactiveColor	Get system inactive color
GCSChangeFocus	Change focus
GCSKeyPress	Process for pressing key
GCSKeyRelease	Process for releasing key
GCSGetFocus	Get focus status
GCSChar	Character input process
GCSAddWindow	Add window
GCSGetFrame	Get frame
GCSGetWDC	Get drawing context

17.8 GCWindow (window)

GWindow is a class controlling the display of window components, click on the window, redrawing process and so on.

Importing Structure

The import/export structure of GCWindow is shown below.

```
GDefPropertyClass( GWindowProperty, GBaseWindowProperty )
    unsigned short      usStringID;                     /* Title bar character string ID        */
GDefPropertyEnd
```

Parent Class

The parent class of GCWindow is GCBaseWindow.

List of Functions

The functions of GCWindow are listed below.

Function name	Function
GCSSetTitleBarStatus	Set title bar present/none status
GCSGetTitleBarStatus	Get title bar present/none status
GCSSetCloseButtonStatus	Set close button present/none status
GCSGetCloseButtonStatus	Get close button present/none status
GCSSetEdgeStatus	Set window frame present/none status
GCSGetEdgeStatus	Get window frame present/none status
GCSImportProperty	Import property settings
GCSAttachPanel	Attach panel object
GCSGetPanel	Get attached GCPanel object
GCSPaint	Drawing process
GCSOnDraw	Client area drawing process
GCSDrawPanel	Related GCPanel object drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSSetFocus	Focus setting process
GCSDeleteChild	Child object deletion process
GCSDelete	Object deletion process
GCSSetModalStatus	Set modal window status
GCSGetModalStatus	Get modal window status
GCSBeginModal	Start process for modal window
GCSEndModal	End process for modal window
GCSGetRetValue	Get return value upon completion of modal window

17.9 GCWDC (drawing attribute control)

GCWDC is a parent class for controlling the drawing attribute related to the window.

Importing Structure

No import member is provided for GCWDC.

Parent Class

There is no parent class for GCWDC.

List of Functions

The functions of GWindow are listed below.

Function name	Function
GCSWDCGetHGDRAW	Get drawing handle
GCSWDCOffsetOrigin	Move position of origin
GCSWDCBeginDraw	Start drawing process
GCSWDCEndDraw	End drawing process
GCSWDCSetClipRect	Define clipping area
GCSWDCInceptClipRect	Define the overlapped area of the two clipping areas.

18. GControl (control management)

This section describes a control, GControl.

18.1 GCControl (control management)

GCControl is a control that can be created on the panel. It is the parent class of GCButton and GCPicture. It calls callback processes in principle.

Importing Structure

GCControl has no importing member.

Parent Class

The parent class of GCControl is GCBaseWindow.

Function List

The functions of GCControl are listed below.

Function name	Function
GCSCreate	Create process
GCSDelete	Delete process
GCSOnDraw	Client area drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSKeyPress	Process for pressing key
GCSKeyRelease	Process for releasing key
GCSSetFocus	Focus setting process
GCSKillFocus	Focus removing process
GCSTimer	Timer process
GCSUser	User process
GCSCallBackMessage	Call back message execution process

18.2 GCBasicControl (basic control)

GCBasicControl is a control that provides the user with an area where the user can perform original drawing or processes.

Importing Structure

GCBasicControl has no importing member.

Parent Class

The parent class of GCBasicControl is GCControl.

Function List

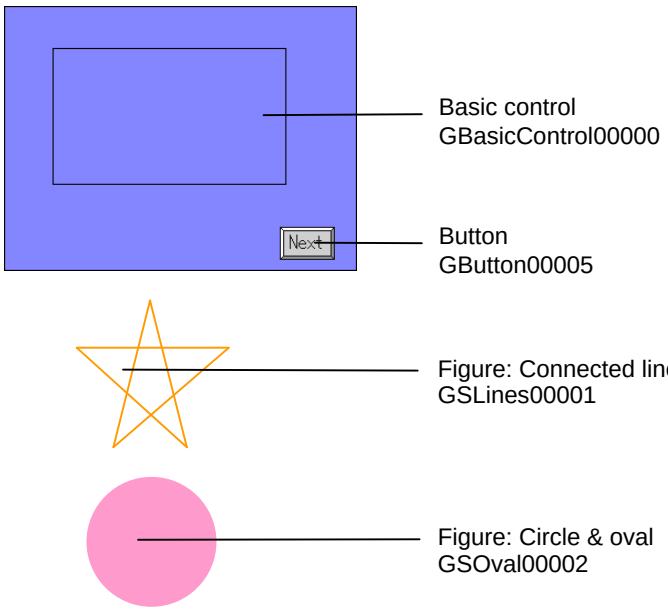
The functions of GCControl are listed below.

Function name	Function
GCSCreate	Create process
GCSDelete	Delete process
GCSOnDraw	Client area drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSKeyPress	Process for pressing key
GCSKeyRelease	Process for releasing key
GCSSetFocus	Focus setting process
GCSKillFocus	Focus removing process
GCSTimer	Timer process
GCSUser	User process

Usage Example

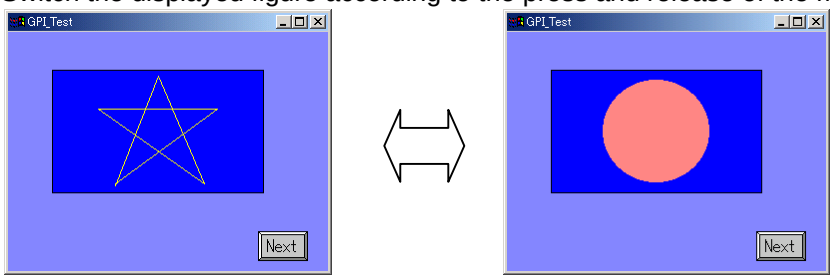
Create the following sample.

Screen Configuration



Action State

Switch the displayed figure according to the press and release of the mouse button.



When the mouse button is pressed

When the mouse button is released

Settings and Action

Create the basic control and figures (connected lines and circle) on the screen. During the procedure, arrange the figure outside the display area and use the program to arrange connected lines and circle over the basic control.
Use the program to specify the following actions in the callback functions of the basic control.

OnCreate()	: Validate the mouse click.
OnDraw()	: Fill the basic control in blue.
OnPress()	: Display connected lines when the left mouse button is pressed.
OnRelease()	: Display a circle when the left mouse button is released.

When the button (character string: "Next") is pressed, switch to the next page.

NOTE

- The default setting of the basic control does not accept mouse clicks. Use OnCreate() to validate the mouse click event.

Source Code

Basic control

```
short GCPanel00000::CreateChildren( )
{
        GBaseObject *pChild;
        GBaseObject *pGBasic;                                   //Basic control object
        GBaseObject *pLines;                                    //Polygon object
        GBaseObject *pOval;                                     //Oval object

        unsigned int i;
        for( i = 1; i < sizeof(pProperty)/sizeof(GBaseWindowProperty*); i++ ){
                pChild = GCSCreateInstance( GCSGetScreen( m_pSelf ),
                pProperty[i]->usType, pProperty[i], GetGBaseObject() );
                if( pChild ){
                        AddChild( pChild );
                        GCSCreate( pChild, GetGBaseObject() );
                }
        }
                                                                //Acquisition of each object
        pGBasic = GetChild( GBASICCONTROL00000 );
        pLines = GetChild( GSLINES00001 );
        pOval = GetChild( GSOVAL00002 );

        MoveShapeCenter( pGBasic, pOval );                      //Arrange the oval in the center
        MoveShapeCenter( pGBasic, pLines );                     //Arrange the star in the center
        GCSSetVisibleStatus( pOval, FALSE );                    //Hide the circle

        return TRUE;
}
```

```
long GCPanel00000::GBASICCONTROL00000OnCreate(unsigned short usMessage, long lLParam,
                                                                    long lUParam)
{
        GBaseObject *pGBasic = GetChild(GBASICCONTROL00000);
        GCSSetEnableStatus( pGBasic, TRUE );                    //Validate the mouse click

        return TRUE;
}

long GCPanel00000::GBASICCONTROL00000OnDraw(unsigned short usMessage, long lLParam,

                                                                    long lUParam)
{
        HGDRAW      hDraw;                                      //Drawing handle
        Grect       gRect;                                      //Fill a rectangle as a basic control
        GBaseObject *pGBasic = GetChild(GBASICCONTROL00000);
        GCSGetBounds( pGBasic, &gRect );                        //Acquire the contour size

        HDraw       = GDOpenDrawHandle(DEFAULT_SCREEN, NULL); //Start drawing
        GDSetFillColor(hDraw, RGB32(255,0,0), RGB32(0,0,255));
        GDDrawRect(hDraw, &gRect);                              //Fill the rectangle in blue
        GDFlushScreen(hDraw, gRect.nXmin, gRect.nYmin,
        short(gRect.nXmax-gRect.nXmin+1), short(gRect.nYmax-gRect.nYmin+1));
        GDCloseDrawHandle(hDraw);                               //Finish drawing

        return TRUE;
}

long GCPanel00000::GBASICCONTROL00000OnPress(unsigned short usMessage, long lLParam,

                                                                    long lUParam)
{
                                                                //Acquire the figure objects
        GBaseObject *pLines     = GetChild(GSLINES00001);
        GBaseObject *pOval      = GetChild(GSOVAL00002);
                                                    //Display connected lines. Hide the circle
        GCSSetVisibleStatus( pLines, FALSE );
        GCSSetVisibleStatus( pOval, TRUE );

        return TRUE;
}
```

```
long GCPanel00000::GBASICCONTROL00000OnRelease(unsigned short usMessage,
                                                        long lLParam, long lUParam)
{
                                                        //Acquire the figure objects
        GBaseObject *pLines     = GetChild(GSLINES00001);
        GBaseObject *pOval      = GetChild(GSOVAL00002);
                                                        //Hide connected lines and display the circle
        GCSSetVisibleStatus( pLines, TRUE );
        GCSSetVisibleStatus( pOval, FALSE );

        return TRUE;
}

//Arrange figures in the center of the basic control.
void MoveShapeCenter(GBaseObject *pParent, GBaseObject *pChild)
{
        GRect gParentRect, gChildRect;
        short nX, nY;
        GCSGetBounds( pChild, &gChildRect );        //Acquire the figure position coordinates
                                                    //Acquire the position coordinates of the
        GCSGetBounds( pParent, &gParentRect );      basic control
        nX = (gParentRect.nXmax-gParentRect.nXmin)/2-(gChildRect.nXmax-gChildRect.nXmin)/2;
        nY = (gParentRect.nYmax-gParentRect.nYmin)/2-(gChildRect.nYmax-gChildRect.nYmin)/2;
                                                    //Re-calculate the figure displaying position
        gChildRect.nXmax = nX+gParentRect.nXmin+(gChildRect.nXmax-gChildRect.nXmin);
        gChildRect.nXmin = nX+gParentRect.nXmin;
        gChildRect.nYmax = nY+gParentRect.nYmin+(gChildRect.nYmax-gChildRect.nYmin);
        gChildRect.nYmin = nY+gParentRect.nYmin;

        GCSSetBounds( pChild, &gChildRect );        //Set in the new coordinate position
}
```

18.3 GCButton (button)

GCButton is a control that switches the ON/OFF state of the button according to the press and release of the mouse (pointing device).

Action

The action of GCButton includes the followings.

Definition	Value	Description
GBTN_ACT_NOACTION	0x00	There is no button action.
GBTN_ACT_MOMENTARY	0x01	Momentary action button. ON state upon the press of the mouse (pointing device), OFF state upon the release.
GBTN_ACT_ALTERNATE	0x02	Alternating action button. The ON and OFF states of the button are switched over each time the mouse button (pointing device) is clicked.

Display

The appearance of GCButton includes the followings.

Definition	Value	Description
GBTN_DISP_RECT	0x00	Regular rectangle display. The button is displayed according to the designated color and filling pattern.
GBTN_DISP_IMAGE	0x01	Image display. The designated image is displayed on the button.
GBTN_DISP_OVAL	0x03	Round display.

State

The state of GCButton includes the followings according to the button type.

Momentary Button

Definition	Value	Description
GBTN_STATE_OFF	0x00	OFF state. The button is not pressed.
GBTN_STATE_ON	0x01	ON state. The button is pressed.

Alternating Button

Definition	Value	Description
GBTN_STATE_OFF	0x00	OFF state.
GBTN_STATE_ON	0x01	ON state.
GBTN_STATE_OFFSELECT	0x02	The button in the OFF state is pressed. When the button is released, transition to the ON state occurs. When other than the button is released, transition to the OFF state occurs.
GBTN_STATE_ONSELECT	0x03	The button in the ON state is pressed. When the button is released, transition to the OFF state occurs.When other than the button is released, transition to the ON state occurs.

Importing Structure

The importing and exporting structures of GCButton are shown below.

```
GDefPropertyClass( GButtonProperty, GControlProperty )
                unsigned short      usStringID;         /* Character string resource ID */
                GCaption            gcCaption;          /* Caption                      */
                unsigned short      usFontID;           /* Font resource ID             */
                unsigned short      usBorderID;         /* Solid frame resource ID      */
                GDesign             gdOnDesign;         /* ON display design            */
                GDesign             gdOffDesign;        /* OFF display design           */
                GDesign             gdFocusDesign;      /* Design of focused button     */
                GDesign             gdDisableDesign;    /* Design of disabled button    */
                unsigned char       ucAction;           /* Action                       */
                unsigned char       ucDisplay;          /* Display                      */
                unsigned char       ucFocusEffect;      /* Focusing effect              */
GDefPropertyEnd
```

Parent Class

The parent class of GCButton is GCControl.

Function List

The functions of GCButton include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

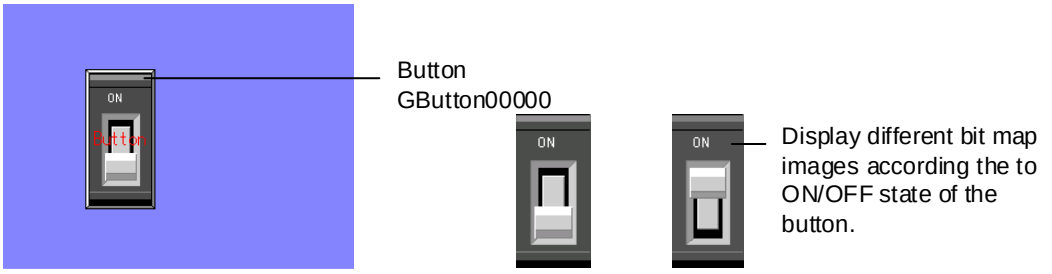
Functions Characteristic to GCButton

Function name	Function
GCSButtonSetAction	Set button operations
GCSButtonGetAction	Get button operations
GCSButtonSetDisplay	Set button display
GCSButtonGetDisplay	Get button display
GCSButtonSetBorderID	Set 3D border resource ID
GCSButtonGetBorderID	Get 3D border resource ID
GCSButtonSetOnDesign	Set ON status design
GCSButtonGetOnDesign	Get ON status design
GCSButtonSetOffDesign	Set OFF status design
GCSButtonGetOffDesign	Get OFF status design
GCSButtonSetFocusDesign	Set FOCUS status design
GCSButtonGetFocusDesign	Get FOCUS status design
GCSButtonSetDisableDesign	Set Disable status design
GCSButtonGetDisableDesign	Get Disable status design
GCSButtonSetFontID	Set font resource ID
GCSButtonGetFontID	Get font resource ID
GCSButtonSetStringID	Set caption character string resource ID
GCSButtonGetStringID	Get caption character string resource ID
GCSButtonSetCaption	Set caption information
GCSButtonGetCaption	Get caption information
GCSButtonSetFocusEffect	Set effect during focus
GCSButtonGetFocusEffect	Get effect during focus
GCSButtonImportProperty	Import property settings
GCSButtonOnDraw	Client area drawing process
GCSButtonOnLButtonPress	Client area process for pressing left mouse button
GCSButtonOnLButtonRelease	Client area process for releasing left mouse button
GCSButtonSetStatus	Set object status
GCSButtonGetStatus	Get object status

Usage Example

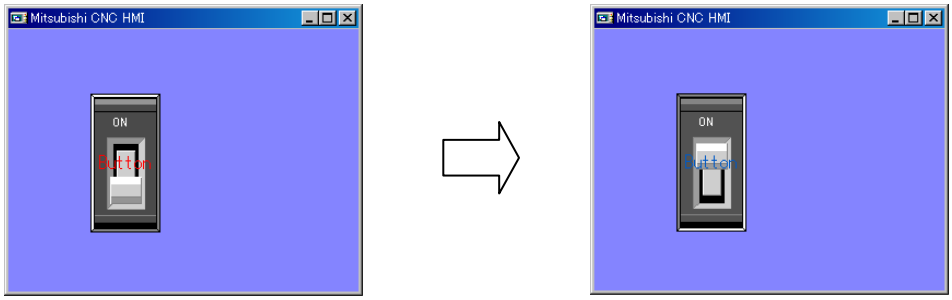
Create the following sample.

Screen Configuration



Action State

Change the ON/OFF state of the button and the character string when the button is clicked.



Settings and Action

Create a button on the screen. Select the image as a button display type, and specify a bit map image of the OFF and ON designs for each. Specify the default momentary button type. Use the program to specify the following actions in callback functions of the control.

OnCreate() : Change to the alternating button type.
OnClick() : Change the color of the displayed character string each time the button is clicked.

NOTE

◆ Change the button type and button character string color in the program.

Source Code

```
long GCPanel00001::GBUTTON00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pButton;				//Button object
	unsigned char cAction;				//Button action

	pButton = GetChild( GBUTTON00000 );		//Acquire the button object
	cAction = GBTN_ACT_ALTERNATE;			//Specify the alternating button action
	GCSButtonSetAction( pButton, cAction );

	return TRUE;
}

long GCPanel00001::GBUTTON00000OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pButton;				//Button object
	GCaption    *gCapt;				//Character string data
	static unsigned char cCol=0;			//Character color
	pButton    = GetChild( GBUTTON00000 );		//Acquire the button object
	gCapt = GCSButtonGetCaption( pButton );
	cCol +=0x10;					//Create the character string color
	gCapt->gcColor = RGB32(cCol%0x100, (cCol+0x40)%0x100, (cCol+0x80)%0x100);
	GCSButtonSetCaption( pButton, gCapt );		//Specify the character string color

	return TRUE;
}
```

18.4 GCPicture (picture)

GCPicture is a control that switches appearance according to the state of an external device or the inside of the software to notify the state to the user.

Display

The appearance of GCPicture includes the followings.

Definition	Value	Description
GPCTR_DISP_RECT	0x00	Rectangle display
GPCTR_DISP_OVAL	0x01	Circle & oval display
GPCTR_DISP_IMAGE	0x02	Image display. Rectangle shape. Displays the designated image on the picture.

State

GCPicture can designate up to 32 display designs (filling patterns and images) according to the state.

Importing Structure

The importing structures of GCPicture are as shown below.

```
GDefPropertyClass( GPictureProperty, GControlProperty )
        unsigned short      usStringID;         /* Character string resource ID  */
        GCaption            gcCaption;          /* Caption                       */
        unsigned short      usFontID;           /* Font resource ID              */
        unsigned short      usBorderID;         /* Solid frame resource ID       */
        unsigned char       ucDisplay;          /* Display                       */
        unsigned char       ucMaxStatus;        /* Max. state number             */
        GDesign             *pStatusDesign;     /* State display design          */
GDefPropertyEnd
```

Parent Class

The parent class of GCPicture is GCControl.

Function List

The functions of GCPicture include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

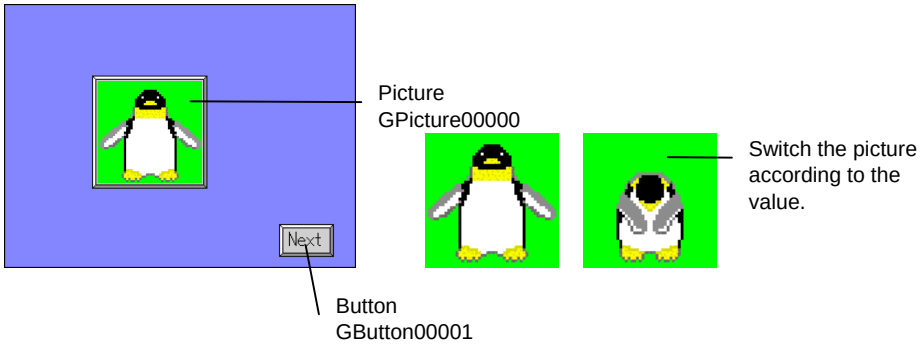
Functions Characteristic to GCPicture

Function name	Function
GCSPictureSetDisplay	Set picture display
GCSPictureGetDisplay	Get picture display
GCSPictureSetMaxStatus	Set maximum picture status
GCSPictureGetMaxStatus	Get maximum picture status
GCSPictureSetStatusDesign	Set picture status design
GCSPictureGetStatusDesign	Get picture status design
GCSPictureSetBorderID	Set 3D border resource ID
GCSPictureGetBorderID	Get 3D border resource ID
GCSPictureSetFontID	Set font resource ID
GCSPictureGetFontID	Get font resource ID
GCSPictureSetStringID	Set caption character string resource ID
GCSPictureGetStringID	Get caption character string resource ID
GCSPictureSetCaption	Set caption information
GCSPictureGetCaption	Get caption information
GCSPictureImportProperty	Import property settings
GCSPictureOnDraw	Client area drawing process
GCSPictureSetStatus	Set object status
GCSPictureGetStatus	Get object status
GCSPictureStartCaptionScroll	Start caption character string scroll operation
GCSPictureStopCaptionScroll	Stop caption character string scroll operation
GCSPicturePauseCaptionScroll	Pause caption character string scroll operation
GCSPictureRestartCaptionScroll	Restart caption character string scroll operation
GCSPictureGetCaptionScrollStatus	Get caption character string scroll status
GCSPictureIsCaptionOutOfBounds	Confirm overflowing of caption character string
GCSPictureSetCaptionScrollDelayTime	Set scroll delay time
GCSPictureGetCaptionScrollDelayTime	Get scroll delay time
GCSPictureSetCaptionScrollRefreshTime	Set scroll refresh time
GCSPictureGetCaptionScrollRefreshTime	Get scroll refresh time
GCSPictureSetCaptionScrollMovementValue	Set scroll movement value
GCSPictureGetCaptionScrollMovementValue	Get scroll movement value
GCSPictureSetCaptionScrollStartPosition	Set scroll start position
GCSPictureGetCaptionScrollStartPosition	Get scroll start position
GCSPictureSetCaptionScrollPosition	Set caption character string scroll position
GCSPictureGetCaptionScrollPosition	Get caption character string scroll position
GCSPictureStartCaptionBlink	Start caption character string blink
GCSPictureStopCaptionBlink	Stop caption character string blink
GCSPictureSetBlinkInterval	Set blink interval
GCSPictureGetBlinkInterval	Get blink interval
GCSPictureSetBlinkType	Set blink type
GCSPictureGetBlinkType	Get blink type

Usage Example

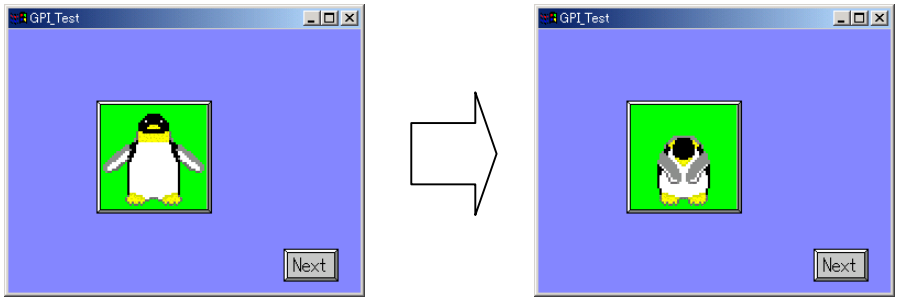
Create the following sample.

Screen Configuration



Action State

Switch the bit map image automatically according to the timer.



Bit map image 1　　Bit map image 2

Settings and Action

Create pictures on the screen and register "2" as a number of states. Paste the above two bit map images to each state. Create a one-second timer event during picture generation to periodically switch the images. Use the program to specify the following actions in callback functions of the control.

OnCreate()　: Create a one-second timer event.
OnTimer()　: Periodically switch the image.

When the button (character string: "Next") is pressed, switch to the next page. At the time, delete the timer created with OnCreate().

NOTE

- When the picture (GPicture00000) is deleted (OnDelete()), the timer is not deleted because the picture has been deleted. The timer is deleted when the page is switched.

Source Code

```
long GCPanel00005::GPICTURE00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pPicture;                                  //Picture object
	pPicture = GetChild( GPICTURE00000 );                   //Acquire the picture object

	return TRUE;
}

long GCPanel00005::GPICTURE00000OnTimer(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pPicture;
	short nValue;
	static int nCounter;                                    //Drawing counter
	pPicture = GetChild(GPICTURE00000);

	nCounter++;
	if(nCounter > 10000) {                                  //Acquire the picture state
		nValue = GCSPictureGetStatus( pPicture );       //Switch the image according to the state
		GCSPictureSetStatus( pPicture, (unsigned char )(nValue ? 0 : 1) );
		nCounter = 0;
	}

	return TRUE;
}
long GCPanel00005::GBUTTON00001OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pPicture;                                  //Picture object
	pPicture = GetChild( GPICTURE00000 );                   //Acquire the picture object

	GESetEvent( GECreateEventMessage( GM_SHOWPANEL,
	GCSGetScreen( GetGBaseObject() ),    6, 0 ) , FALSE );

	return TRUE;
}
```

18.5 GCLabel (label)

GCLabel is a control that displays a character string in the designated rectangle.

Importing Structure

The importing structure of GCLabel includes the followings.

```
GDefPropertyClass( GLabelProperty, GControlProperty )
                unsigned short      usStringID;         /* Character string resource ID   */
                GCaption            gcCaption;          /* Caption                        */
                unsigned            short usFontID;     /* Font resource ID               */
GDefPropertyEnd
```

Parent Class

The parent class of GCLabel is GCControl.

Function List

The functions of GCLabel include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

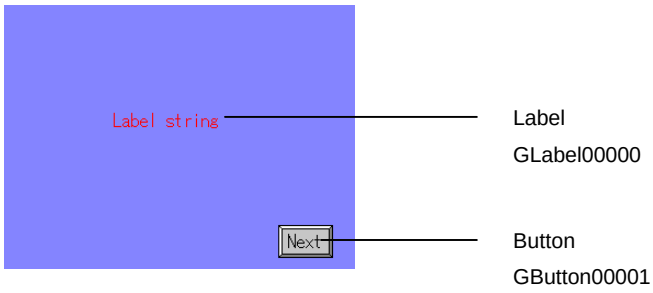
Functions Characteristic to GCLabel

Function name	Function
GCSLabelSetFontID	Set font resource ID
GCSLabelGetFontID	Get font resource ID
GCSLabelSetStringID	Set caption character string resource ID
GCSLabelGetStringID	Get caption character string resource ID
GCSLabelSetCaption	Set caption information
GCSLabelGetCaption	Get caption information
GCSLabelImportProperty	Import property settings
GCSLabelOnDraw	Client area drawing process
GCSLabelStartCaptionScroll	Start scroll operation
GCSLabelStopCaptionScroll	Stop scroll operation
GCSLabelPauseCaptionScroll	Pause scroll operation
GCSLabelRestartCaptionScroll	Restart scroll operation
GCSLabelGetCaptionScrollStatus	Get caption character string scroll status
GCSLabelIsCaptionOutOfBounds	Confirm overflowing of caption character string
GCSLabelSetCaptionScrollDelayTime	Set scroll delay time
GCSLabelGetCaptionScrollDelayTime	Get scroll delay time
GCSLabelSetCaptionScrollRefreshTime	Set scroll refresh time
GCSLabelGetCaptionScrollRefreshTime	Get scroll refresh time
GCSLabelSetCaptionScrollMovementValue	Set scroll movement value
GCSLabelGetCaptionScrollMovementValue	Set scroll movement value
GCSLabelSetCaptionScrollStartPosition	Set scroll start position
GCSLabelGetCaptionScrollStartPosition	Get scroll start position
GCSLabelSetCaptionScrollPosition	Set total scroll movement value
GCSLabelGetCaptionScrollPosition	Get total scroll movement value
GCSLabelStartCaptionBlink	Start caption blink
GCSLabelStopCaptionBlink	Stop caption blink
GCSLabelSetBlinkInterval	Set blink interval
GCSLabelGetBlinkInterval	Get blink interval
GCSLabelSetBlinkType	Set blink type
GCSLabelGetBlinkType	Get blink type

Usage Example

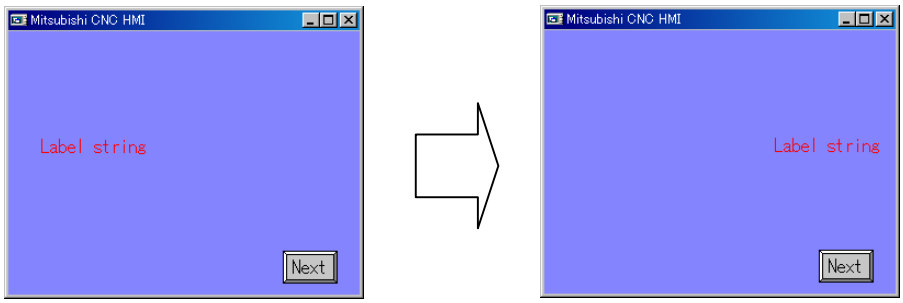
Create the following sample.

Screen Configuration



Action State

Display a label string as flowing characters according to a timer.



From left　　To right

Settings and Action

Create a label on the screen. Specify a skipping timer event to the label. Each time the timer event is triggered, shift the displaying position from left to right to realize flowing characters with the label. In addition, use the program to specify the following actions in callback functions of the control.

OnCreate()　: Issue a 100ms timer event to itself.
OnTimer()　: Shift the label displaying position from left to right at every 100ms.

When the button (character string: "Next") is pressed, switch to the next page. At the time, delete the timer created with OnCreate().

NOTE

- When the label (GLabel00000) is deleted (OnDelete()), the timer is not deleted because the label has been deleted. The timer is deleted when the page is switched.

Source Code

```
long GCPanel00003::GLABEL00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pLabel;				//Label object
	GRect gRect;					//Label rectangle size
	pLabel = GetChild( GLABEL00000 );		//Acquire the label object
	GCSGetBounds( pLabel, &gRect );
	gRect.nXmax= gRect.nXmax-gRect.nXmin;		//Calculate the displaying position
	gRect.nXmin= 0;
	GCSSetBounds( pLabel, &gRect );			//Specify the initial displaying position

	return TRUE;
}
long GCPanel00003::GLABEL00000OnTimer(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pLabel;				//Label object
	GBaseObject *pPanel;				//Panel object
	GRect gRect;					//Label rectangle size
	short nWidth;					//Horizontal label width
	static int nCounter;				//Drawing counter

	nCounter++;
	if(nCounter > 1000) {
		pLabel = GetChild( GLABEL00000 );	//Acquire the label object
		pPanel =				//Acquire the panel object
		GCSGetParent( GetGBaseObject());

		GCSGetBounds( pLabel, &gRect );		//Acquire the label displaying position
		nWidth = gRect.nXmax-gRect.nXmin;
		gRect.nXmax = gRect.nXmin+nWidth;	//Re-calculate the label displaying position
		gRect.nXmin =	(gRect.nXmin+5)%(GCSGetWidth( pPanel ));
		gRect.nXmax =	gRect.nXmin+nWidth;
		GCSSetBounds( pLabel, &gRect );		//Re-specify the label displaying position
		AddRefreshRect(NULL, 0);		//Re-draw the label
		nCounter = 0;
	}
	return TRUE;
}

long GCPanel00003::GBUTTON00001OnClick(unsigned short usMessage, long lLParam, long
lUParam)
{
	GBaseObject *pLabel;				//Label object
	pLabel = GetChild( GLABEL00000 );		//Acquire the label object
							//Switch to the next page

	GESetEvent( GECreateEventMessage( GM_SHOWPANEL,
	GCSGetScreen( GetGBaseObject() ),	4, 0 ) , FALSE );

	return TRUE;
}
```

18.6 GCTextBox (text box)

GCTextBox is a control that displays values and character strings in the designated rectangle and processes entered values and character strings.

Type

The text displayed inside the control includes the followings.

Definition	Value	Description
GTEXT_TYPE_STRING	0x00	Character string
GTEXT_TYPE_SHORT	0x01	Short value
GTEXT_TYPE_USHORT	0x02	Unsigned short value
GTEXT_TYPE_LONG	0x03	Long value
GTEXT_TYPE_ULONG	0x04	Unsigned long value
GTEXT_TYPE_FLOAT	0x05	Floating point value

Shape

The following pieces of information are set in the shape variable of GCBaseWindow.

Definition	Corresponding bit	Description
GTEXT_STYLE_COMMA	Bit 16	Indicates presence of the comma. (0: Absent. 1: Present)
GTEXT_STYLE_REFUSE_NUM	Bit 17	Indicates whether the value is rejected or not. (0: Accepted. 1: Rejected)
GTEXT_STYLE_REFUSE_SML	Bit 18	Indicates whether the single byte lower case letters are rejected or not. (0: Accepted. 1: Rejected)
GTEXT_STYLE_REFUSE_CPT	Bit 19	Indicates whether the upper case letters are rejected or not. (0: Accepted. 1: Rejected)
GTEXT_STYLE_REFUSE_SYM	Bit 20	Indicates whether the single byte symbols are rejected or not. (0: Accepted. 1: Rejected)
GTEXT_STYLE_REFUSE_FULL	Bit 21	Indicates whether the two-byte characters or rejected or not. (0: Accepted. 1: Rejected)
GTEXT_STYLE_CHK_MAX	Bit 22	Indicates whether the maximum limit check is performed or not. (0: Not checked. 1: Checked)
GTEXT_STYLE_CHK_MIN	Bit 23	Indicates whether the minimum limit check is performed or not. (0: Not checked. 1: Checked)
GTEXT_STYLE_PASSWORD	Bit 24	Indicates whether the password is displayed or not. (0: Not displayed. 1: Displayed)

Display

The background of GCTextBox indicates the following states.

Definition	Description
NO_DRAW	The background is not filled.
FILL_BACKCOLOR	Filled in background color

State

The focusing effect of GCTextBox includes the following states.

Definition	Description
GFOCUS_EFFECT_NON	No focusing effect
GFOCUS_EFFECT_SHOW_CURSOR	Cursor display at focus

Importing Structure

The importing structures of GCTextBox are shown below.

```
GDefPropertyClass( GCTextBoxProperty, GControlProperty )
        unsigned short      usFormatID;         /* Display format                */
        GCaption            gcCaption;          /* Nameplate state               */
        unsigned short      usFontID;           /* Font                          */
        unsigned short      usBorderID;         /* Solid frame                   */
        GColor              gcNormalColor;      /* Regular filling color         */
        GColor              gcFocusColor;       /* Focusing filling color        */
        GColor              gcDisableColor;     /* Filling color                 */
        unsigned char       ucType;             /* Type                          */
        unsigned long       ulLimitLength;      /* Max. number of characters     */
        GValue              gvMaxValue;         /* Max. value                    */
        GValue              gvMinValue;         /* Min. value                    */
        short               nBackGroundPattern; /* Background state              */
        unsigned char       ucFocusEffect;      /* Focusing effect               */
GDefPropertyEnd
```

Parent Class

The parent class of GCTextBox is GCControl.

Function List

The functions of GCTextBox include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

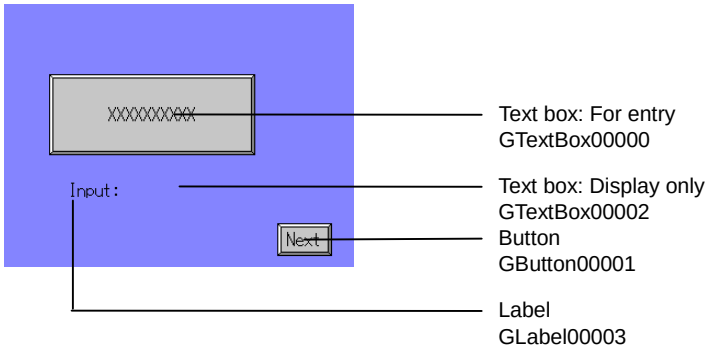
Functions Characteristic to GCTextBox

Function name	Function
GCSTextboxSetTextType	Set character string type
GCSTextboxGetTextType	Get character string type
GCSTextboxSetFormatID	Set character string display format resource ID
GCSTextboxGetFormatID	Get character string display format resource ID
GCSTextboxSetBorderID	Set 3D border resource ID
GCSTextboxGetBorderID	Get 3D border resource ID
GCSTextboxSetFontID	Set font resource ID
GCSTextboxGetFontID	Get font resource ID
GCSTextboxSetCaption	Set caption information
GCSTextboxGetCaption	Get caption information
GCSTextboxSetNormalColor	Set normal background color
GCSTextboxGetNormalColor	Get normal background color
GCSTextboxSetFocusColor	Set background color during focus
GCSTextboxGetFocusColor	Get background color during focus
GCSTextboxSetDisableColor	Set background color when disabled
GCSTextboxGetDisableColor	Get background color when disabled
GCSTextboxSetPasswordStatus	Set password display status
GCSTextboxGetPasswordStatus	Get password display status
GCSTextboxSetCommaStatus	Set comma display status
GCSTextboxGetCommaStatus	Get comma display status
GCSTextboxSetRefuseInputNumberStatus	Set no number input status
GCSTextboxGetRefuseInputNumberStatus	Get no number input status
GCSTextboxSetRefuseInputSmallLetterStatus	Set one-byte lower case character input disabled status
GCSTextboxGetRefuseInputSmallLetterStatus	Get one-byte lower case character input disabled status
GCSTextboxSetRefuseInputCapitalLetterStatus	Set one-byte upper case character input disabled status
GCSTextboxGetRefuseInputCapitalLetterStatus	Get one-byte upper case character input disable status
GCSTextboxSetRefuseInputSymbolLetterStatus	Set one-byte symbol input disabled status
GCSTextboxGetRefuseInputSymbolLetterStatus	Get one-byte symbol input disabled status
GCSTextboxSetRefuseInputFullLetterStatus	Set two-byte character input disabled status
GCSTextboxGetRefuseInputFullLetterStatus	Get two-byte character input disabled status
GCSTextboxSetCheckMaxStatus	Set maximum value check status
GCSTextboxGetCheckMaxStatus	Get maximum value check status
GCSTextboxSetCheckMinStatus	Set minimum value check status
GCSTextboxGetCheckMinStatus	Get minimum value check status
GCSTextboxSetMaxGValue	Set maximum value
GCSTextboxGetMaxGValue	Get maximum value
GCSTextboxSetMinGValue	Set minimum value
GCSTextboxGetMinGValue	Get minimum value
GCSTextboxSetString	Set display character string
GCSTextboxGetString	Get display character string
GCSTextboxSetGValue	Set display value
GCSTextboxGetGValue	Get display value
GCSTextboxSetStringBuffer	Set display buffer
GCSTextboxGetStringBuffer	Get display buffer
GCSTextboxSetBackGroundPattern	Set background fill status
GCSTextboxGetBackGroundPattern	Get background fill status
GCSTextboxSetFocusEffect	Set effect during focus
GCSTextboxGetFocusEffect	Get effect during focus

Usage Example

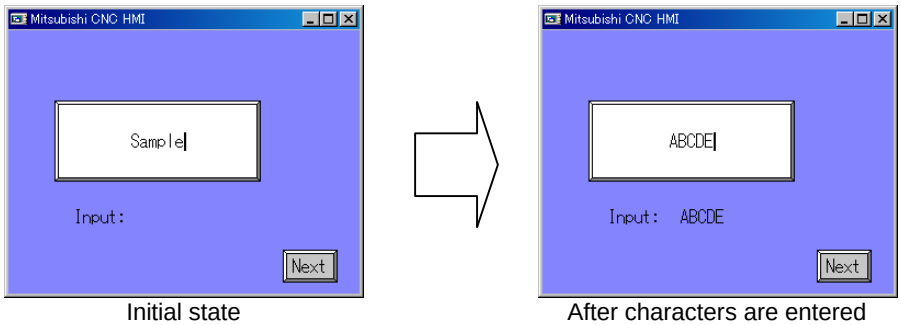
Create the following sample.

Screen Configuration



Action State

Enter "ABCDE" and press Enter.



Initial state　　After characters are entered

Settings and Action

Create an entry text box (GTextBox00000) and displaying text box (GTextBox00002) on the screen. Prohibit entry for the displaying text box (GTextBox00002), using the properties setting. Permit character entry for the entry text box (GTextBox00000) and, each time the ENTER (0x0D) key is pressed during key entry, the comfirmed characters are displayed in the displaying text box (GTextBox00002). Use the program to specify the following actions in callback functions of the control.

OnCreate()　: Display "Sample" as an initial character string.
OnKeyPress(): Acquire the entered character string and display it in the displaying text box.

When the button (character string: "Next") is pressed, switch to the next page.

NOTE

- The character cursor is automatically displayed in the text box control. The following keys are processed inside the text control.

GK_LEFT (left arrow key)	: Move the character cursor to be left.
GK_RIGHT (right arrow key)	: Move the character cursor to be right.
GK_BACKSPACE (BS key)	: Delete one character before the cursor.
GK_DELETE (DeL key)	: Delete one character after the cursor.

Source Code

```
long GCPanel00002::GTEXTBOX00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pText;					//Text box object

	pText = GetChild( GTEXTBOX00000 );		//Acquire the text box object
	GCSTextboxSetString( pText, GRCLoadString( ID_STRING00076 ) ); //Specify the initial character string

	return TRUE;
}
long GCPanel00002::GTEXTBOX00000OnKeyPress(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pText;					//Text box object
	GTCHAR szText[128];					//Character string buffer

	if(lLParam==GK_ENTER){				//Acquire characters only when ENTER is entered
		pText = GetChild( GTEXTBOX00000 );	//Acquire the text box object
		GCSTextboxGetString( pText, szText, sizeof(szText) );//Acquire entered characters.
		pText = GetChild( GTEXTBOX00002 );	//Acquire the text box object
		GCSTextboxSetString( pText, szText );	//Display the entered characters
	}

	return TRUE;
}
```

18.7 GCList (list)

GCList is a control that displays a list and allows selection of an item in it.

Importing Structure

The importing structures of GCList are as shown below.

```
GDefPropertyClass( GListProperty, GControlProperty )
        Gcaption        gcCaption;          /* Caption                              */
        unsigned short  usFontID;           /* Font resource ID                     */
        unsigned short  usBorderID;         /* Solid frame resource ID              */
        GColor          gcNormalColor;      /* Regular background color             */
        GColor          gcFocusColor;       /* Background color of focused object   */
        GColor          gcDisableColor;     /* Background color of disabled object  */
        GColor          gcScrollBarColor;   /* Scroll bar color                     */
        GColor          gcScrollButtonColor;/* Scroll button color                  */
        unsigned char   ucScrollBarWidth;   /* Scroll bar width                     */
        GColor          gcSelectBarColor;   /* Selection bar color                  */
        unsigned short  usMaxListLines;     /* Max. number of lines                 */
GDefPropertyEnd
```

Parent Class

The parent class of GCList is GCControl.

Function List

The functions of GCList include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

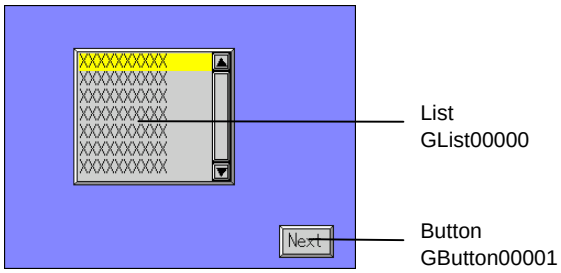
Functions Characteristic to GCList

Function name	Function
GCSListSetBorderID	Set 3D border resource ID
GCSListGetBorderID	Get 3D border resource ID
GCSListSetFontID	Set font resource ID
GCSListGetFontID	Get font resource ID
GCSListSetCaption	Set caption information
GCSListGetCaption	Get caption information
GCSListSetNormalColor	Set normal background color
GCSListGetNormalColor	Get normal background color
GCSListSetFocusColor	Set background color during focus
GCSListGetFocusColor	Get background color during focus
GCSListSetDisableColor	Set background color when disabled
GCSListGetDisableColor	Get background color when disabled
GCSListSetScrollBarColor	Set scroll bar color
GCSListGetScrollBarColor	Get scroll bar color
GCSListSetScrollButtonColor	Set scroll button color
GCSListGetScrollButtonColor	Get scroll button color
GCSListSetScrollBarWidth	Set scroll bar width
GCSListGetScrollBarWidth	Get scroll bar width
GCSListSetSelectBarColor	Set select bar color
GCSListGetSelectBarColor	Get select bar color
GCSListSetMaxListLines	Set maximum numter of lines in list
GCSListGetMaxListLines	Get maximum numter of lines in list
GCSListImportProperty	Import property settings
GCSListOnDraw	Client area drawing process
GCSListAddString	Add list character string
GCSListInsertString	Insert list character string
GCSListRemoveString	Delete list character string
GCSListGetString	Get list character string
GCSListRemoveAllStrings	Delete all list character strings
GCSListSetCurrentSelect	Set selected line
GCSListGetCurrentSelect	Get selected line
GCSListGetLineCount	Get number of lines in list
GCSListOnLButtonPress	Client area process for pressing left mouse button
GCSListOnLButtonRelease	Client area process for releasing left mouse button
GCSListOnScroll	Scroll bar call-back processing
GCSListKeyPress	Key press processing
GCSListSetVisibleScrollBarStatus	Set scroll bar display/non-display status
GCSListGetVisibleScrollBarStatus	Get scroll bar display/non-display status
GCSListSetEnableScrollBarStatus	Set scroll bar controllable/uncontrollable status
GCSListGetEnableScrollBarStatus	Get scroll bar controllable/uncontrollable status
GCSListSetTopLine	Set list top line
GCSListGetTopLine	Get list top line
GCSListCheckBorder	Confirm object border status
GCSListCheckStyle	Confirm object status color

Usage Example

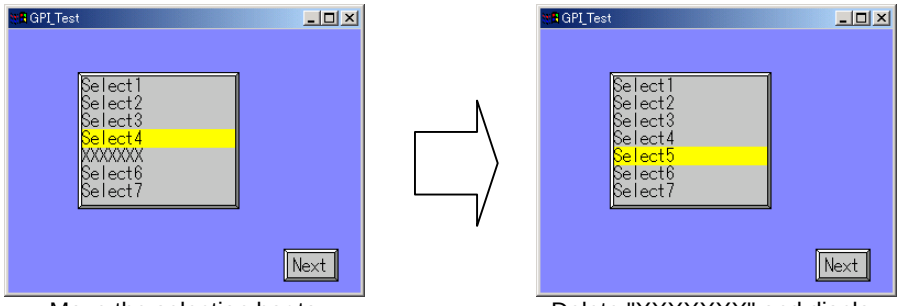
Create the following screen.

Screen Configuration



Action State

Move the selection bar to "XXXXXXX" and click.



Move the selection bar to "XXXXXXX."

Delete "XXXXXXX" and display "Select5."

Settings and Action

Arrange a list on the screen. When arranged, there is no item in the list. Register selection items during list generation. Click the mouse button on the "XXXXXXX" item to select it. "XXXXXXX" is deleted and the "Select5" item is added to the list. During generation, delete the side scroll bar. Use the program to specify the following actions in callback functions of the control.

OnCreate() : Register selection items to the list during generation.
OnClick() : Delete an item when the mouse button is clicked, and add a new list item.

When the button (character string: "Next") is pressed, switch to the next page.

NOTE

◆ The selection bar moves automatically at the list control. Use the following keys during the procedure.

GK_UP (up arrow key) : Move the selection bar up.
GK_DOWN (down arrow key) : Move the selection bar down.

Source Code

```
long GCPanel00004::GLIST00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
    GBaseObject *pList = GetChild( GLIST00000 );

    GCSListSetVisibleScrollBarStatus( pList, FALSE );    //Delete the scroll bar

    GCSListAddString( pList, GRCLoadString( ID_STRING00078 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00079 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00080 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00081 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00088 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00083 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00084 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00085 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00086 ) );//Add a selection character string at
    the scroll bar
    GCSListAddString( pList, GRCLoadString( ID_STRING00087 ) );//Add a selection character string at
    the scroll bar

    GCSListSetCurrentSelect( pList, 3 );                    // Designate the first selection line position

    return TRUE;
}

long GCPanel00004::GLIST00000OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
    GBaseObject *pList = GetChild( GLIST00000 );
    short nIndex;                                           //Selection line index
    GTCHAR szText[80];                                      //Selection line character string

    nIndex = (short)GCSListGetCurrentSelect( pList );    //Acquire the current selection line
    GCSListGetListString( pList, nIndex, szText, sizeof(szText) ); //Acquire the current selection
    character string
    if(!wcscmp(szText, L"XXXXXXX")){                        //If the selection line is "XXXXXXX,"
        GCSListRemoveString( pList, nIndex );               //Delete the item
                                                            //Add a selection character string at the scroll bar
        nIndex = GCSListInsertString( pList, nIndex, GRCLoadString( ID_STRING00082 ) );
    }

    return TRUE;
}
```

18.8 GCCheckBox (checkbox)

GCCheckBox is a control that holds the ON/OFF state and accepts state change requests of the user, upon which it displays the ON/OFF state.

Display

The state of GCCheckBox includes the followings.

Definition	Value	Description
GCHK_STATE_OFF	0x00	OFF state
GCHK_STATE_ON	0x01	ON state
GCHK_STATE_OFFSELECT	0x02	The button in the OFF state is pressed. If the button is released, transition to the ON state occurs. If other than the button is released, transition to the OFF state occurs.
GCHK_STATE_ONSELECT	0x03	The button in the ON state is pressed. If the button is released, transition to the OFF state occurs. If other than the button is released, transition to the ON state occurs.

Importing Structure

The importing structures of GCCheckBox are as shown below.

```
GDefPropertyClass( GCheckBoxProperty, GControlProperty )
        unsigned short      usStringID;          /* Character string resource ID */
        GCaption            gcCaption;           /* Caption */
        unsigned short      usFontID;            /* Font resource ID */
        unsigned char       ucFocusEffect;       /* Focusing effect */
        GColor              gcFocusColor;        /* Background color of focused
                                                    object */
        GColor              gcDisableBoxColor;   /* Color of disabled box */
        GColor              gcDisableCaptionColor; /* Color of disabled caption
                                                    character */
        unsigned short      usBoxSize;           /* Box size */
        unsigned short      usBorderID;          /* Solid frame resource ID */
        GColor              gcBoxColor;          /* Box color */
GDefPropertyEnd
```

Parent Class

The parent class of GCCheckBox is GCControl.

Function List

The functions of GCCheckBox include the followings.

Source Code

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

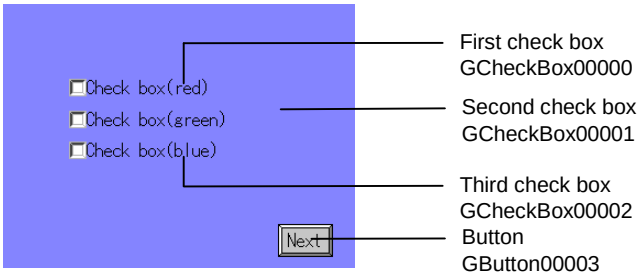
Functions Characteristic to GCCheckBox

Function name	Function
GCSCheckboxSetBoxSize	Set box size
GCSCheckboxGetBoxSize	Get box size
GCSCheckboxSetBorderID	Set 3D border resource ID
GCSCheckboxGetBorderID	Get 3D border resource ID
GCSCheckboxSetFontID	Set font resource ID
GCSCheckboxGetFontID	Get font resource ID
GCSCheckboxSetStringID	Set caption character string resource ID
GCSCheckboxGetStringID	Get caption character string resource ID
GCSCheckboxSetCaption	Set caption information
GCSCheckboxGetCaption	Get caption information
GCSCheckboxSetBoxColor	Set box color
GCSCheckboxGetBoxColor	Get box color
GCSCheckboxSetFocusColor	Set background color during focus
GCSCheckboxGetFocusColor	Get background color during focus
GCSCheckboxSetDisableCaptionColor	Set character color when disabled
GCSCheckboxGetDisableCaptionColor	Get character color when disabled
GCSCheckboxSetDisableBoxColor	Set box color when disabled
GCSCheckboxGetDisableBoxColor	Get box color when disabled
GCSCheckboxSetFocusEffect	Set effect during focus
GCSCheckboxGetFocusEffect	Get effect during focus
GCSCheckboxSetStatus	Set object status
GCSCheckboxGetStatus	Get object status
GCSCheckboxStartCaptionScroll	Start caption character string scroll operation
GCSCheckboxStopCaptionScroll	Stop caption character string scroll operation
GCSCheckboxPauseCaptionScroll	Pause caption character string scroll operation
GCSCheckboxRestartCaptionScroll	Restart caption character string scroll operation
GCSCheckboxGetCaptionScrollStatus	Get caption character string scroll status
GCSCheckboxIsCaptionOutOfBounds	Confirm overflowing of caption character string
GCSCheckboxSetCaptionScrollDelayTime	Set scroll delay time
GCSCheckboxGetCaptionScrollDelayTime	Get scroll delay time
GCSCheckboxSetCaptionScrollRefreshTime	Set scroll refresh time
GCSCheckboxGetCaptionScrollRefreshTime	Get scroll refresh time
GCSCheckboxSetCaptionScrollMovementValue	Set scroll movement value
GCSCheckboxGetCaptionScrollMovementValue	Get scroll movement value
GCSCheckboxSetCaptionScrollStartPosition	Set scroll start position
GCSCheckboxGetCaptionScrollStartPosition	Get scroll start position
GCSCheckboxSetCaptionScrollPosition	Set caption character string scroll position
GCSCheckboxGetCaptionScrollPosition	Get caption character string scroll position
GCSCheckboxStartCaptionBlink	Start caption character string blink
GCSCheckboxStopCaptionBlink	Stop caption character string blink
GCSCheckboxSetBlinkInterval	Set blink interval
GCSCheckboxGetBlinkInterval	Get blink interval
GCSCheckboxSetBlinkType	Set blink type
GCSCheckboxGetBlinkType	Get blink type

Usage Example

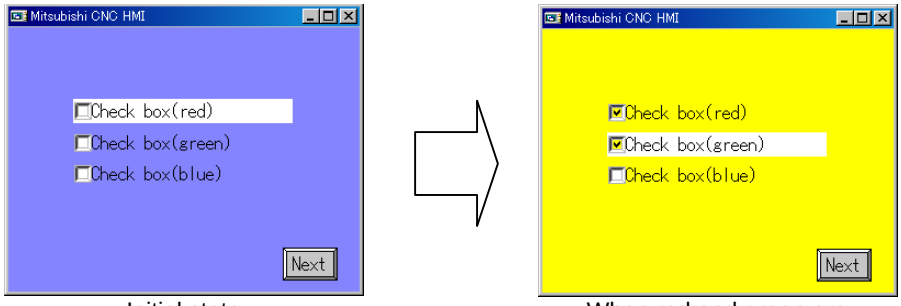
Create the following sample.

Screen Configuration



Action State

Change the background color of the panel when each check box is selected.



Initial state

When red and green are selected, the background color becomes yellow.

Settings and Action

Create three check boxes on the screen. When each item is selected, the inside of the check box becomes yellow and the corresponding color is added to the background color of the panel. For example, when red and green are selected, the background color becomes yellow. Use the program to specify the following action in callback function of the control.

OnClick() : Change the color of the inside of the check box and that of the background of the panel when the mouse button is clicked.

When the button (character string: "Next") is pressed, switch to the next page.

NOTE

- Processes for the three check boxes in the following sample program are similar. Only the program sample for the first check box (GCheckBox00000) is shown here.

Source Code

```
long GCPanel00006::GCHECKBOX00000OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
	UpdateBackground();					//Refresh the background screen

	return TRUE;
}

//This function changes the color of the background of the panel according to the designated check box.
void GCPanel00006::UpdateBackground( void )
{
	GBaseObject *pCheck[3];
	GBaseObject *pPanel;				//Panel object
	GDesign *pDesign;				//Background design
	GColor gColor;					//Background color variable

	GetCheckBox(pCheck);
	pPanel = GCSGetPanel( GCSGetScreen( GetGBaseObject() ) );//Acquire the panel object
	pDesign = GCSGetBackGroundDesign( pPanel );	//Acquire the background design

	gColor =0;
	if(GCSGetStatus( pCheck[0] )){			//Acquire the state of the check box
		GCSSetBoxColor( pCheck[0], RGB32(0xFF,0xFF,0x00) );
							//Change the color of the check box (yellow)
		gColor =RGB32(0xFF, 0x00, 0x00);	//Add a red color
	}else{
		GCSSetBoxColor( pCheck[0], RGB32(0xFF,0xFF,0xFF) );
							//Change the color of the check box (white)
	}
	if(GCSGetStatus( pCheck[1] )){			//Acquire the state of the check box
		GCSSetBoxColor( pCheck[1], RGB32(0xFF,0xFF,0x00) );
							//Change the color of the check box (yellow)
		gColor =RGB32(0x00, 0xFF, 0x00);	//Add a green color
	}else{
		GCSSetBoxColor( pCheck[1], RGB32(0xFF,0xFF,0xFF) );
							//Change the color of the check box (white)
	}
	if(GCSGetStatus( pCheck[2] )){			//Acquire the state of the check box
		GCSSetBoxColor( pCheck[2], RGB32(0xFF,0xFF,0x00) );
							//Change the color of the check box (yellow)
		gColor =RGB32(0x00, 0x00, 0xFF);	//Add a blue color.
	}else{
		GCSSetBoxColor( pCheck[2], RGB32(0xFF,0xFF,0xFF) );
							//Change the color of the check box (white)
	}
	pDesign->gbBrush.gcBackColor = gColor;		//Synthesize the background color

	GCSAddRefreshRect( pPanel, NULL, 0 );		//Re-draw the background
}

//This function acquires each check box object.
void GCPanel00006::GetCheckBox(GBaseObject *pCheck[3])
{
	pCheck[0] = GetChild( GCPanel00006::GCHECKBOX00000 );//Acquire the check box object
	pCheck[1] = GetChild( GCPanel00006::GCHECKBOX00001 );//Acquire the check box object
	pCheck[2] = GetChild( GCPanel00006::GCHECKBOX00002 );//Acquire the check box object
}
```

18.9 GCRadioButton (radio button)

GCRadioButton is a control that specifies a group of multiple buttons and realize exclusive selection.

State

The state of GCRadioButton includes the followings.

Definition	Value	Description
GRADIO_STATE_OFF	0x00	OFF state
GRADIOSTATE_ON	0x01	ON state
GRADIO_STATE_OFFSELECT	0x02	The button in the OFF state is pressed. When the button is released, transition to the ON state occurs.When released other than at the button, transition to the OFF state occurs.

Importing Structure

The importing structures of GCRadioButton are as shown below.

```
GDefPropertyClass( GRadioButtonProperty, GControlProperty )
        unsigned short      usStringID;          /* Character string resource ID   */
        GCaption            gcCaption;           /* Caption                        */
        unsigned short      usFontID;            /* Font resource ID               */
        unsigned char       ucFocusEffect;       /* Focusing effect                */
        GColor              gcFocusColor;        /* Background color of            */
                                                 focused object
        GColor              gcDisableBoxColor;   /* Color of disabled box          */
        GColor              gcDisableCaptionColor; /* Color of disabled caption    */
                                                 character
        unsigned short      usRadioGroup;        /* Radio group No.                */
        unsigned short      usBoxSize;           /* Box size                       */
        unsigned short      usBorderID;          /* Solid frame resource ID        */
        GColor              gcBoxColor;          /* Box color                      */
GDefPropertyEnd
```

Parent Class

The parent class of GCRadioButton is GCControl.

Function List

The functions of GCRadioButton include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

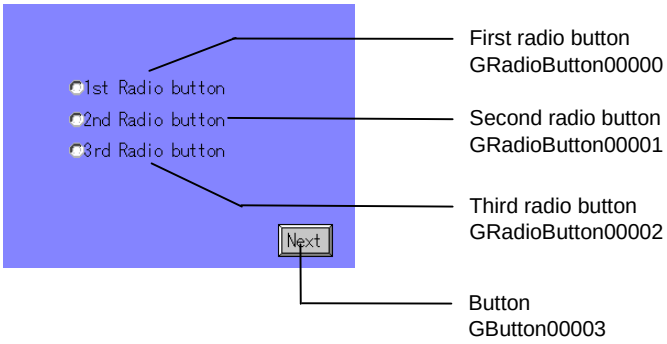
Functions Characteristic to GCRadioButton

Function name	Function
GCSRadiobuttonSetBoxSize	Set box size
GCSRadiobuttonGetBoxSize	Get box size
GCSRadiobuttonSetRadioGroup	Set radio group No.
GCSRadiobuttonGetRadioGroup	Get radio group No.
GCSRadiobuttonSetBorderID	Set 3D border resource ID
GCSRadiobuttonGetBorderID	Get 3D border resource ID
GCSRadiobuttonSetFontID	Set font resource ID
GCSRadiobuttonGetFontID	Get font resource ID
GCSRadiobuttonSetStringID	Set caption character string resource ID
GCSRadiobuttonGetStringID	Get caption character string resource ID
GCSRadiobuttonSetCaption	Set caption information
GCSRadiobuttonGetCaption	Get caption information
GCSRadiobuttonSetBoxColor	Set box color
GCSRadiobuttonGetBoxColor	Get box color
GCSRadiobuttonSetFocusColor	Set background color during focus
GCSRadiobuttonGetFocusColor	Get background color during focus
GCSRadiobuttonSetDisableCaptionColor	Set character color when disabled
GCSRadiobuttonGetDisableCaptionColor	Get character color when disabled
GCSRadiobuttonSetDisableBoxColor	Set box color when disable
GCSRadiobuttonGetDisableBoxColor	Get box color when disable
GCSRadiobuttonSetFocusEffect	Set effect during focus
GCSRadiobuttonGetFocusEffect	Get effect during focus
GCSRadiobuttonImportProperty	Import property settings
GCSRadiobuttonOnDraw	Client area drawing process
GCSRadiobuttonOnLButtonPress	Client area process for pressing left mouse button
GCSRadiobuttonOnLButtonRelease	Client area process for releasing left mouse button
GCSRadiobuttonSetStatus	Set object status
GCSRadiobuttonGetStatus	Get object status

Usage Example

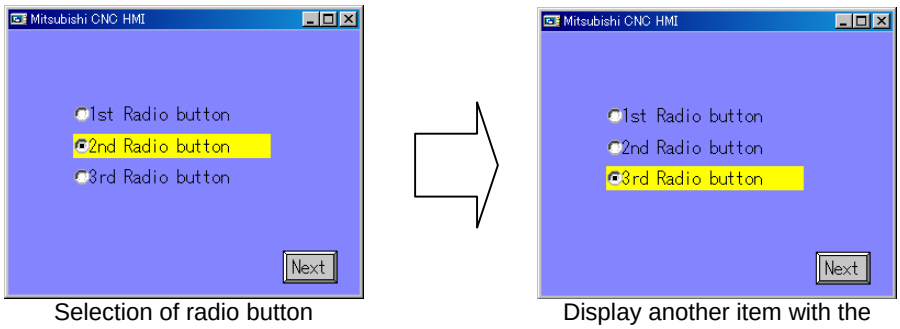
Create the following sample.

Screen Configuration



Action State

Select each radio button and change the item character string, using the up and down keys.



Selection of radio button

Display another item with the down arrow.

Settings and Action

Create three radio buttons on the screen. Specify yellow as a background color of the focused character and, change the radio button character string with the up or down arrow key in the selected state. Use the program to specify the following action in callback function of the control.

OnKeyPress() : Switch the displayed character string when the up or down arrow key is pressed.

When the button (character string: "Next") is pressed, switch to the next page.

NOTE

- Processes for the three radio buttons in the following sample program are similar. Only the program sample for the first radio button (GRadioButtOn00000) is shown here.

Source Code

```
long GCPanel00007::GRADIOBUTTON00000OnKeyPress(unsigned short usMessage,
                                                long lLParam, long lUParam)
{
        GBaseObject *pRadio0;                   //Radio button object
        GBaseObject *pRadio1;                   //Radio button object
        GBaseObject *pRadio2;                   //Radio button object

        if(lLParam==GK_DOWN){                   //Change the caption with the down arrow
                pRadio0 = GetChild( GRADIOBUTTON00000 );
                GCSRadiobuttonSetStringID( pRadio0, ID_STRING00055 );
                pRadio1 = GetChild( GRADIOBUTTON00001 );
                GCSRadiobuttonSetStringID( pRadio1, ID_STRING00056 );
                pRadio2 = GetChild( GRADIOBUTTON00002 );
                GCSRadiobuttonSetStringID( pRadio2, ID_STRING00057 );
        }else if(lLParam==GK_UP){               //Return the caption with the up or down arrow
                pRadio0 = GetChild( GRADIOBUTTON00000 );
                GCSRadiobuttonSetStringID( pRadio0, ID_STRING00035 );
                pRadio1 = GetChild( GRADIOBUTTON00001 );
                GCSRadiobuttonSetStringID( pRadio1, ID_STRING00038 );
                pRadio2 = GetChild( GRADIOBUTTON00002 );
                GCSRadiobuttonSetStringID( pRadio2, ID_STRING00039 );
        }

        return TRUE;
}
```

18.10 GCProgressBar (progress bar)

GCProgressBar is a control that indicates an increase or decrease of the value with the filled area.

Direction

The direction of progress of GCProgressBar includes the followings.

Definition	Value	Description
GPROG_DIR_LEFT2RIGHT	0x00	Filling from left to right
GPROG_DIR_RIGHT2LEFT	0x01	Filling from right to left
GPROG_DIR_TOP2BOTTOM	0x02	Filling from top to bottom
GPROG_DIR_BOTTOM2TOP	0x03	Filling from bottom to top

Importing Structure

The importing structures of GCProgressBar are shown below.

```
GDefPropertyClass( GProgressBarProperty, GControlProperty )
        GDesign         gdBackGroundDesign;   /* Background design   */
        GDesign         gdBarDesign;          /* Bar design          */
        unsigned char   ucDirection;          /* Filling direction   */
        short           nMin;                 /* Min. value          */
        short           nMax;                 /* Max. value          */
GDefPropertyEnd
```

Parent Class

The parent class of GCProgressBar is GCControl.

Function List

The functions of GCProgressBar include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

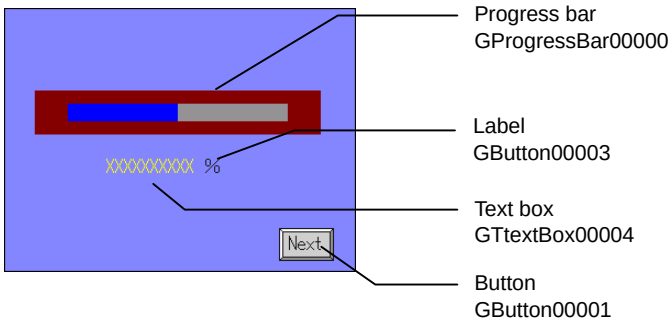
Functions Characteristic to GCProgressBar

Function name	Function
GCSProgressbarSetBackGroundDesign	Set background design
GCSProgressbarGetBackGroundDesign	Get background design
GCSProgressbarSetBarDesign	Set bar design
GCSProgressbarGetBarDesign	Get bar design
GCSProgressbarSetDirection	Set fill direction
GCSProgressbarGetDirection	Get fill direction
GCSProgressbarSetRange	Set progress bar range
GCSProgressbarGetRange	Get progress bar range
GCSProgressbarImportProperty	Import property settings
GCSProgressbarOnDraw	Client area drawing process
GCSProgressbarSetValue	Set current progress bar value
GCSProgressbarGetValue	Get current progress bar value

Usage Example

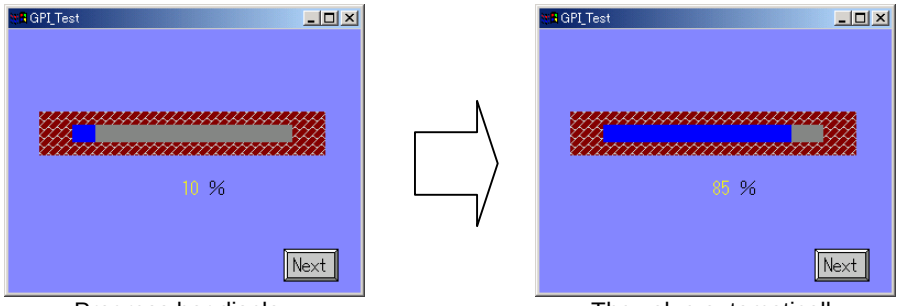
Create the following sample.

Screen Configuration



Action State

A timer automatically increases the value in the progress bar.



Progress bar display

The value automatically increases.

Settings and Action

Create a progress bar on the screen and arrange an entry-prohibited text box below the progress bar. Create a 500ms timer when the progress bar is created, and increase the progress bar value repetitively between 0 and 100. Display the current progress bar value in the text box. Use the program to specify the following actions in callback functions of the control.

OnCreate() : Create a 500ms timer.
OnTimer() : Increase the progress bar value and refresh the display.

When the button (character string: "Next") is pressed, switch to the next page. At the time, delete the timer.

NOTE

- When the progress bar (GProgressBar00000) is deleted (OnDelete()), the timer is not deleted because the progress bar is deleted. Delete the timer when the page is switched.

Source Code

```
long GCPanel00008::GPROGRESSBAR00000OnCreate(unsigned short usMessage,
                                                    long lLParam, long lUParam)
{
        GBaseObject *pProgress;                     //Progress bar object
        pProgress = GetChild( GPROGRESSBAR00000 );

        return TRUE;
}

long GCPanel00008::GPROGRESSBAR00000OnTimer(unsigned short usMessage, long lLParam,
                                                        long lUParam)
{
        GBaseObject *pProgress;                     //Progress bar object
        GBaseObject *pTextBox;                      //Text box object
        GValue gValue;                              //Current value for displaying the text
        short nValue;                               //Current value for progress bar
        static int nCounter;                        //Drawing counter

        nCounter++;
        if(nCounter > 10000) {
                pProgress = GetChild( GPROGRESSBAR00000 );
                pTextBox = GetChild( GTEXTBOX00004 );
        nValue = GCSProgressbarGetValue( pProgress );   //Acquire the current value
        nValue = nValue>=100 ? 0:nValue+5;          //Increment
        gValue.nValue = nValue;                     //Current value for displaying the text
        GCSTextboxSetGValue( pTextBox, gValue );    //Refresh the text
                GCSProgressbarSetValue( pProgress, nValue );    //Refresh the progress bar
        nCounter = 0;
        }

        return TRUE;
}

long GCPanel00008::GBUTTON00001OnClick(unsigned short usMessage, long lLParam, long
lUParam)
{
        GBaseObject *pProgress;                     //Progress bar object
        pProgress = GetChild( GPROGRESSBAR00000 );
                                                    //Switch to the next page
        GESetEvent( GECreateEventMessage( GM_SHOWPANEL,
        GCSGetScreen( GetGBaseObject() ),    9, 0 ) , FALSE );

        return TRUE;
}
```

18.11 GCHtmlBrowser (HTML browser)

GCHtmlBrowser is a control that interprets the HTML file and displays it on the screen.

Importing Structure

The importing structures of GCHtmlBrowser are shown below.

```
GDefPropertyClass( GHtmlBrowserProperty, GControlProperty )
        GColor          gcTextColor;        /* Text color                  */
        GColor          gcBackColor;        /* Background color            */
        GColor          gcLinkColor;        /* Link color                  */
        unsigned short  usBorderID;         /* Solid frame resource ID     */
        unsigned short  usFontID;           /* Font resource ID            */
        unsigned short  usFolderID;         /* Folder identification No.   */
        unsigned short  usFileID;           /* File resource ID            */
        unsigned char   ucScrollDispType;   /* Scroll bar display type     */
        GColor          gcScrollBar;        /* Scroll bar color            */
        GColor          gcScrollBarButton;  /* Scroll bar button color     */
        GColor          gcScrollBarArrow;   /* Scroll bar arrow color      */
        unsigned short  usImage1OnID;       /* Design 1ON resource ID      */
        unsigned short  usImage1OffID;      /* Design 1OFF resource ID     */
        unsigned short  usImage2OnID;       /* Design 2ON resource ID      */
        unsigned short  usImage2OffID;      /* Design 2OFF resource ID     */
        unsigned short  usImage3OnID;       /* Design 30N resource ID      */
        unsigned short  usImage3OffID;      /* Design 3OFF resource ID     */
        unsigned short  usImage4OnID;       /* Design 4ON resource ID      */
        unsigned short  usImage4OffID;      /* Design 4OFF resource ID     */
        GColor          gcScrollBarBorder;  /* Scroll bar border color     */
        GColor          gcScrollBarBack;    /* Scroll bar background color */
        unsigned char   ucScrollBarWidth;   /* Scroll bar width            */
GdefPropertyEnd
```

Parent Class

The parent class of GCHtmlBrowser is GCControl.

Function List

The functions of GCHtmlBrowser include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

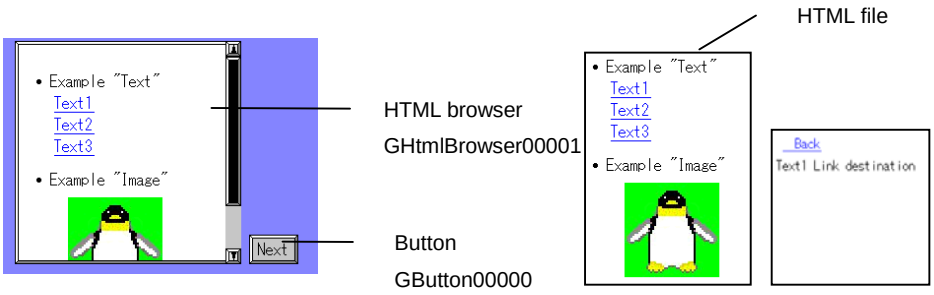
Functions Characteristic to GCProgressBar

Function name	Function
GCSHtmlbrowserSetBorderID	Set solid frame resource ID
GCSHtmlbrowserGetBorderID	Get slid frame resource ID
GCSHtmlbrowserSetTextColor	Set text color
GCSHtmlbrowserGetTextColor	Get text color
GCSHtmlbrowserSetBackColor	Set background color
GCSHtmlbrowserGetBackColor	Get background color
GCSHtmlbrowserSetLinkColor	Set link color
GCSHtmlbrowserGetLinkColor	Get link color
GCSHtmlbrowserSetHtmlFileName	Set display HTML file name setting
GCSHtmlbrowserGetHtmlFileName	Get display HTML file name acquisition
GCSHtmlbrowserSetScrollBarWidth	Set scroll bar width setting
GCSHtmlbrowserGetScrollBarWidth	Get scroll bar width acquisition

Usage Example

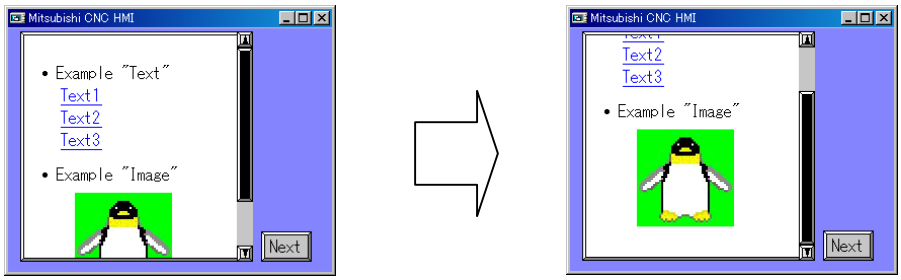
Create the following sample.

Screen Configuration



Action State

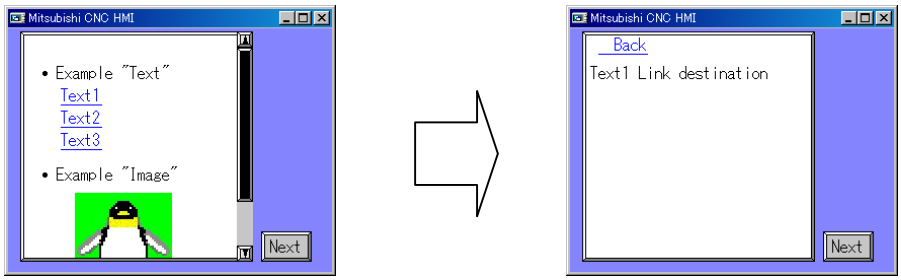
Click on the vertical scroll bar.



Click on the vertical scroll bar.

The display area moves up or down.

Click on the link part.



Click on the link part.

Jump to the linked HTML file.

Settings and Action

Arrange an HTML browser on the screen. Copy the HTML file to the directory designated in HTMLDATA1 of [HTML_BROUWSER] section of config.ini. Change the "HTML file" property to HTMLDATA1 and change the "HTML folder" property to "index.htm." Because the displaying size of the HTML file is larger in the vertical direction than the displaying size of the HTML browser, the vertical scroll bar moves up and down. In addition, you can jump from a link inside the HTML file to another HTML file in the HTMLDATA1 folder.

When the button (character string: "Next") is pressed, switch to the next page.

18.12 GCScrollBarEx (scroll bar)

GCScrollBarEx is a control that moves the display area up, down, left or right inside the screen.

Display

GCScrollBarEx includes the following.

Definition	Value	Description
GSBAR_DISP_NORMAL	0x00	Filling
GSBAR_DISP_IMAGE	0x01	Image display

Scroll Bar Direction

GCScrollBarEx includes the following.

Definition	Value	Description
GSBAR_VERTICAL	0x00	Vertical scroll bar
GSBAR_HORIZONTAL	0x01	Horizontal scroll bar

Importing Structure

The importing structures of GCScrollBarEx are shown below.

```
GDefPropertyClass( GScrollBarExProperty, GScrollBarProperty )
        unsigned char       ucType              /* Direction of scroll bar          */
        unsigned short      usBar1ID;           /* ID of bar design 1               */
        unsigned short      usBar2ID;           /* ID of bar design 2               */
        unsigned short      usBar3ID;           /* ID of bar design 3               */
        GColor              gcScrollArrowColor; /* Color of button arrow            */
        unsigned short      usPrevBtnONImgID;   /* ON design ID of button 1         */
        unsigned short      usPrevBtnOFFImgID;  /* OFF design ID of button 1        */
        unsigned short      usNextBtnONImgID;   /* ON design ID of button 2         */
        unsigned short      usNextBtnOFFImgID;  /* OFF design ID of button 2        */
        GColor              gcBackColor;        /* Background color                 */
        unsigned short      usBackID;           /* ID of background design          */
        unsigned short      usPageSize;         /* Page size                        */
        GColor              gcOutLineColor;     /* Color of border line             */
GdefPropertyEnd
```

Parent Class

The parent class of GCScrollBarEx is GCControl.

Function List

The functions of GCScrollBarEx include the following.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

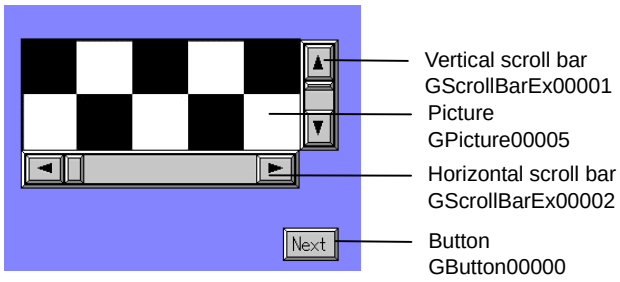
Functions Characteristic to GCProgressBar

Function name	Function
GCSScrollbarexSetBorderID	Set 3D border resource ID
GCSScrollbarexGetBorderID	Get 3D border resource ID
GCSScrollbarexSetBarColor	Set scroll bar color
GCSScrollbarexGetBarColor	Get scroll bar color
GCSScrollbarexSetButtonColor	Set button color
GCSScrollbarexGetButtonColor	Get button color
GCSScrollbarexSetBackGroundColor	Set background color
GCSScrollbarexGetBackGroundColor	Get background color
GCSScrollbarexSetScrollPosition	Set scroll bar current position
GCSScrollbarexGetScrollPosition	Get scroll bar current position
GCSScrollbarexSetRange	Set scroll bar range (max. value, min. value)
GCSScrollbarexGetRange	Get scroll bar range (max. value, min. value)
GCSScrollbarexSetPageSize	Set page size
GCSScrollbarexGetPageSize	Get page size
GCSScrollbarexSetEnableStatus	Set scroll bar controllable/uncontrollable status
GCSScrollbarexGetEnableStatus	Get scroll bar controllable/uncontrollable status

Usage Example

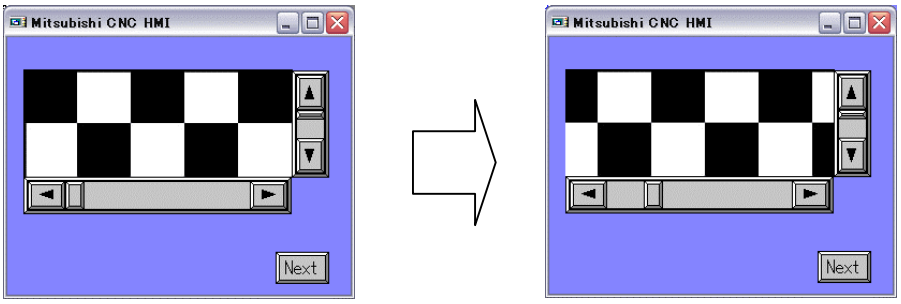
Create the following screen.

Screen Configuration



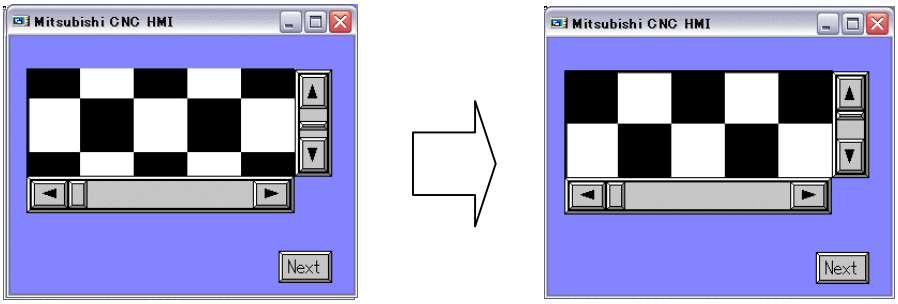
Action State

Click on the horizontal scroll bar.



Click on the horizontal scroll bar. The picture moves left or right.

Click on the vertical scroll bar.



Click on the vertical scroll bar. The picture moves up or down.

Settings and Action

Arrange a horizontal scroll bar, vertical scroll bar and picture on the screen. Use the program of the horizontal and vertical scroll bars to specify the following actions to the callback functions of the control.

OnClick() : Move the picture up, down, left or right when the mouse button is clicked.
OnScroll() : Move the picture up, down, left or right when the mouse button is clicked.

When the button (character string: "Next") is pressed, switch to the next page.

Source Code

```
long GCPanel00011::GSCROLLBAREX00001OnClick(unsigned short usMessage, long lLParam,
                                                                    long lUParam)
{
        GBaseObject *pPicture;                          //Picture object
        GBaseObject *pScrollBar;                        //Scroll bar object
        GRect gRect;                                    //Picture rectangle size
        short nScroll;                                  //Current position of scroll

        pPicture = GetChild(GPICTURE00005);             //Acquire the picture object
        pScrollBar = GetChild(GSCROLLBAREX00001);
        nScroll = GCSGetScrollPosition(pScrollBar);     //Acquire the current position of the
                                                        scroll
        GCSGetBounds(pPicture,&gRect);                  //Acquire the picture displaying position
        gRect.nYmin = -(nScroll*0.5) +PIC_Y_POSITION;   //From the initial Y coordinate of the
                                                        picture
        gRect.nYmax = gRect.nYmin+PIC_HEIGHT;           //Change the coordinate by the scrolling
                                                        position
        GCSSetBounds(pPicture,&gRect);                  //Re-specify the picture displaying
                                                        position
        AddRefreshRect(NULL, 0);                        //Re-draw the picture

        return TRUE;
}

long GCPanel00011::GSCROLLBAREX00001OnScroll(unsigned short usMessage, long lLParam,
                                                                    long lUParam)
{
        GBaseObject *pPicture;                          //Picture object
        GBaseObject *pScrollBar;                        //Scroll bar object
        GRect gRect;                                    //Picture rectangle size
        short nScroll;                                  //Current position of scroll

        pPicture = GetChild(GPICTURE00005);             //Acquire the picture object
        pScrollBar    = GetChild(GSCROLLBAREX00001);    //Acquire the scroll bar object
        nScroll = GCSGetScrollPosition(pScrollBar);     //Acquire the current position of the
                                                        scroll
        GCSGetBounds(pPicture,&gRect);                  //Acquire the picture displaying position
        gRect.nYmin = -(nScroll*0.5) + PIC_Y_POSITION;  //Calculate the Y coordinate of the
                                                        picture
        gRect.nYmax = gRect.nYmin+ PIC_HEIGHT;
        GCSSetBounds(pPicture,&gRect);                  //Re-specify the picture displaying
                                                        position
        GCSAddRefreshRect(GCSGetScreen( m_pSelf ),NULL, 0);       //Re-draw the picture

        return TRUE;
}
```

```
long GCPanel00011::GSCROLLBAREX00002OnClick(unsigned short usMessage, long lLParam,
                                                                    long lUParam)
{
        GBaseObject *pPicture;                          //Picture object
        GBaseObject *pScrollBar;                        //Scroll bar object
        GRect gRect;                                    //Picture rectangle size
        short nScroll;                                  //Current position of scroll

        pPicture = GetChild(GPICTURE00005);             //Acquire the picture object
        pScrollBar    = GetChild(GSCROLLBAREX00002);
        nScroll = GCSGetScrollPosition(pScrollBar);     //Acquire the current position of the
                                                        scroll
        GCSGetBounds(pPicture,&gRect);                  //Acquire the picture displaying position
        gRect.nXmin = -nScroll + PIC_X_POSITION;        //From the initial X coordinate of the
                                                        picture
        gRect.nXmax = gRect.nXmin+ PIC_WIDTH;           //Change the coordinate by the scrolling
                                                        position
        GCSSetBounds(pPicture,&gRect);                  //Re-specify the picture displaying
                                                        position
        AddRefreshRect(NULL, 0);                        //Re-draw the picture

        return TRUE;
}

long GCPanel00011::GSCROLLBAREX00002OnScroll(unsigned short usMessage, long lLParam,
                                                                    long lUParam)
{
        GBaseObject *pPicture;                          //Picture object
        GBaseObject *pScrollBar;                        //Scroll bar object
        GRect gRect;                                    //Picture rectangle size
        short nScroll;                                  //Current position of scroll

        pPicture = GetChild(GPICTURE00005);             //Acquire the picture object
        pScrollBar = GetChild(GSCROLLBAREX00002);       //Acquire the scroll bar object
        nScroll = GCSGetScrollPosition(pScrollBar);     //Acquire the current position of the
                                                        scroll
        GCSGetBounds(pPicture,&gRect);                  //Acquire the picture displaying position
        gRect.nXmin = -nScroll + PIC_X_POSITION;        //Calculate the X coordinate of the
                                                        picture
        gRect.nXmax = gRect.nXmin+ PIC_WIDTH;
        GCSSetBounds(pPicture,&gRect);                  //Re-specify the picture displaying
                                                        position
        GCSAddRefreshRect(GCSGetScreen( m_pSelf ),NULL, 0);     //Re-draw the picture

        return TRUE;
}
```

18.13 GCEdit (edit control)

GCEdit is a control that displays, inserts or overwrites a character string in the designated rectangle.

State

GCEdit includes the followings.

Definition	Value	Description
GEDIT_EDIT_TYPE_INSERT	0x00	Insert mode
GEDIT_EDIT_TYPE_OVER	0x01	Overwrite mode

Importing Structure

The importing structures of GCEdit are shown below.

```
GDefPropertyClass(GEditProperty, GControlProperty )
        GColor          gcTextColor;        /* Character color                          */
        GBrush          gbBrush;            /* Background color, foreground color and   */
                                            filing pattern
        unsigned short  usBorderID;         /* Solid frame resource ID                  */
        unsigned short  usFontID;           /* Font resource ID                         */
        unsigned char   ucEditMode;         /* Editing mode (insert/overwrite)          */
        unsigned char   ucScrollDispType;   /* Scroll bar display type                  */
        unsigned char   ucScrollBarWidth;   /* Scroll bar width                         */
        GColor          gcScrollBar;        /* Scroll bar color                         */
        GColor          gcScrollBarButton;  /* Scroll bar button color                  */
        GColor          gcScrollBarArrow;   /* Scroll bar arrow color                   */
        unsigned short  usImage1OnID;       /* ON resource ID of design 1               */
        unsigned short  usImage1OffID;      /* OFF resource ID of design 1              */
        unsigned short  usImage2OnID;       /* ON resource ID of design 2               */
        unsigned short  usImage2OffID;      /* OFF resource ID of design 2              */
        unsigned short  usImage3OnID;       /* ON resource ID of design 3               */
        unsigned short  usImage3OffID;      /* OFF resource ID of design 3              */
        unsigned short  usImage4OnID;       /* ON resource ID of design 4               */
        unsigned short  usImage4OffID;      /* OFF resource ID of design 4              */
        GColor          gcScrollBarBorder;  /* Scroll bar border color                  */
        GColor          gcScrollBarBack;    /* Scroll bar background color              */
        unsigned short  usScrollBarBorderID;/* Scroll bar solid frame resource ID       */
        unsigned long   ulTotalSize;        /* Total buffer size                        */
        unsigned long   ulLineSize;         /* Line buffer size                         */
GDefPropertyEnd
```

Parent Class

The parent class of GCEdit is GCControl.

Function List

The functions of GCEdit include the followings.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

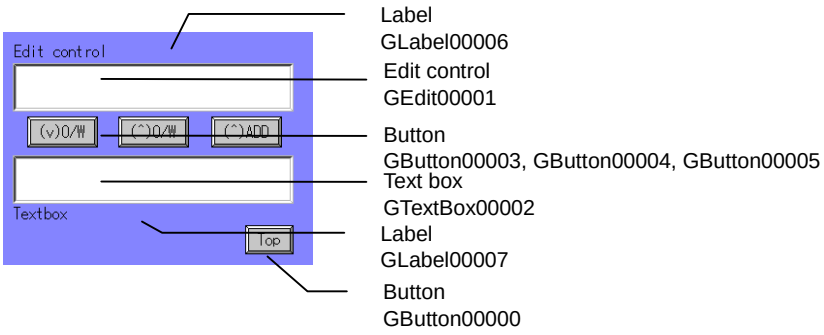
Functions Characteristic to GCProgressBar

Function name	Function
GCSEditSetString	Set display character string
GCSEditGetString	Get display character string
GCSEditAddString	Add character string
GCSEditRemoveString	Delete character string
GCSEditInsertString	Insert character string
GCSEditDeleteStirng	Delete all character strings
GCSEditSetInsertMode	Set insert/overwrite mode
GCSEditGetInsertMode	Get insert/overwrite mode
GCSEditSetLineFeedCode	Set line feed character string
GCSEditGetLineFeedCode	Get line feed character string
GCSEditSetLineTextColor	Set line text display color
GCSEditGetLineTextColor	Get line text display color
GCSEditSetLineBrush	Set line brush
GCSEditGetLineBrush	Get line brush
GCSEditFind	Search forward
GCSEditReverseFind	Search backward
GCSEditReplace	Replace
GCSEditReplaceAll	Replace all
GCSEditSetCursor	Set cursor position
GCSEditGetCursor	Get cursor position
GCSEditInsertLine	Insert line
GCSEditDeleteLine	Delete line
GCSEditGetLineStirng	Get specified line's character string
GCSEditGetLineCount	Get number of lines

Usage Example

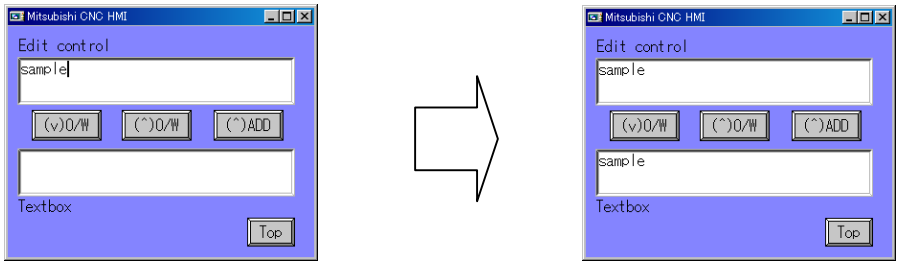
Create the following screen.

Screen Configuration



Action State

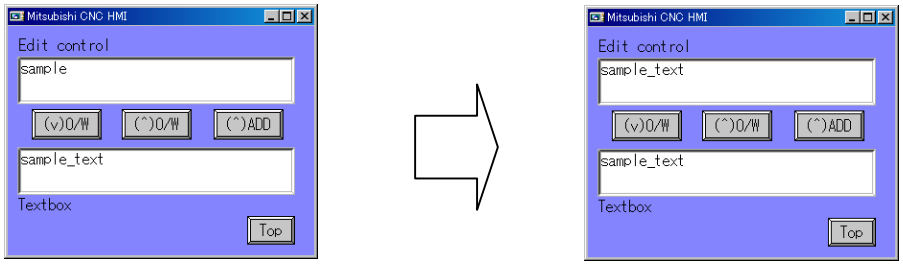
Enter "sample" to the edit control and press the button (character string: "↓Overwrite").



Press the button
(character string: "↓Overwrite").

The characters are overwritten in the text box.

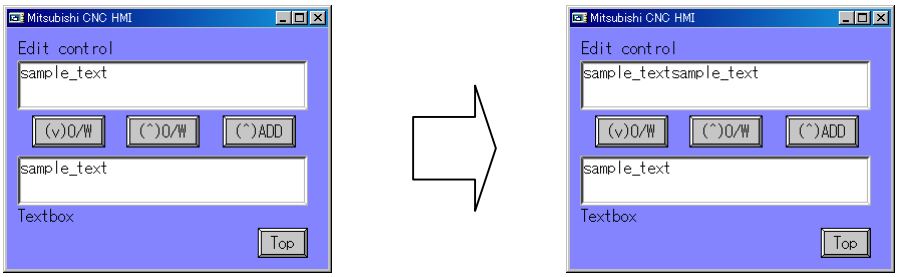
Enter "sample_text" and press the button (character string: "↑Overwrite").



Press the button
(character string: "↑Overwrite").

Characters are overwritten in the edit control.

Enter "sample_text" in the text box and press the button (character string: "↑Add").



Press the button
(character string: "↑Add").

Characters are added in the edit control.

Settings and Action

Arrange the edit control, text box and button on the screen.
After entering characters at the edit control, press the button (character string: "↓Overwrite") to overwrite the text box with the character string entered at the edit control.
After entering characters in the text box, press the button (character string: "↑Overwrite") to overwrite the edit control with the character string entered at the text box. Press the button (character string: "↑Insert") to insert the character string, which is entered at the text box, to the edit control.
Use the program of the "↓Overwrite," "↑Overwrite" and "↑Add" buttons to specify the following action in callback function of the control.

OnClick() : Overwrite or insert a character string when the mouse button is clicked.

When the button (character string: "Top") is pressed, switch to the top page.

NOTE

◆ The character cursor is automatically displayed in the edit control. The following keys are processed inside the edit control.

GK_LEFT (left arrow key)	: Move the character cursor to the left.
GK_RIGHT (right arrow key)	: Move the character cursor to the right.
GK_UP (up arrow key)	: Move the character cursor up.
GK_DOWN (down arrow key)	: Move the character cursor down.
GK_BACKSPACE (BS key)	: Delete one character before the cursor.
GK_DELETE (DEL key)	: Delete one character after the cursor.
GK_RETURN (Enter key)	: Carriage return.

Source Code

```
// "↓  Overwrite" button
long GCPanel00012::GBUTTON00003OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pEdit;			//Edit control object
	GBaseObject *pTextBox;			//Text box object
	GTCHAR szString[257];			//Character string buffer

	pEdit = GetChild( GEDIT00001 );		//Acquire the edit control object
	pTextBox = GetChild( GTEXTBOX00002 );	//Acquire the text object
	GCSGetString( pEdit, szString, 256 );	//Acquire the edit control character string
	GCSSetString( pTextBox, szString );	//Specify the text box character string

	return TRUE;
}
// "↑ Overwrite" button
long GCPanel00012::GBUTTON00004OnClick(unsigned short usMessage, long lLParam, long
lUParam)
{
	GBaseObject *pEdit;			//Edit control object
	GBaseObject *pTextBox;			//Text box object
	GTCHAR szString[257];			//Character string buffer

	pEdit = GetChild( GEDIT00001 );		//Acquire the edit control object
	pTextBox = GetChild( GTEXTBOX00002 );	//Acquire the text object
	GCSGetString( pTextBox, szString, 256 );	//Acquire the text box character string
	GCSSetString( pEdit, szString );	//Specify the edit control character string

	return TRUE;
}
// "↑ Add" button
long GCPanel00012::GBUTTON00005OnClick(unsigned short usMessage, long lLParam, long
lUParam)
{
	GBaseObject *pEdit;			//Edit control object
	GBaseObject *pTextBox;			//Text box object
	GTCHAR    szString[257];		//Character string buffer

	pEdit = GetChild( GEDIT00001 );		//Acquire the edit control object
	pTextBox = GetChild( GTEXTBOX00002 );	//Acquire the text object
	GCSGetString( pTextBox, szString, 256 );	//Acquire the text box character string
	GCSAddString( pEdit, szString );	//Specify the edit control character string

	return TRUE;
}
```

18.14 GCNCControl (NC Control Management)

GCNCControl is an NC control that can be created on the panel. It is the parent class of GNXCounter and GNXCycleTime. It calls callback processes in principle.

Importing Structure

The importing and exporting structures of GCNCControl are shown below.

```
GDefPropertyClass( GNCControlProperty, GControlProperty )
        long            lMachine;       /* NC No.                  */
        long            lSystem;        /* Number of part systems  */
        long            lGround;        /* Ground                  */
        unsigned long   ulAxis;         /* Axis flag               */
GDefPropertyEnd
```

Parent Class

The parent class of GCNCControl is GCControl.

Function List

The functions of GCNCControl are listed below.

Function name	Function
GCSCreate	Create process
GCSDelete	Delete process
GCSOnDraw	Client area drawing process
GCSOnLButtonPress	Client area process for pressing left mouse button
GCSOnLButtonRelease	Client area process for releasing left mouse button
GCSKeyPress	Process for pressing key
GCSKeyRelease	Process for releasing key
GCSSetFocus	Focus setting process
GCSKillFocus	Focus removing process
GCSTimer	Timer process
GCSUser	User process
GCSSetSystemNumber	Part system switching process

18.14.1 GNCDataTextBox (NC Data Textbox)

GNCDataTextBox is a control that displays the internal data of NC (numerical values and character strings) in the designated rectangle and processes entered.

Type

The text displayed inside the control includes the followings.

Definition	Value	Description
GNCTEXT_TYPE_STRING	0x00	Character string
GNCTEXT_TYPE_BIN	0x01	Binary integer value
GNCTEXT_TYPE_DEC	0x02	Signed decimal integer value
GNCTEXT_TYPE_UDEC	0x03	Unsigned decimal integer value
GNCTEXT_TYPE_HEX	0x04	Hexadecimal integer value
GNCTEXT_TYPE_DOUBLE	0x05	Real number (double) value

Shape

The following pieces of information are set in the shape variable of GCBaseWindow.

Definition	Corresponding bit	Description
GNCTEXT_STYLE_COMMA	Bit 16	Indicates presence of the comma. (0: Absent. 1: Present)
GNCTEXT_STYLE_REFUSE_NUM	Bit 17	Indicates whether the value is rejected or not. (0: Accepted. 1: Rejected)
GNCTEXT_STYLE_REFUSE_SML	Bit 18	Indicates whether the single byte lower case letters are rejected or not. (0: Accepted. 1: Rejected)
GNCTEXT_STYLE_REFUSE_CPT	Bit 19	Indicates whether the upper case letters are rejected or not. (0: Accepted. 1: Rejected)
GNCTEXT_STYLE_REFUSE_SYM	Bit 20	Indicates whether the single byte symbols are rejected or not. (0: Accepted. 1: Rejected)
GNCTEXT_STYLE_ZEROSUPPRESS	Bit 25	Indicate whether the zero suppress is displayed or not. (0: Not displayed. 1: Displayed.)
GNCTEXT_STYLE_EXPONENTIAL	Bit 26	Indicate whether the exponential notation is displayed or not. (0: Not displayed. 1: Displayed.)

Data type

The data type of the NC's internal data includes the followings.

Definition	Value	Description
GNCDATA_TYPE_CHAR	0x01	One-byte integer type
GNCDATA_TYPE_SHORT	0x02	Two-byte integer type
GNCDATA_TYPE_LONG	0x03	Four-byte integer type
GNCDATA_TYPE_DOUBLE	0x05	Real number (double) type
GNCDATA_TYPE_STRING	0x10	Character string type

Display

The background of GNCDataTextBox indicates the following states.

Definition	Description
NO_DRAW	The background is not filled.
FILL_BACKCOLOR	Filled in background color

State

The focusing effect of GNCDataTextBox includes the following states.

Definition	Description
GFOCUS_EFFECT_NON	No focusing effect
GFOCUS_EFFECT_CURSOR	Cursor display at focus
GFOCUS_EFFECT_SELECT	All selected at focus

Importing Structure

The importing and exporting structures of GNCDataTextBox are shown below.

```
GDefPropertyClass( GNCDataTextBoxProperty, GControlProperty )
                GCaption            gcCaption;              /* Caption                          */
                unsigned short      usFontID;               /* Font                             */
                unsigned short      usBorderID;             /* Solid frame                      */
                GColor              gcNormalColor;          /* Regular filling color            */
                GColor              gcFocusColor;           /* Focusing filling color           */
                GColor              gcDisableColor;         /* Filling color when disabled      */
                unsigned long       ulLimitLength;          /* Max. number of characters        */
                short               nBackGroundPattern;     /* Background state                 */
                unsigned char       ucType;                 /* Type                             */
                unsigned char       ucFocusEffect;          /* Focusing effect                  */
                long                lRefreshFrequency;      /* Display refresh frequency        */
                long                lRefreshTiming;         /* Display refresh timing           */
                long                lSection;               /* Section No.                      */
                long                lSubSection             /* Sub-section No.                  */
                long                lDataType;              /* Data type                        */
                double              dScale;                 /* Scale                            */
                long                lOffset;                /* Offset                           */
                long                lCross;                 /* Axis cross                       */
                unsigned char       ucIntegerWidth;         /* Number of integer part digits    */
                unsigned char       ucDecimalWidth;         /* Number of decimal part digits    */
GDefPropertyEnd
```

Parent Class

The parent class of GNCDataTextBox is GCNCControl.

Function List

GNCDataTextBox are listed below.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

Common NC Control Function (GCNCControl)

Refer to Section 18.14 GCNCControl (NC control management).

Functions Characteristic to GNCDataTextBox

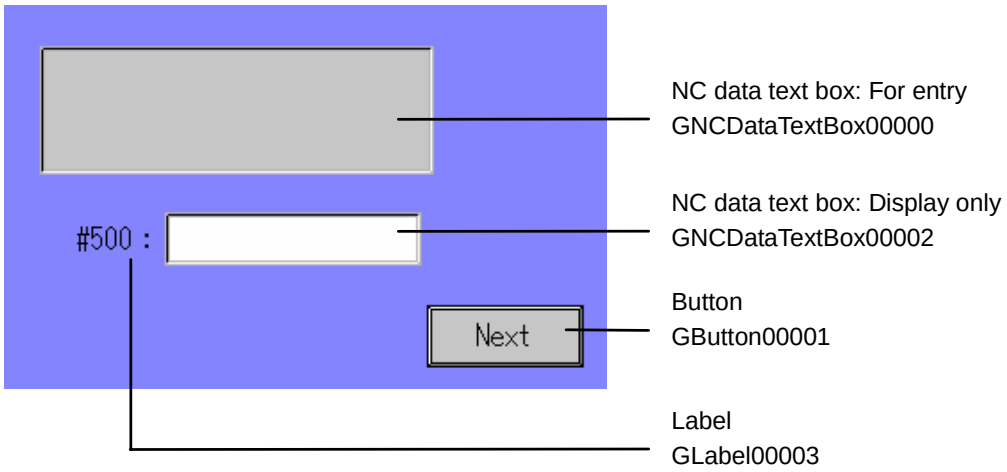
Function name	Function
GCSNCDataTextBoxSetBorderID	Set 3D border resource ID
GCSNCDataTextBoxGetBorderID	Get 3D border resource ID
GCSNCDataTextBoxSetFontID	Set font resource ID
GCSNCDataTextBoxGetFontID	Get font resource ID
GCSNCDataTextBoxSetCaption	Set caption information
GCSNCDataTextBoxGetCaption	Get caption information
GCSNCDataTextBoxSetFocusEffect	Set effect during focus
GCSNCDataTextBoxGetFocusEffect	Get effect during focus
GCSNCDataTextBoxSetFocusColor	Set background color during focus
GCSNCDataTextBoxGetFocusColor	Get background color during focus
GCSNCDataTextBoxSetNormalColor	Set normal background color
GCSNCDataTextBoxGetNormalColor	Get normal background color
GCSNCDataTextBoxSetDisableColor	Set background color when disabled
GCSNCDataTextBoxGetDisableColor	Get background color when disabled
GCSNCDataTextBoxSetTextType	Set character string type
GCSNCDataTextBoxGetTextType	Get character string type
GCSNCDataTextBoxSetCommaStatus	Set display status with commas
GCSNCDataTextBoxGetCommaStatus	Get display status with commas
GCSNCDataTextBoxSetRefuseInputNumberStatus	Set no number input status
GCSNCDataTextBoxGetRefuseInputNumberStatus	Get no number input status
GCSNCDataTextBoxSetRefuseInputSmallLetterStatus	Set one-byte lower case character input disabled status
GCSNCDataTextBoxGetRefuseInputSmallLetterStatus	Get one-byte lower case character input disabled status
GCSNCDataTextBoxSetRefuseInputCapitalLetterStatus	Set one-byte upper case character input disabled status
GCSNCDataTextBoxGetRefuseInputCapitalLetterStatus	Get one-byte upper case character input disable status
CSNCDataTextBoxSetRefuseInputSymbolLetterStatus	Set one-byte symbol input disabled status
GCSNCDataTextBoxGetRefuseInputSymbolLetterStatus	Get one-byte symbol input disabled status
GCSNCDataTextBoxSetZeroSuppressStatus	Set zero suppress display status
GCSNCDataTextBoxGetZeroSuppressStatus	Get zero suppress display status

Function name	Function
GCSNCDataTextBoxSetString	Set display character string
GCSNCDataTextBoxGetString	Get display character string
GCSNCDataTextBoxSetStringBuffer	Set input character string
GCSNCDataTextBoxGetStringBuffer	Get input character string
GCSNCDataTextBoxSetGNCValue	Set display value
GCSNCDataTextBoxGetGNCValue	Get display value
GCSNCDataTextBoxSetBackGroundPattern	Set background fill status
CSNCDataTextBoxGetBackGroundPattern	Get background fill status
GCSNCDataTextBoxSetSelectStringStartPos	Set start position of selected character string
GCSNCDataTextBoxGetSelectStringStartPos	Get start position of selected character string
GCSNCDataTextBoxSetSelectStringEndPos	Set end position of selected character string
GCSNCDataTextBoxGetSelectStringEndPos	Get end position of selected character string
GCSNCDataTextBoxSetIntegerWidth	Set number of integer part digits
GCSNCDataTextBoxGetIntegerWidth	Get number of integer part digits
GCSNCDataTextBoxSetDecimalWidth	Set number of decimal part digits
GCSNCDataTextBoxGetDecimalWidth	Get number of decimal part digits
GCSNCDataTextBoxSetScale	Set scale
GCSNCDataTextBoxGetScale	Get scale
GCSNCDataTextBoxSetOffset	Set offset
GCSNCDataTextBoxGetOffset	Get offset
GCSNCDataTextboxSetSystemNumber	Set part system No.
GCSNCDataTextboxGetSystemNumber	Get part system No.
GCSNCDataTextboxSetGround	Set ground information
GCSNCDataTextboxGetGround	Get ground information
GCSNCDataTextBoxSetAxisInfo	Set axis information
GCSNCDataTextBoxGetAxisInfo	Get axis information
GCSNCDataTextBoxSetSection	Set section No.
GCSNCDataTextBoxGetSection	Get section No.
GCSNCDataTextBoxSetSubSection	Set sub-section No.
GCSNCDataTextBoxGetSubSection	Get sub-section No.
CSNCDataTextBoxSetDataType	Set data type
GCSNCDataTextBoxGetDataType	Get data type

Usage Example

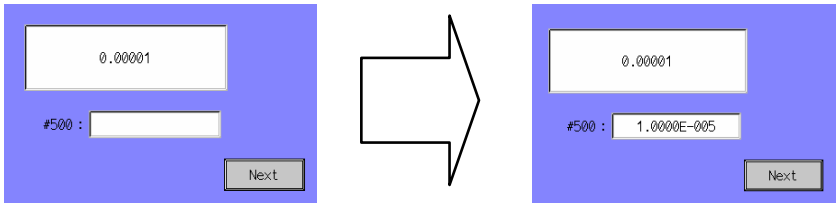
Create the following sample.

Screen Configuration



Action State

Enter "0.00001" and press Enter.



Initial state

After characters are entered

Settings and Action

Create an entry NC data text box (GNCDataTextBox00000) and displaying NC data text box (GNCDataTextBox00002) on the screen.
Prohibit entry for the displaying NC data text box (GNCDataTextBox00002), using the properties setting. Permit character entry for the entry NC data text box (GNCDataTextBox00000) and, each time the ENTER (0x0D) key is pressed during key entry, the confirmed characters are displayed in the displaying NC data text box (GNCDataTextBox00002). Use the program to specify the following actions in callback functions of the control.
OnKeyPress() : Get the entered character string and display it in the NC data text box for display.
When the button (character string: "Next") is pressed, the display is switched to the next page.

Source Code

```
long GCPanel00000::GNCDATATEXTBOX00000OnKeyPress(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pNCTextBox;                          //NC data text box object
	GTCHAR szNCDataText[128];                         //Character string buffer

	if(lLParam == GK_ENTER){                          // Acquire characters only when ENTER is input
		pNCTextBox =                                  // Acquire the NC data text box object
		GetChild(GNCDATATEXTBOX00000);
		GCSNCDataTextBoxGetString(pNCTextBox,         // Acquire entered characters
			szNCDataText, sizeof(szNCDataText);
		pNCTextBox =                                  // Acquire the NC data text box object
		GetChild(GNCDATATEXTBOX00002);
		GCSNCDataTextBoxSetString(pNCTextBox,         // Display the entered characters
			szNCDataText);
	}

	return TRUE;
}
```

18.14.2 GCNXMenu (Menu Display)

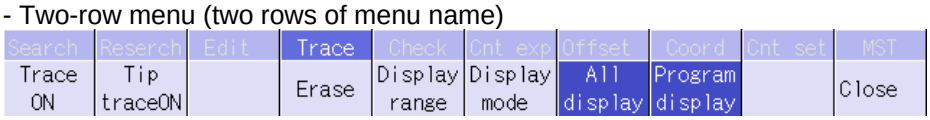
GCNXMenu is a control for displaying the names of menu buttons (10 buttons) and their icons.

Type

The display of GCNXMenu includes the followings.

- One-row menu (icon + menu name)

- Two-row menu (two rows of menu name)



State

There are three types of menu state for each menu button: menu ON (highlighted), menu OFF (normal state) and menu disabled.

Importing Structure

The importing and exporting structures of GCNXMenu are shown below.

```
GDefPropertyClass( GNXMenuProperty, GNCControlProperty )
        unsigned short      usMenuType;             /* Menu type                                    */
        GColor              gcForeColor;            /* Normal character string color on menu        */
        GColor              gcPushedColor;          /* Selected character string color on menu      */
        GColor              gcForeBackColor;        /* Normal background color
                                                                        for one-row menu             */
        GColor              gcPushedBackColor;      /* Selected background color
                                                                        for one-row menu             */
        GColor              gcUpperForeBackColor;   /* Normal background color
                                                                    for upper row of two-row menu    */
        GColor              gcUpperPushedBackColor; /* Selected background color
                                                                    for upper row of two-row menu    */
        GColor              gcLowerForeBackColor    /* Normal background color
                                                                    for lower row of two-row menu    */
        GColor              gcUpperPushedBackColor; /* Selected background color
                                                                    for lower row of two-row menu    */
GDefPropertyEnd
```

Parent Class

The parent class of GCNXMenu is GCNCControl.

Function List

The functions of GCNXMenu are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

NCCommon Control Function（GCNCControl）

Refer to Section 18.14 GCNCControl (NC control management).

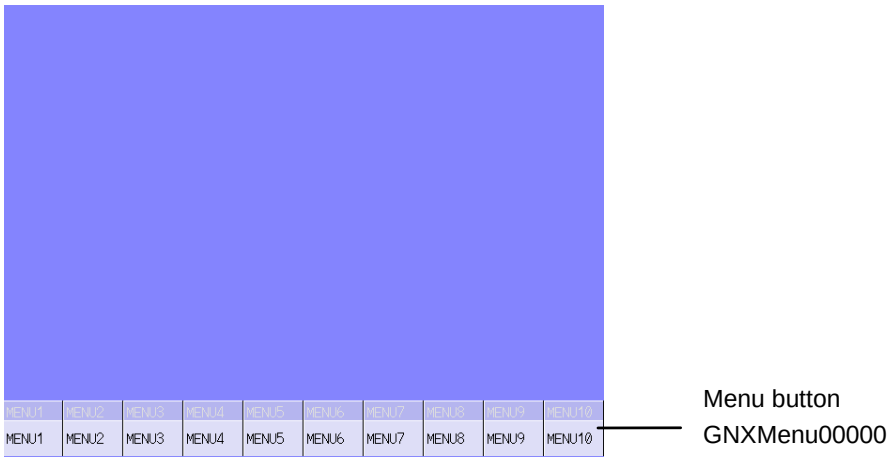
Functions Characteristic to GCNXMenu

Function name	Function
GCSRefresh	Redraw menu.
GCSSetRefreshValidFlag	Switch enabled/disabled state of menu display
GCSSetMenuButtonName_all	Set menu name for one-row menu (10 menus)
GCSSetMenuButtonName_one	Set menu name for one-row menu (one menu)
GCSSetMenuButtonOffIcon	Set icon for one-row menu (10 menus) at OFF
GCSSetMenuButtonOffIcon_one	Set icon for one-row menu (one menu) at OFF
GCSSetMenuButtonOnIcon	Set icon for one-row menu (10 menus) at ON
GCSSetMenuButtonOnIcon_one	Set icon for one-row menu (one menu) at ON
GCSSetMenuButtonUpperName_all	Set menu name for upper row of two-row menu (10 menus)
GCSSetMenuButtonUpperName_one	Set menu name for upper row of two-row menu (one menu)
GCSSetMenuButtonLowerName_all	Set menu name for lower row of two-row menu (10 menus)
GCSSetMenuButtonLowerName_one	Set menu name for lower row of two-row menu (one menu)
GCSSetMenuButtonState_all	Set menu state for one-row menu (10 menus)
GCSSetMenuButtonState_one	Set menu state for one-row menu (one menu)
GCSGetMenuButtonState_all	Get menu state for one-row menu (10 menus)
GCSGetMenuButtonState_one	Get menu state for one-row menu (one menu)
GCSSetMenuButtonUpperState_all	Set menu state for upper row of two-row menu (10 menus)
GCSSetMenuButtonUpperState_one	Set menu state for upper row of two-row menu (one menu)
GCSGetMenuButtonUpperState_all	Get menu state for upper row of two-row menu (10 menus)
GCSGetMenuButtonUpperState_one	Get menu state for upper row of two-row menu (one menu)
GCSSetMenuButtonLowerState_all	Set menu state for lower row of two-row menu (10 menus)
GCSSetMenuButtonLowerState_one	Set menu state for lower row of two-row menu (one menu)
GCSGetMenuButtonLowerState_all	Get menu state for lower row of two-row menu (10 menus)
GCSGetMenuButtonLowerState_one	Get menu state for lower row of two-row menu (one menu)
GCSSendProcessID	Send process ID to panel

Usage Example

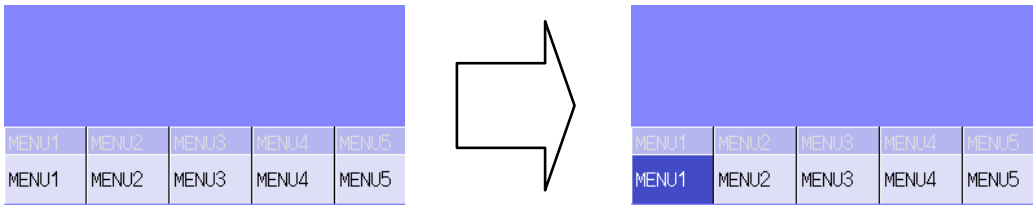
Create the following sample.

Screen Configuration



Action State

The menu state changes when the menu key is pressed.



Settings and Action

Create a menu button (GNXMenu00000) on the screen. Select "Permission" for InputPermission, using the properties setting, and select 2StepMenu (two-row menu) for MenuType. Use the program to specify the following actions in callback functions of the control.

OnCreate() : Display a character string on a menu button.
OnKeyPress() : Switch the menu key selection state using a menu key and display the state.
OnPress() : At a left-click of the mouse, the selection state of the menu button is changed and displayed.

By the menu operation, the ON/OFF state of the menu button is changed.

NOTE

◆ When a menu button (GNXMenu00000) is created (OnCreate()), all the menu buttons are in OFF state.

Source Code

```
long GCPanel00000::GNXMENU00000OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pGCNXMenu;                                    //Menu object
	char *psMenu[10];

	pGCNXMenu = GetChild( GNXMENU00000 );                      //Get menu object

	psMenu[0] = (char*)&L"MENU1";
	psMenu[1] = (char*)&L"MENU2";
	psMenu[2] = (char*)&L"MENU3";
	psMenu[3] = (char*)&L"MENU4";
	psMenu[4] = (char*)&L"MENU5";
	psMenu[5] = (char*)&L"MENU6";
	psMenu[6] = (char*)&L"MENU7";
	psMenu[7] = (char*)&L"MENU8";
	psMenu[8] = (char*)&L"MENU9";
	psMenu[9] = (char*)&L"MENU10";

	//Set character string to display on menu
	GCSSetMenuButtonUpperName_all(pGCNXMenu, 1, psMene);
	GCSSetMenuButtonLowerName_all(pGCNXMenu, 1, psMenu);

	return TRUE;
}

long GCPanel00000::GNXMENU00000OnKeyPress(unsigned short usMessage, long lLParam, long lUParam)
{
	GbaseObject *pGCNXMenu;
	long lMenuState;                                           //Menu selection state
	short nMenuNo;                                             //Menu No.

	pGCNXMenu = GetChild( GNXMENU00000 );
	nMenuNo = (short)(lLParam - GK_F1) + 1;                    //Calculate menu No.
	lMenuState =                                               //Get menu state
		GCSGetMenuButtonLowerState_one(pGCNXMenu, 1,
                                                   nMenuNo);

	if(lMenuState == 0){
	    GCSSetMenuButtonLowerState_one(pGCNXMenu, 1,           //Switch to menu ON state
                                                   nMenuNo, 1);
	}else{
	    GCSSetMenuButtonLowerState_one(pGCNXMenu, 1,           //Switch to menu OFF state
                                                   nMenuNo, 0);
	}

	return TRUE;
}
```

```
GRect grPosTable[10] = {
    {    2, 0,   61, 0 },                                    //F1 menu
    {   66, 0, 125, 0 },                                     //F2 menu
    { 130, 0, 189, 0 },                                      //F3 menu
    { 194, 0, 253, 0 },                                      //F4 menu
    { 258, 0, 317, 0 },                                      //F5 menu
    { 322, 0, 381, 0 },                                      //F6 menu
    { 386, 0, 445, 0 },                                      //F7 menu
    { 450, 0, 509, 0 },                                      //F8 menu
    { 514, 0, 573, 0 },                                      //F9 menu
    { 578, 0, 637, 0 }                                       //F10 menu
};

long GCPanel00000::GNXMENU00000OnPress(unsigned short usMessage, long lLParam, long lUParam)
{
        GBaseObject *pGCNXMenu;
        long lMenuState;                                     //Menu selection state
        short nMenuNo;                                       //Menu No.
        short nX;                                            //X coordinate

        pGCNXMenu = GetChild( GNXMENU00000 );                //Get menu object

        nX = (short)(lUParam & 0xffff);                      //Get X coordinate

        for(short idx = 0; idx < 10; idx++){
            if((grPosTable[idx].nXmin <= nX)                 //Check X coordinate range
            && (grPosTable[idx].nXmax >= nX)){
                nMenuNo = idx + 1;                           //Calculate menu No.
                lMenuState =                                 //Get menu state
                    GCSGetMenuButtonLowerState_one(pGCNXMenu, 1, nMenuNo);

                if(lMenuState == 0){
                    GCSSetMenuButtonLowerState_one(pGCNXMen  //Switch to menu ON state
                    u, 1,
                                                        nMenuNo, 1);
                }else{
                    GCSSetMenuButtonLowerState_one(pGCNXMen  //Switch to menu OFF state
                    u, 1,
                                                        nMenuNo, 0);
                }
                break;
            }
        }

        return TRUE;
}
```

18.14.3 GNXFileInOut (Input/Output Control)

GNXFileInOut is a control for inputting and outputting NC data between the NC memory and an external device. The hard disk built in the NC unit is also handled as an external device.

Parent Class

The parent class of GNXFileInOut is GCNCControl.

Function List

The functions of GNXFileInOut are listed below.

Common Control Function (GCBaseWindow)

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function (GCControl)

Refer to Section 18.1 GCControl (control management).

Common NC Control Function (GCNCControl)

Refer to Section 18.14 GCNCControl (NC control management).

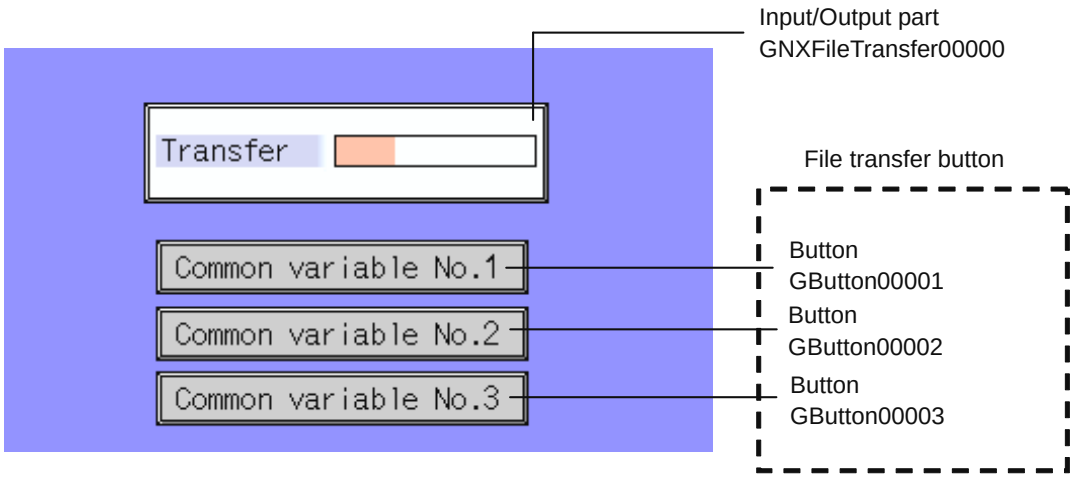
Functions Characteristic to GNXFileInOut

Function name	Function
GCSIOCheckUpDateFile	Check the existence of file
GCSIOFileTransfer	Start file transfer
GCSIODeleteFile	Delete file/directory
GCSIOCreateDirectory	Create directory
GCSIORenameFile	Rename file/directory
GCSInterrupt	Interrupt file transfer
GCSIOinitializeMessage	Initialize file input/output part
GCSGetLastError	Get error information
GCSIOGetTransferStatus	Get transfer status

Usage Example

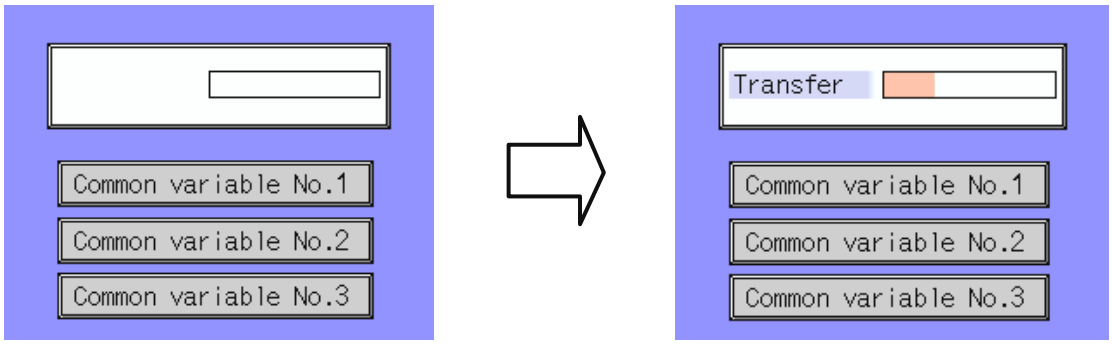
Create the following sample.

Screen Configuration



Action State

When the button "Common variable No. 1" is clicked, the common variable data stored in an external input/output device is transferred to the internal memory of NC.



Initial state

After clicked

Settings and Action

Align an input/output part (GNXDataTransfer00000) and three types of file transfer buttons (GButton00001, GButton00002 and GButton00003) on the screen. When the file transfer button is clicked, the file specified in advance is transferred.

OnClick(GButton00001) :Start transferring the common variable data 1 (COMMONDATA1.VAR) prepared in the external device (D:/COMMON/).

OnClick(GButton00001):Start transferring the common variable data 2 (COMMONDATA2.VAR) prepared in the external device (D:/COMMON/).

OnClick(GButton00001) :Start transferring the common variable data 3 (COMMONDATA3.VAR) prepared in the externaldevice (D:/COMMON/).

NOTE

- How to specify the file to transfer
  Give the information of file to transfer to the argument of the GCS function by full path.
- The following character strings have to be set in the character string resource in advance.
  ID_STRING00060 ..... M01:/DAT/COMMON.VAR
  ID_STRING00061 ..... D:/COMMON/COMMONDATA1.VAR
  ID_STRING00062 ..... D:/COMMON/COMMONDATA2.VAR
  ID_STRING00063 ..... D:/COMMON/COMMONDATA3.VAR

Source Code

```
// Transfer the file (Common variable data 1) prepared in an external device to the NC memory.
long GCPanel00000::GBUTTON00001OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
        GBaseObject *pInOut;                                    // Input/Output part object
        pInOut = GetChild(GNXFILEINOUT00000);                   // Get input/output part object
        long lResult;                                           // Process result

        // Start file transfer

        lResult = GCSIOFileTransfer(pInOut, GRCLoadString( ID_STRING00061 ),
                                                                GRCLoadString( ID_STRING00060 ));


        return TRUE;

}

   // Transfer the file (Common variable data 2) prepared in an external device to the NC memory.
long GCPanel00000::GBUTTON00002OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
        GBaseObject *pInOut;                                    // Input/Output part object
        pInOut = GetChild(GNXFILEINOUT00000);                   // Get input/output part object
        long lResult;                                           // Process result

        // Start file transfer

        lResult = GCSIOFileTransfer(pInOut, GRCLoadString( ID_STRING00062 ),
                                                                GRCLoadString( ID_STRING00060 ));


        return TRUE;

}
```

```
// Transfer the file (Common variable data 3) prepared in an external device to the NC memory.
long GCPanel00000::GBUTTON00003OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
	GBaseObject *pInOut;                                    // Input/Output part object
	pInOut = GetChild(GNXFILEINOUT00000);                   // Get input/output part object
	long lResult;                                           // Process result

	// Start file transfer

	lResult = GCSIOFileTransfer(pInOut, GRCLoadString( ID_STRING00063 ),
                                                                GRCLoadString( ID_STRING00060 ));


	return TRUE;

}
```

18.14.4 GNXAlarmMessage (Alarm Display)

GNXAlarmMessage is a control for displaying the alarm No. and alarm message text when an alarm occurs.

Parent Class

The parent class of GNXAlarmMessage is GCNCControl.

Function List

The functions of GNXAlarmMessage are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

Common NC Control Function（GCNCControl）

Refer to Section 18.14 GCNCControl (NC control management).

Functions Characteristic to GNXAlarmMessage

Function name	Function
GCSSetDispMessageType	Set message display form

18.14.5 GNXMonitorStatus (Operation Status Display)

GNXMonitorStatus is a control for displaying the NC operation status.

Parent Class

The parent class of GNXMonitorStatus is GCNCControl.

Function List

The functions of GNXMonitorStatus are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window management).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

Common NC Control Function（GCNCControl）

Refer to Section 18.14 GCNCControl (NC control management).

Functions Characteristic to GNXMonitorStatus

Function name	Function
GCSMonStatusSetDispSysNumber	Set the part system No. to display
GCSMonStatusGetDispSysNumber	Get the part system No. being displayed

18.14.6 GNXTime (Time Display)

GNXTime is a control for displaying the time.

Parent Class

The parent class of GNXTime is GCNCControl.

Function List

The functions of GNXTime are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window management).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

NCCommon Control Function（GCNCControl）

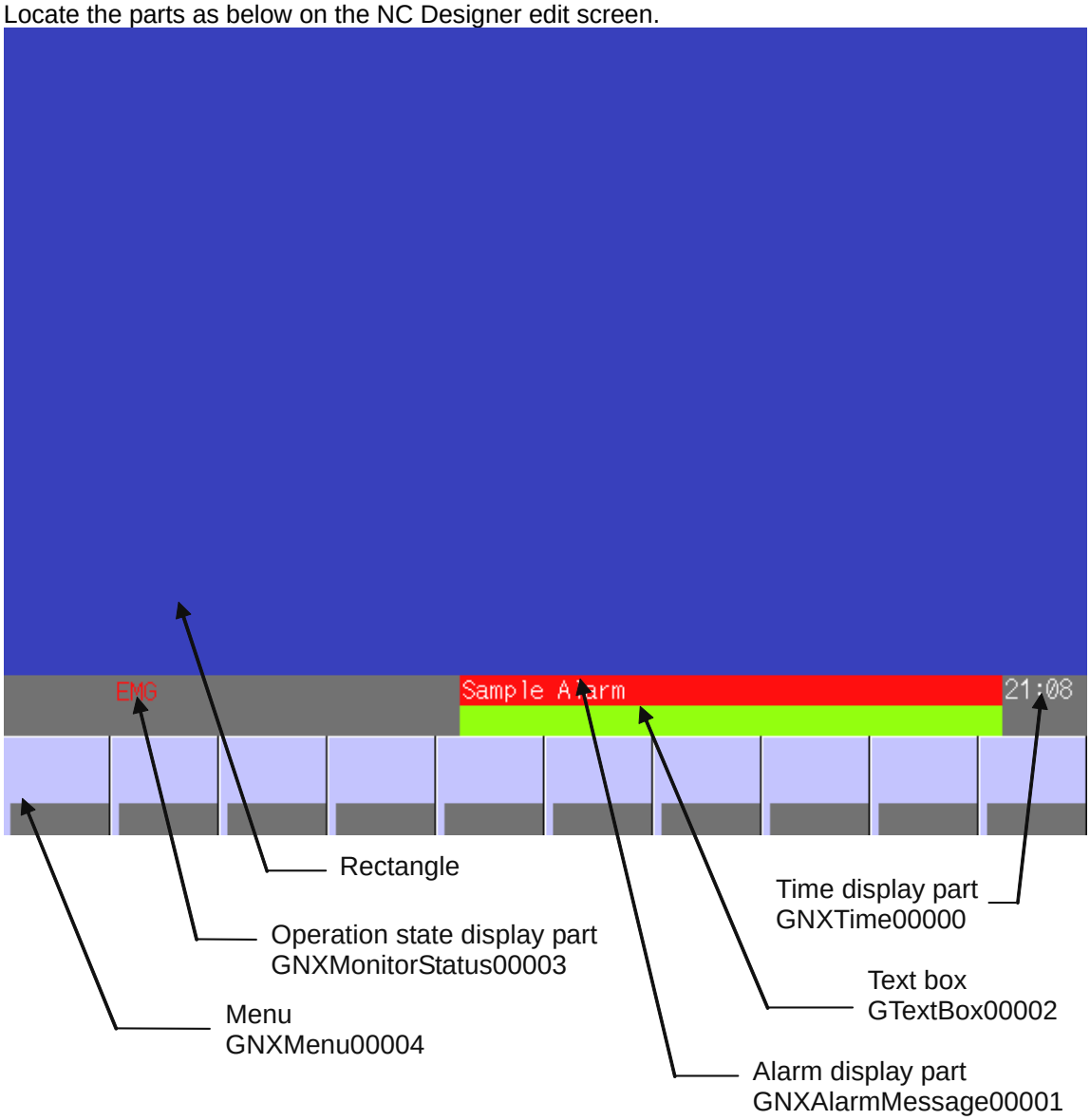
Refer to Section 18.14 GCNCControl (NC control management).

Usage Example

Create the following sample.

Screen Configuration

Locate the parts as below on the NC Designer edit screen.



Property Settings

Set the properties of the located parts. Below are the example settings.

1	GNXTime00000	Time display part
X		590
Y		385
WIDTH		45
HEIGHT		19
ForeColor		RGU (230,230,230)
BackColor		RGU (98, 98, 98)
FrameVisible		False
FontType		Normal

2	GNXAlarmMessage00001	Alarm display part
X		270
Y		385
WIDTH		320
HEIGHT		19
BackColor		RGU (98, 98, 98)
FontType		Normal
DisplayMessage		0

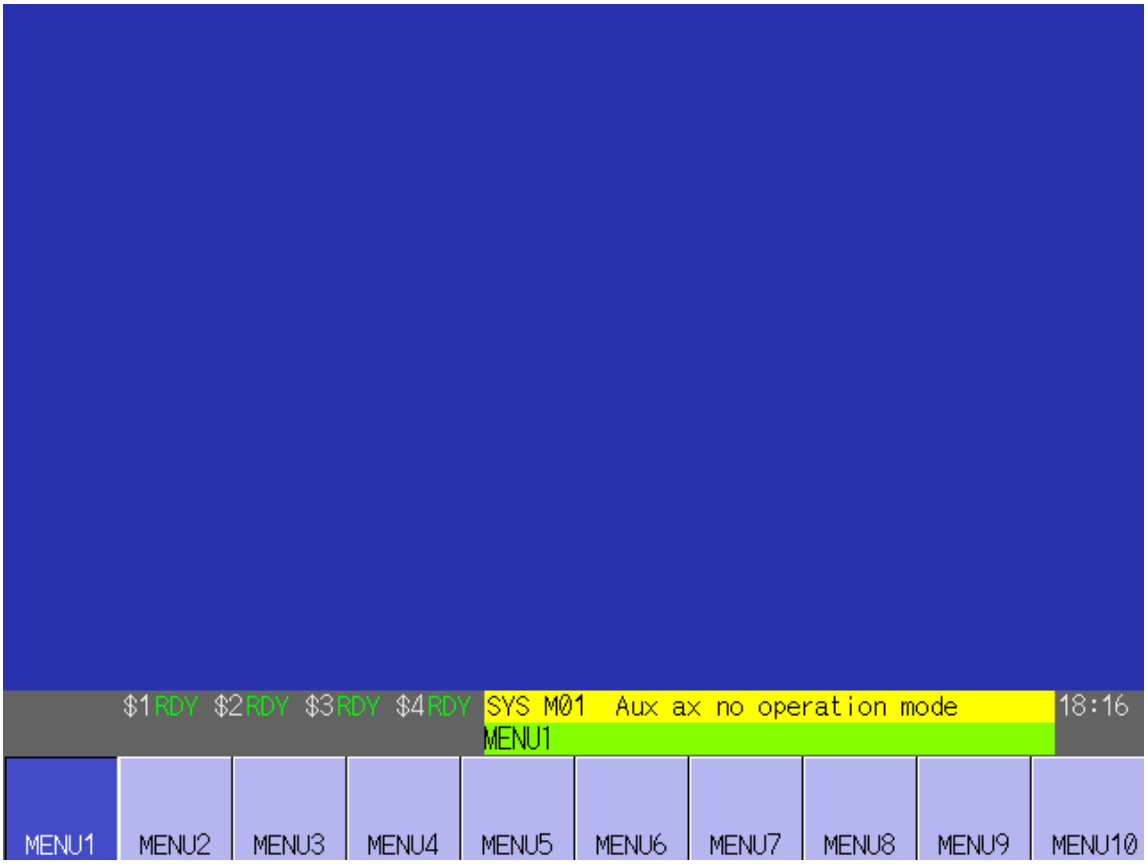
3	GTextBox00002	Text box
X		270
Y		403
WIDTH		320
HEIGHT		18
Background color		RGU (98, 98, 98)
Background color at the time of disable		RGU (98, 98, 98)
Character color		RGU (00, 00, 00)
Existence of solid frame		None
Input permission		Prohibition

4	GNXMonitorStatus00003	Operation state display part
X		66
Y		385
WIDTH		206
HEIGHT		19
BackColor		RGU (98, 98, 98)
FrameVisible		False
FontType		Normal
SystemNumber		0
SystemNameColor		RGU (230, 230, 230)
RefreshFrequency		4
RefreshTiming		3

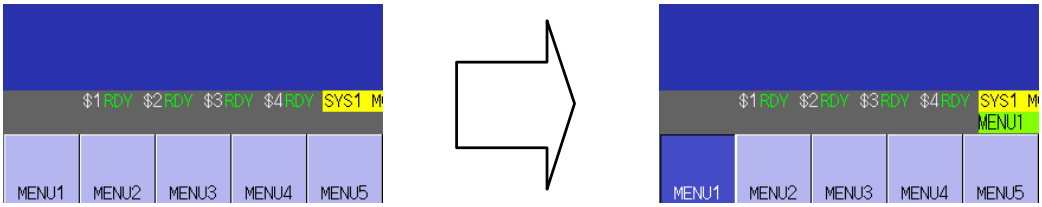
5	GNXMenu00004	Menu
X		1
Y		422
WIDTH		648
HEIGHT		59
MenuType		1StepMenu(VGA)

Action State

When the custom release function is applied, the following screen is displayed. This screen shows the operation status according to the NC type.
While an alarm is occurring, the alarm message is displayed on the alarm display part.



The menu state changes when the menu key is pressed or clicked.
When the menu display is highlighted, the menu name is displayed on the text box.



Initial state

After menu key is entered

NOTE

- When no message is displayed, the backgrounds of the alarm display part (GNXAlarmMessage00001) and the text box (GTextBox00002) are displayed in the same color as of the panel background.

Source Code

```
long GCPanel00000:: GCPanel00000::Init()
{
	GBaseObject *pGCNXMenu =                                    //Get menu object
	GetChild( GNXMENU00004 );                                   //Get text object
	GBaseObject *pText = GetChild( GTEXTBOX00002 );
	char *psMenu[10];
	static wchar_t sMenuString[10][7];
	GTCHAR _gtText[20];

	for(short i = 0; i < 10; i++)                          //Set character string to display on menu
	{
	    swprintf( sMenuString[i], TEXT(" MENU%d"), (i + 1) );
	    psMenu[i] = (char*)sMenuString[i];
	}
	GCSSetMenuButtonName_all(pGCNXMenu, 1, psMenu);          //Set name to menu

	Long _lStsMain[10] = {0,0,0,0,0,0,0,0,0,0};
	GCSSetMenuButtonState_all(pGCNXMenu, 1, _lStsMain);      //Initialize menu state
	_sgtprintf(_gtText, TEXT(""));
	GCSSetString(pText, _gtText);                            //Initialize text box
	GCSTextboxSetDisableColor(pText, 0x626262);    //Set the same color as of panel background

	return TRUE;
}

long GCPanel00000::GNXMENU00000OnKeyPress(unsigned short usMessage, long lLParam, long
lUParam)
{
	GbaseObject *pGCNXMenu = GetChild( GNXMENU00004 );
	GBaseObject *pText = GetChild( GTEXTBOX00002 );
	GTCHAR _gtText[20];
	long lMenuState;                                         //Menu selection state
	short nMenuNo;                                           //Menu No.

	nMenuNo = (short)(lLParam - GK_F1) + 1;                  //Calculate menu No.
	lMenuState =                                             //Get menu state
	    GCSGetMenuButtonState_one(pGCNXMenu, 1, nMenuNo);

	if(lMenuState == 0){
	    long _lStsMain[10] = {0,0,0,0,0,0,0,0,0,0};
	    GCSSetMenuButtonState_all(pGCNXMenu, 1, _lStsMain);  //Initialize menu state
	     GCSSetMenuButtonState_one(pGCNXMenu,          1,    //Switch to menu ON state
	     nMenuNo, 1);
	     _sgtprintf(_gtText, TEXT("MENU%d"), nMenuNo);       //Display menu name
	     GCSSetString(pText, _gtText);
	     GCSTextboxSetDisableColor(pText,          //Set background color to yellow-green
	                                0x80FF00);
	}else{
	     GCSSetMenuButtonState_one(pGCNXMenu, 1,             //Switch to menu OFF state
	                                          nMenuNo, 0);
	     _sgtprintf(_gtText, TEXT(""));
	GCSSetString(pText, _gtText);                            //Initialize text box
	GCSTextboxSetDisableColor(pText, 0x626262);              //Set panel background color
	}

	return TRUE;

}
```

```
GRect grPosTable[10] = {
    {    2, 0,   61, 0 },                                    //F1 menu
    {   66, 0, 125, 0 },                                     //F2 menu
    { 130, 0, 189, 0 },                                      //F3 menu
    { 194, 0, 253, 0 },                                      //F4 menu
    { 258, 0, 317, 0 },                                      //F5 menu
    { 322, 0, 381, 0 },                                      //F6 menu
    { 386, 0, 445, 0 },                                      //F7 menu
    { 450, 0, 509, 0 },                                      //F8 menu
    { 514, 0, 573, 0 },                                      //F9 menu
    { 578, 0, 637, 0 }                                       //F10 menu
};

long GCPanel00000::GNXMENU00000OnPress(unsigned short usMessage, long lLParam, long lUParam)
{
        GbaseObject *pGCNXMenu = GetChild( GNXMENU00004 );
        GBaseObject *pText = GetChild( GTEXTBOX00002 );
        GTCHAR _gtText[20];
        long lMenuState;                                     //Menu selection state
        short nMenuNo;                                       //Menu No.
        short nX;                                            //X coordinate

        nX = (short)(lUParam & 0xffff);                      //Get X coordinate

        for(short idx = 0; idx < 10; idx++){
            if((grPosTable[idx].nXmin <= nX)                 //Check X coordinate range
            && (grPosTable[idx].nXmax >= nX)){
                nMenuNo = idx + 1;                           //Calculate menu No.
                lMenuState =                                 //Get menu state
                    GCSGetMenuButtonState_one(pGCNXMenu, 1, nMenuNo);

                if(lMenuState == 0){
                    long _lStsMain[10] = {0,0,0,0,0,0,0,0,0,0};
                    GCSSetMenuButtonState_all(pGCNXMenu, 1, _lStsMain);   //Initialize menu state
                    GCSSetMenuButtonState_one(pGCNXMenu, 1,      //Switch to menu ON state
                                                        nMenuNo, 1);
                    _sgtprintf(_gtText, TEXT("MENU%d"), nMenuNo);
                    GCSSetString(pText, _gtText);            //Display menu name
                    GCSTextboxSetDisableColor(pText,      //Set background color to yellow-green
                                                    0x80FF00);
                }else{
                    GCSSetMenuButtonState_one(pGCNXMenu, 1,      //Switch to menu OFF state
                                                        nMenuNo, 0);
                    _sgtprintf(_gtText, TEXT(""));
                    GCSSetString(pText, _gtText);            //Initialize text box
                    GCSTextboxSetDisableColor(pText, 0x626262);      //Set panel background color
                }
                break;
            }
        }

        return TRUE;
}
```

18.15 GCInputBox (Input box)

GCInputBox is a control that has the same function as the existing text box but differs in the following points.
- Operation function
- The input values can be reflected to another specified control.

Display

The background of GCInputBox includes the following states.

Definition	**Description**
NO_DRAW	The background is not filled.
FILL_BACKCOLOR	Filled in background color.

State

The focusing effect of GCInputBox includes the following states.

Definition	**Description**
GFOCUS_EFFECT_NON	No focusing effect
GFOCUS_EFFECT_CURSOR	Cursor display at focus
GFOCUS_EFFECT_SELECT	All selected at focus

The cursor type of GCInputBox includes the following states.

Definition	**Description**
GCURSOR_TYPE_NORMAL	Vertical line
GCURSOR_TYPE_RECT	Block

Input method

The input method of GCInputBox includes the following states.

Definition	**Description**
GCINPUTBOX_INPUTTYPE_ABS	Absolute input
GCINPUTBOX_INPUTTYPE_INC	Incremental input

Input mode

The input mode of GCInputBox includes the following states.

Definition	**Description**
GCINPUTBOX_INPUTMODE_INSERT	Insert mode
GCINPUTBOX_INPUTMODE_OVERWRITE	Overwrite mode

Importing Structure

The importing and exporting structures of GCInputBox are shown below.

```
GDefPropertyClass( GCInputBoxProperty, GControlProperty )
        unsigned short      usFontID;               /* Font                          */
        unsigned short      usBorderID;             /* Solid frame                   */
        GColor              gcNormalBackColor;      /* Regular filling color         */
        GColor              gcNormalForeColor;      /* Regular character color       */
        GColor              gcFocusBackColor;       /* Focusing filling color        */
        GColor              gcFocusForeColor;       /* Focusing character color      */
        GColor              gcDisableBackColor;     /* Filling color when disabled   */
        unsigned long       ulLimitLength;          /* Max. number of characters     */
        short               nBackGroundPattern;     /* With or without filling color */
        unsigned char       ucFocusEffect;          /* Focusing effect               */
        unsigned char       ucCursorType;           /* Cursor type                   */
        GColor              gCursorColor;           /* Cursor color                  */
        unsigned char       ucCalcFlag;             /* Operation function            */
        unsigned char       ucEchobackFlag;         /* Echo back                     */
        unsigned char       ucAbsIncType;           /* Abs/Inc input method          */
        unsigned char       ucReserve;              /* Reserve(1)                    */
        GColor              gcSubCursorBackColor    /* Sub cursor filling color      */
        GColor              gcSubCursorForeColor    /* Sub cursor character color    */
        GCaption            gcCaption               /* Caption                       */
        unsigned short      usReserve;              /* Reserve(2)                    */
        long                lReserve;               /* Reserve(3)                    */
        char                szReserve[12]           /* Reserve(4)                    */
        unsigned short      usReserve2;             /* Reserve(5)                    */
        long                lReserve2;              /* Reserve(6)                    */
        char                szReserve2[2];          /* Reserve(7)                    */
GDefPropertyEnd
```

Parent Class

The parent class of GCInputBox is GControl.

Function List

Functions of GCInputBox are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

Functions Characteristic to GCInputBox

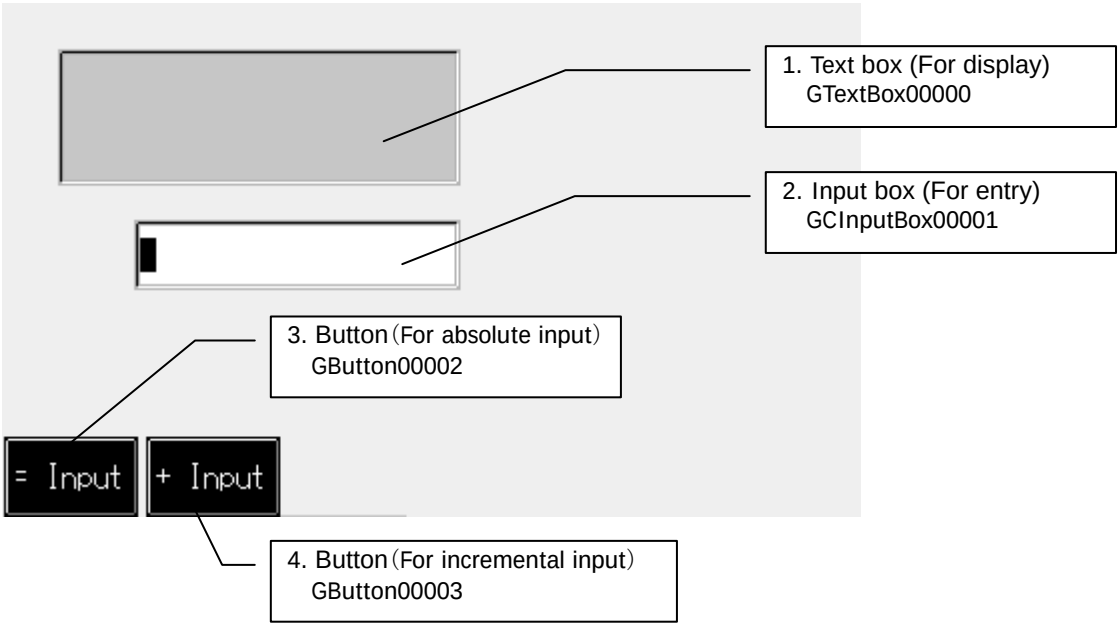
Method name	Method Description
GCSInputBoxSetPasswordStatus	Set password display status
GCSInputBoxGetPasswordStatus	Get password display status
GCSInputBoxSetString	Set display character string
GCSInputBoxGetString	Get display character string
GCSInputBoxSetGValue	Set display value
GCSInputBoxGetGValue	Get display value
GCSInputBoxSetFocusEffect	Set effect during focus
GCSInputBoxGetFocusEffect	Get effect during focus
GCSInputBoxSetBackGroundPattern	Set background fill status
GCSInputBoxGetBackGroundPattern	Get background fill status
GCSInputBoxSetSelectStringStartPos	Set start position of selected character string
GCSInputBoxGetSelectStringStartPos	Get start position of selected character string
GCSInputBoxSetSelectStringEndPos	Set end position of selected character string
GCSInputBoxGetSelectStringEndPos	Get end position of selected character string
GCSInputBoxSetCalcFlag	Set operation function (enable/disable)
GCSInputBoxGetCalcFlag	Get operation function (enable/disable)
GCSInputBoxSetInputValue	Set the control to reflect the INPUT
GCSInputBoxSetEchoback	Set echo back (enable/disable)
GCSInputBoxGetEchoback	Get echo back (enable/disable)
GCSInputBoxSetAbsIncType	Set input method (absolute/incremental)
GCSInputBoxGetAbsIncType	Get input method (absolute/incremental)
GCSInputBoxSetRefrectControl	Set the control to reflect the INPUT
GCSInputBoxGetRefrectControlTextType	Set character string type of control to reflect the INPUT (character string/numerical value/real number)
GCSInputBoxGetInputMode	Get input mode (insert/overwrite)
GSCInputBoxGetSubCursorControl	Get the ID of the sub cursor display control

Usage Example

Create the following sample.

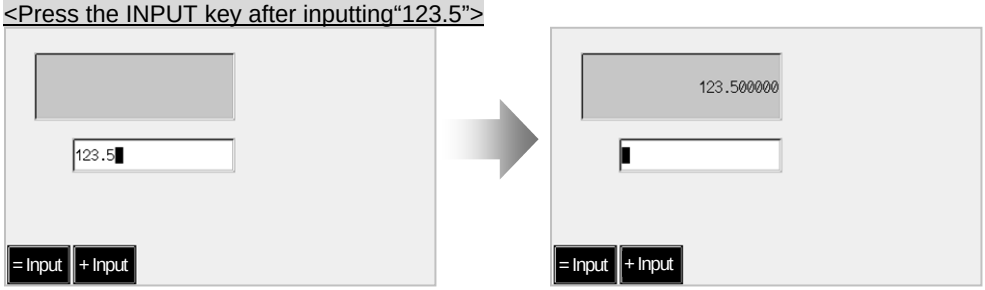
Screen Configuration

Set the value in the input box to the text box by the INPUT key.
Press [=Input] button to directly set the value in the input box to the text box (absolute). And press [+Input] button to add the value in the input box to the value in the text box.

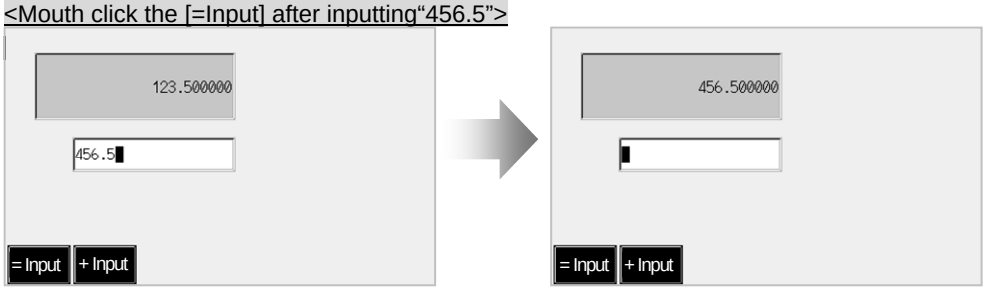


Action State

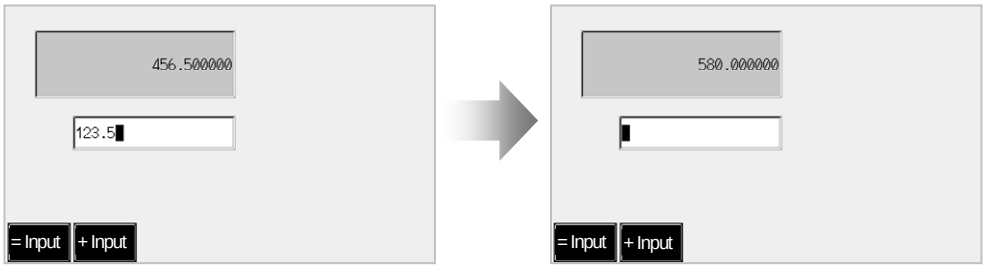
1) The focus is placed on the input box when the sample screen opens.
2) Press the INPUT key after entering data keys (numerical values) to set the input value to the text box.
<Press the INPUT key after inputting“123.5”>



3) Mouth-click the [=Input] button after entering data keys (numerical values) to directly set the input value to the text box. After the setting, the focus will be placed on the input box.
<Mouth click the [=Input] after inputting“456.5”>



4) Mouth-click the [+Input] button after entering data keys (numerical values) to add the value in the input box to the value in the text box and display the added value in the text box. After the setting, the focus will be placed on the input box.
<Mouth click the [+Input] after inputting“123.5”>



Property settings

Set the properties of the located parts.

GInputBox00001	**Input box**
Effect during focus	With cursor
Cursor type	Block
Horizontal position	Align left
Abs/Inc	Abs

GTextBox00000	**Text box**
Type	Character sequence
Existence of caption	Yes
Display format	%f
Effect at the time of focus	No effect
Horizontal position	Align right
Input permission	Prohibition

Interpreter method

Macro

```
$ GCInputBox00001-OnCreate
'Specify a text box as the control box to which the INPUT is to be reflected
GCSInputBoxSetRefrectControl(-1, "GCInputBox00001", "GTextBox00000");
'Set the input box control as the focus point at default
GCSChangeActiveFocus(-1, "GCInputBox00001");
$End

$GButton00002-OnClick
'Directly set the input content to the control specified to reflect the INPUT (absolute)
GCSInputBoxSetInputValue(-1, "GCInputBox00001", 0);
'Set the focus at the input box control.
GCSChangeActiveFocus(-1, "GCInputBox00001");
$End

$GButton00003-OnClick
'Add the value in the input box to the value in the control to reflect the INPUT, and set.
GCSInputBoxSetInputValue(-1, "GCInputBox00001", 1);
'Set the focus at the input box control.
GCSChangeActiveFocus(-1, "GCInputBox00001");
$End
```

Compilation method

Source code

```
long GCPanel00000::GCINPUTBOX00001OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
  GBaseObject *pPanel = NULL;
  GBaseObject *pChild = NULL;
  pPanel = GetGBaseObject();
  pChild = GCSGetChild( pPanel, GCINPUTBOX00001 ); // Get the input box control
  if (pChild != NULL) {                                    // When it succeeds to get the input box control
    // Specify a text box as the control box to which the INPUT is to be reflected
    GCSSetRefrectControl( pChild, GTEXTBOX00000 );
    GCSChangeActiveFocus( pPanel, pChild );          // Set the focus at the input box control
  }
  return TRUE;
}

long GCPanel00000::GBUTTON00002OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
  GBaseObject *pPanel = NULL;
  GBaseObject *pChild = NULL;
  pPanel = GetGBaseObject();
  pChild = GCSGetChild( pPanel, GCINPUTBOX00001 ); // Get the input box control
  if (pChild != NULL) {                                    // When it fails to get the input box control
    // Directly set the input content to the control specified to reflect the INPUT (absolute)
    GCSInputBoxSetInputValue( pChild, GCINPUTBOX_INPUTTYPE_ABS );
    GCSChangeActiveFocus( pPanel, pChild );          //   Set the focus at the input box control
  }
  return TRUE;
}

long GCPanel00000::GBUTTON00003OnClick(unsigned short usMessage, long lLParam, long lUParam)
{
  GBaseObject *pPanel = NULL;
  GBaseObject *pChild = NULL;
  pPanel = GetGBaseObject();
  pChild = GCSGetChild( pPanel, GCINPUTBOX00001 ); // Get the input box control
  if (pChild != NULL) {                                    // When it succeeds to get the input box control
    // To the value in the control to reflect the INPUT
    // Add the value in the input box and set
    GCSInputBoxSetInputValue( pChild, GCINPUTBOX_INPUTTYPE_INC );
    GCSChangeActiveFocus( pPanel, pChild );          //   Set the focus at the input box control
  }
  return TRUE;
}
```

18.16 GCSoftKey(Ten-key)

GCSoftKey is a control which displays numerical values and character strings as well as input keys. The key buttons within the ten-key control and the keys on the operation board can be used as input keys.

State

The focusing effect of GCSoftKey includes the following states.

Definition	**Description**
GFOCUS_EFFECT_NON	No focusing effect
GFOCUS_EFFECT_CURSOR	Cursor display at focus

The cursor type of GCSoftKey includes the following states.

Definition	**Description**
GCURSOR_TYPE_NORMAL	Vertical line
GCURSOR_TYPE_RECT	Block

Input method

The input method of GCSoftKey includes the following states.

Definition	**Description**
GCSOFTKEY_INPUTTYPE_ABS	Absolute input
GCSOFTKEY_INPUTTYPE_INC	Incremental input

Input mode

The input mode of GCSoftKey includes the following states.

Definition	**Description**
GCSOFTKEY_INPUTMODE_INSERT	Insert mode
GCSOFTKEY_INPUTMODE_OVERWRITE	Overwrite mode

Importing Structure

The importing and exporting structures of GCSoftKey are shown below.

```
GDefPropertyClass(GSoftKeyProperty, GControlProperty )
        unsigned short      usTitleStringID;        /* Title bar character string resource ID   */
        unsigned short      usBorderID;             /* Solid frame resource ID                  */
        unsigned char       ucDispTitlebar;         /* Existence of title bar                   */
        unsigned char       ucDispOldData;          /* Existence of original value              */
        unsigned char       ucCalcFlag;             /* Operation function                       */
        unsigned char       ucReserve1;             /* Reserve(1)                               */
        GColor              gcOldDataDispBackColor; /* Original value background color          */
        GColor              gcOldDataDispForeColor; /* Original value character color           */
        GColor              gcNewDataDispBackColor; /* Input data display background color      */
        GColor              gcNewDataDispForeColor; /* Input data display character color       */
        GColor              gcDisableBackColor;     /* Background color at the time of disable  */
        unsigned long       ulLimitLength;          /* Number of the maximum characters         */
        unsigned char       ucInputType;            /* Input type                               */
        unsigned char       ucDispType;             /* Display type                             */
        unsigned char       ucCursorDisp;           /* Cursor display                           */
        unsigned char       ucCursorType;           /* Cursor type                              */
        GColor              gCursorBackColor;       /* Cursor background color                  */
        GColor              gCursorForeColor;       /* Cursor character color                   */
        unsigned char       ucAbsIncType;           /* Abs/Inc input method                     */
        unsigned char       ucCharacterType;        /* Uppercase/lowercase character input method */
        unsigned short      usReserve1;             /* Reserve(2)                               */
        GColor              gcSubCursorBackColor    /* Sub cursor background color              */
        GColor              gcSubCursorForeColor    /* Sub cursor character color               */
        unsigned long       ulDispMove;             /* Position movement                        */
        unsigned char       ucReserve2;             /* Reserve(3)                               */
        unsigned char       ucReserve3;             /* Reserve(4)                               */
        unsigned short      usReserve2;             /* Reserve(5)                               */
        long                lReserve1;              /* Reserve(6)                               */
        long                lReserve2;              /* Reserve(7)                               */
        long                lReserve3;              /* Reserve(8)                               */
        long                lReserve4;              /* Reserve(9)                               */
        char                szReserve2[8];          /* Reserve(10)                              */
        char                szReserve2[8];          /* Reserve(11)                              */
GDefPropertyEnd
```

Parent Class

The parent class of GCSoftKey is GControl.

Function List

Functions of GCSoftKey are listed below.

Common Control Function（GCBaseWindow）

Refer to Section 17.3 GCBaseWindow (window control).

Common Control Function（GCControl）

Refer to Section 18.1 GCControl (control management).

Functions Characteristic to GCSoftKey

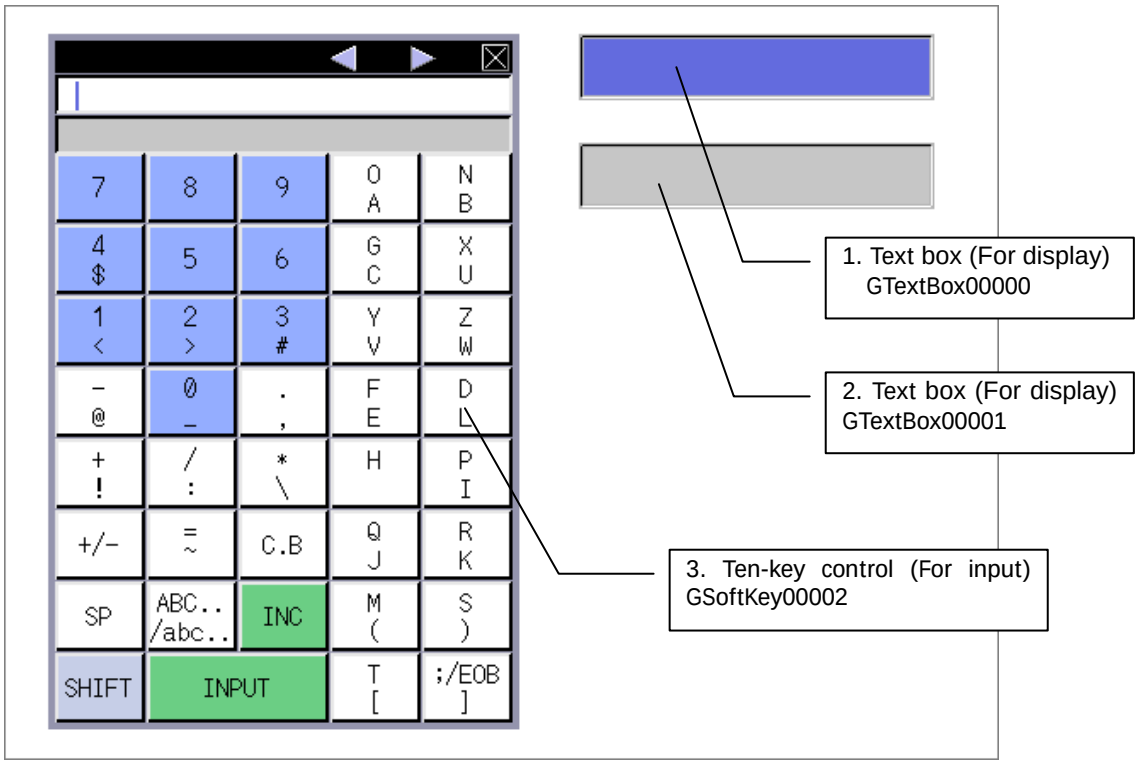
Method name	Method Description
GCSSoftKeySetPasswordStatus	Set password display status
GCSSoftKeyGetPasswordStatus	Get password display status
GCSSoftKeySetString	Set display character string
GCSSoftKeyGetString	Get display character string
GCSSoftKeySetGValue	Set display value
GCSSoftKeyGetGValue	Get display value
GCSSoftKeySetFocusEffect	Set effect during focus
GCSSoftKeyGetFocusEffect	Get effect during focus
GCSSoftKeySetCalcFlag	Set operation function (enable/disable)
GCSSoftKeyGetCalcFlag	Get operation function (enable/disable)
GCSSoftKeySetInputValue	Set the control to reflect the INPUT
GCSSoftKeySetAbsIncType	Set input method (absolute/incremental)
GCSSoftKeyGetAbsIncType	Get input method (absolute/incremental)
GCSSoftKeySetRefrectControl	Set the control to reflect the INPUT
GCSSoftKeyGetRefrectControlTextType	Set character string type of control to reflect the INPUT (character string/numerical value/real number)
GCSSoftKeyGetInputMode	Get input mode (insert/overwrite)
GCSSoftKeySetTitleString	Set title bar display character string
GCSSoftKeyGetSubCursorControl	Get the ID of the sub cursor display control

Usage Example

Create the following sample.

Screen Configuration

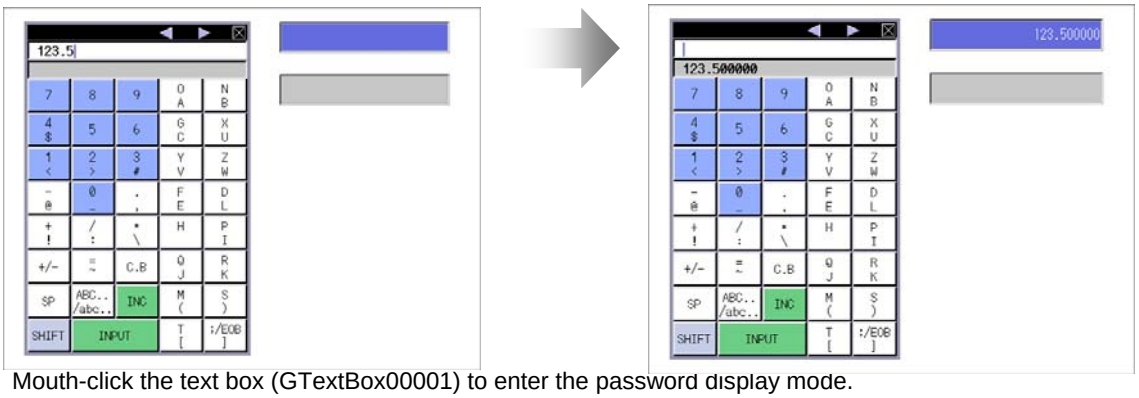
Set the value in the ten-key control to the text box (GTextBox00000) by operating key buttons. The password display mode can be established by moving the sub cursor to the text box (GTextBox00001).



Action State

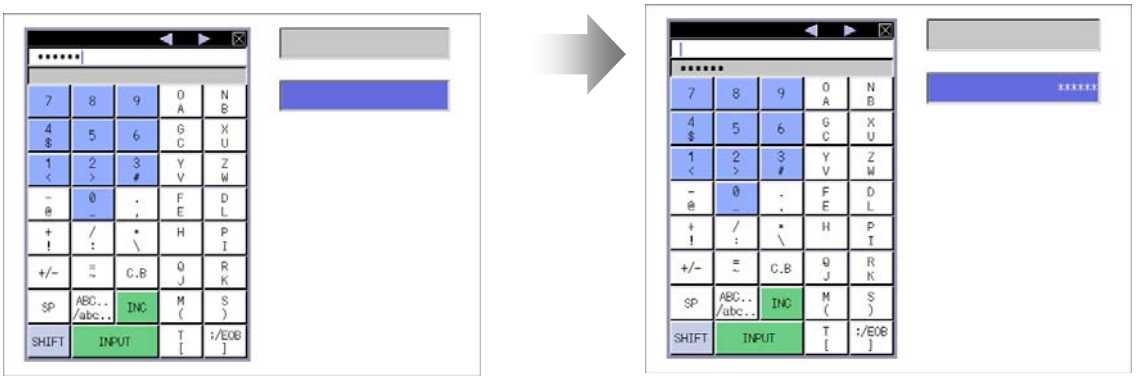
(1) The focus is placed on the ten-key control (GCSoftKey00002) when the sample screen opens. The sub cursor is displayed in the text box (GTextBox00000).
(2) Press the INPUT key after pressing key buttons to set the input value to the text box (GTextBox00000).

<Press the INPUT key after inputting“123.5”>



(3) Mouth-click the text box (GTextBox00001) to enter the password display mode.
(4) The characters input by pressing the key buttons will be displayed in asterisks ”*”. Press the input key to set the input value in the text box (GTextBox00001).

<Press the INPUT key after inputting “PASSWD”>



Property settings

Set the properties of the located parts.

GSoftKey00002	Ten-key
Cursor Display	With cursor
Cursor type	Vertical line
Password setup	None
Input type	ALL key type

GTextBox00000	Text box
Type	float
Existence of caption	Yes
Display format	%f
Password setup	None
Effect at the time of focus	No effect
Horizontal position	Align right
Input permission	Prohibition

GTextBox00001	Text box
Type	Character string
Existence of caption	Yes
Display format	%s
Password setup	Yes
Effect at the time of focus	No effect
Horizontal position	Align right
Input permission	Prohibition

Sub cursor setting

Input area control		GCSoftKey00002			
Key transfer control		NULL			
Init	List of controls	Destination setting			
Select		←	↑	↓	→
*	GTextBox00000			GTextBox00001	
	GTextBox00001		GTextBox00000		

Interpreter method

Macro

```
$GSoftKey00002-OnCreate
'Set the ten-key control as the focus point at default
GCSChangeActiveFocus(-1, " GSoftKey00002");
$End

$GTextBox00000-OnClick
'Cancel the password display setting
GCSSoftKeySetPasswordStatus(-1, "GSoftKey00002", 0);
'Clear the content of input data display part
GCSSoftKeySetString(-1, "GSoftKey00002", "");
$End

$GTextBox00001-OnClick
'Set the password display setting
GCSSoftKeySetPasswordStatus(-1, "GSoftKey00002", 1);
'Clear the content of input data display part
GCSSoftKeySetString(-1, "GSoftKey00002", "");
$End

$GSoftKey00002-OnKeyPress
'When the down arrow key is pressed
if (LLPARAM == 40)
  'Set the password display setting
  GCSSoftKeySetPasswordStatus(-1, "GSoftKey00002", 1);
  'Clear the content of input data display part
  GCSSoftKeySetString(-1, "GSoftKey00002", "");
'When the up arrow key is pressed
elseif(LLPARAM == 38)
  'Cancel the password display setting
  GCSSoftKeySetPasswordStatus(-1, "GSoftKey00002", 0);
  'Clear the content of input data display part
  GCSSoftKeySetString(-1, "GSoftKey00002", "");
endif
$End
```

Compilation method

Source code

```
long GCPanel00000::GSOFTKEY00002OnCreate(unsigned short usMessage, long lLParam, long lUParam)
{
   GBaseObject *pPanel = NULL;
   GBaseObject *pChild = NULL;
   pPanel = GetGBaseObject();
   pChild = GCSGetChild( pPanel, GSOFTKEY00002);      // Get the ten-key control
   if (pChild != NULL) {                                    // When it succeeds to get the ten-key control
      // Set the focus at the ten-key control
      GCSChangeActiveFocus( pPanel, pChild );
   }
   return TRUE;
}

long GCPanel00000:: GSOFTKEY00002OnSubCursorMove (unsigned short usMessage, long lLParam, long
lUParam)
{
   // Get the ID of the sub cursor display control
   long lBfrCtrlId = ((GParam *)lLParam)->lParam[0];    // Before the movement
   long lAftCtrlId = ((GParam *)lUParam)->lParam[0];    // After the movement

   GBaseObject *pPanel = NULL;
   GBaseObject *pChild = NULL;
   GTCHAR _gtText[20];
   _sgtprintf(_gtText, TEXT(""));

   pPanel = GetGBaseObject();
   pChild = GCSGetChild( pPanel, GSOFTKEY00002);      // Get the ten-key control
   if (pChild != NULL) {                                    // When it succeeds to get the ten-key control
      if (lAftCtrlId == GTEXTBOX00001) {
         // When the sub cursor moves to the text box (GTextBox00001).
         GCSSoftKeySetPasswordStatus(pChild, 1);    // Set the password display setting
         if (lBfrCtrlId != lAftCtrlId) {
            // When the controls differ before and after the movement
            GCSSoftKeySetString(pChild, _gtText);         // Clear the content of input data display part
         }
      }
      else {
         // When the sub cursor moves to the text box (GTextBox00000).
         GCSSoftKeySetPasswordStatus(pChild, 0);      // Cancel the password display setting
         if (lBfrCtrlId != lAftCtrlId) {
            // When the controls differ before and after the movement
            GCSSoftKeySetString(pChild, _gtText);         // Clear the content of input data display part
         }
      }
   }

   return TRUE;
}
```

19. GShape (figure control)

This section describes the figure (GShape).

19.1 GCShape (figure control)

GCShape is a parent class of figures. It holds border and brush attributes.

Importing Structure

The importing structures of GCShape are shown below.

```
GDefPropertyClass( GShapeProperty, GBaseWindowProperty )
        GColor          gcPenColor;     /* Line color    */
        unsigned short  usPenSize;      /* Line width    */
        unsigned short  usPenDash;      /* Line type     */
        GBrush          gbFillBrush;    /* Filling brush */
GDefPropertyEnd
```

Parent Class

The parent class of GCShape is GCBaseWindow.

Function List

The functions of GCShape include the followings.

Function name	Function	Access designator
GCShape	Constructor	public
~GCShape	Destructor	public
SetPenColor	Set border color	public
GetPenColor	Get border color	public
SetPenDash	Set line type	public
GetPenDash	Get line type	public
SetPenSize	Set line width setting	public
GetPenSize	Get line width	public
SetBrush	Set filling brush	public
GetBrush	Get filling brush	public
ImportProperty	Import properties setting	public
SetDrawAttribute	Set figure drawing attribute	public
AddRefreshRect	Register client area re-drawing area	public

19.2 GCSRect (rectangle)

Importing Structure

There is no importing member for GCSRect.

Class (structure) Name

GCSRect

Parent Class

GCShape

Function List

The functions of GCSRect include the followings.

Function name	Function	Access designator
GCSRect	Constructor	public
~GCSRect	Destructor	public
OnDraw	Client area drawing process	public

19.3 GCSOval (circle & oval)

Importing Structure

There is no importing member for GSCvOval.

Class (structure) Name

GCSOval

Parent Class

GCShape

Function List

The functions of GCSRect include the followings.

Function name	Function	Access designator
GCSOval	Constructor	public
~GCSOval	Destructor	public
OnDraw	Client area drawing process	public

19.4 GCSPoly (polygon)

Importing Structure

The importing structure of GCSPoly is shown below.

```
GDefPropertyClass( GSPolyProperty, GShapeProperty )
                    GPoly       gpPoly;             /* Polygon          */
GDefPropertyEnd
```

Class (structure) Name

GCSPoly

Parent Class

GCShape

Function List

The functions of GCSPoly include the followings.

Function name	Function	Access designator
GCSPoly	Constructor	public
~GCSPoly	Destructor	public
SetPoly	Set polygon	public
GetPoly	Get polygon	public
ImportProperty	Import properties setting	public
OnDraw	Client area drawing process	public

19.5 GCSPie (sector)

Importing Structure

The importing structure of GCSPie is shown below.

```
GDefPropertyClass( GSPieProperty, GShapeProperty )
        short nStartAngle;                  /* Starting angle */
        short nEndAngle;                    /* Last angle     */
GDefPropertyEnd
```

Class (structure) Name

GCSPie

Parent Class

GCShape

Function List

The functions of GCSPie include the followings.

Function name	Function	Access designator
GCSPie	Constructor	public
~GCSPie	Destructor	public
SetAngle	Set sector starting angle and end angle	public
GetAngle	Get sector starting angle and end angle	public
ImportProperty	Import properties setting	public
OnDraw	Client area drawing process	public

19.6 GCLineShape (line drawing control)

Importing Structure

The importing structure of GCLineShape is shown below.

```
GDefPropertyClass( GLineShapeProperty, GBaseWindowProperty )
        GColor          gcPenColor;     /* Line color */
        unsigned short  usPenSize;      /* Line width*/
        unsigned short  usPenDash;      /* Line type  */
GDefPropertyEnd
```

Class (structure) Name

GCLineShape

Parent Class

GCBaseWindow

Function List

The functions of GCLineShape include the followings.

Function name	Function	Access designator
GCLineShape	Constructor	public
~GCLineShape	Destructor	public
SetPenColor	Set line color	public
GetPenColor	Get line color	public
SetPenDash	Set line type	public
GetPenDash	Get line type	public
SetPenSize	Set line width	public
GetPenSize	Get line width	public
ImportProperty	Import properties setting	public
SetDrawAttribute	Set line figure drawing attribute	public
AddRefreshRect	Register client area re-drawing area	public

19.7 GCSLine (line)

Importing Structure

The importing structure of GCSLine is shown below.

```
GDefPropertyClass( GSLineProperty, GLineShapeProperty )
        GLine                   glLine;             /* Coordinates of line      */
GDefPropertyEnd
```

Class (structure) Name

GCSLine

Parent Class

GCLineShape

Function List

The functions of GCSLine include the followings.

Function name	Function	Access designator
GCSLine	Constructor	public
~GCSLine	Destructor	public
SetStartPoint	Set line starting point	public
GetStartPoint	Get line starting point	public
SetEndPoint	Set line end point	public
GetEndPoint	Get line end point	public
ImportProperty	Import properties setting	public
OnDraw	Client area drawing process	public

19.8 GCSLines (connected lines)

Importing Structure

The importing structure of GCSLines is shown below.

```
GDefPropertyClass( GSLinesProperty, GLineShapeProperty )
        GPoly          gpPoly;          /* Coordinates of connected lines    */
GDefPropertyEnd
```

Class (structure) Name

GCSLines

Parent Class

GCLineShape

Function List

The functions of GCSLines include the followings.

Function name	Function	Access designator
GCSLines	Constructor	public
~GCSLines	Destructor	public
SetPoly	Set connected line	public
GetPoly	Get connected line	public
ImportProperty	Import properties setting	public
OnDraw	Client area drawing process	public

19.9 GCSArc (arc)

Importing Structure

The importing structure of GCSArc is shown below.

```
GDefPropertyClass( GSArcProperty, GLineShapeProperty )
        short nStartAngle;          /* Starting angle */
        short nEndAngle;            /* End angle      */
GDefPropertyEnd
```

Class (structure) Name

GCSArc

Function List

GCLineShape

Function List

The functions of GCSArc include the followings.

Function name	Function	Access designator
GCSArc	Constructor	public
~GCSArc	Destructor	public
SetAngle	Set arc starting angle and end angle	public
GetAngle	Get arc starting angle and end angle	public
ImportProperty	Import properties setting	public
OnDraw	Client area drawing process	public

20. GDraw (drawing)

This section describes GDraw, which gives settings of the drawing environment for basic figures, characters, coordinates, color and so on.

20.1 Outline

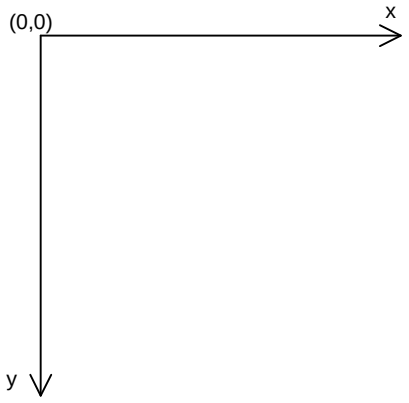
The GDraw module sets the drawing environment for basic figures, characters, coordinates, color and so on in NC Designer. The basic specifications necessary for the operation of GDraw are described below.

Coordinate System

NC Designer uses the traditional Cartesian rectangular coordinate system in which the X-axis is drawn from the left to the right on the screen and the Y-axis is drawn from the top to the bottom on the screen.
The following two coordinate systems are supported.

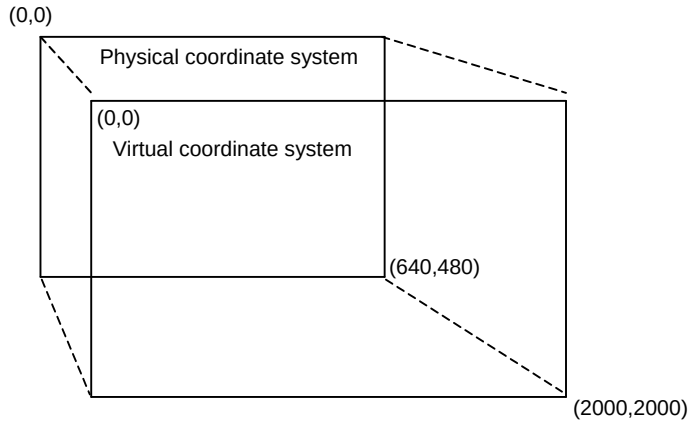
Physical Coordinate System

Each dot (pixel) of the screen corresponds to "1" in the X/Y coordinate. The upper left of the screen is the origin and the default coordinates of the origin are (0, 0).



Virtual Coordinate System

In this coordinate system, coordinates can be specified independent of the physical coordinate system while calculation of coordinates of the displayed object needs an overhead.
The screen size is arbitrarily specified and programs can be built independent of the screen size so that consistency is assured.
Example: While the physical X and Y coordinates are 0 to 640 and 0 to 480 respectively, designate the virtual X and Y coordinates at 0 to 2000 and 0 to 2000 respectively.



In these two coordinate systems, X and Y values indicate coordinates. To control these coordinates, NC Designer defines the following types.

Rectangle Structure

```
typedef struct _GRect{
                short       nXmin;      /* X coordinate of upper left corner of rectangle   */
                short       nYmin;      /* Y coordinate of upper left corner of rectangle   */
                short       nXmax;      /* X coordinate of lower right corner of rectangle */
                short       nYmax;      /* Y coordinate of lower right corner of rectangle */
}GRect;
```

GRect is a rectangle structure of NC DESIGNER. The drawing area in which the user program calls the drawing functions of NC DESIGNER is designated with this Grect in principle. The size of the oval is determined so that it inscribes this rectangle.

Point Structure

```
typedef struct _GPoint{
                short       nX;         /* X coordinate */
                short       nY;         /* Y coordinate */
}GPoint;
```

GPoint is a point structure of NC Designer. Each vertex of a polygon is indicated with GPoint.

Polygon Structure

```
typedef struct _GPoly{
        short       nNumPoints;   /* Number of vertices                    */
        short       *pgptPoints;  /* Coordinate data array pointer         */
        GRect       grBounds;     /* Rectangular area enclosing the polygon*/
}GPoint;
```

GPoly is a structure of NC Designer indicating a polygon. Use GPoly to indicate a polygon such as a triangle.

Line Structure

```
typedef struct _GLine{
        short       nX1;          /* X coordinate of starting point */
        short       nY1;          /* Y coordinate of starting point */
        short       nX2;          /* X coordinate of end point      */
        short       nY2;          /* Y coordinate of end point      */
}GLine;
```

GLine is a structure of NC Designer indicating a line.

Color

The color of NC Designer supports the direct color system.

Direct Color System

In the direct color system, the R, G and B values are designated and they are written in the VRAM.

Use GColor as a color variable.

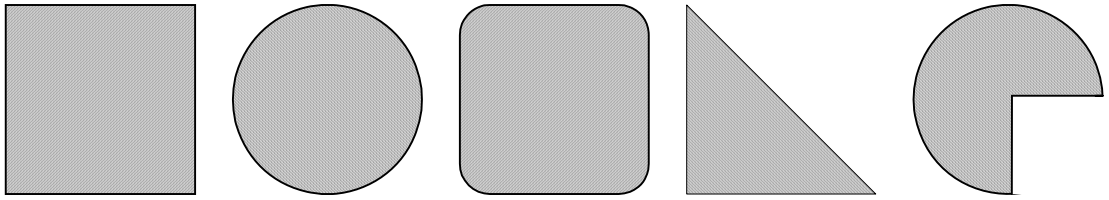
```
#define GColor      long            /* For 24-bit color      */
```

If NO_DRAW (-1) is designated as a color variable of the pen color or foreground or background color of a fill, drawing is not made.

Basic Figure Drawing

NC Designer is provided with a function for drawing the following basic figures. Each figure is drawn with or without a fill. The line pattern of the border of the figure follows the current setting.

- Rectangle ...GDDrawRect()
- Circle and oval ...GDDrawOval()
- Rectangle with round edges ...GDDrawRRect()
- Polygon ...GDDrawPoly()
- Sector ...GDDrawPie()



Figures

In addition, the following line drawing functions are provided.

- Line ...GDLineTo()
- Arc ...GDDrawArc()

Line figures

Drawing Attributes

The following drawing attributes are controlled in the drawing handle structure HGDRAW and these parameters are used during execution of drawing processes.

- Filling pattern ...GDSetFillPattern(),GDDefinePattern(),GDSetFillAttribute()
- Foreground color of fill ...GDSetForeColor () ,GDSetFillColor(),GDSetFillAttribute()
- Background color of fill ...GDSetBackColor() ,GDSetFillColor(),GDSetFillAttribute()
- Brush ...GDSetBrush()
- Line thickness ...GDSetPenSize()
- Line pattern ...GDSetPenDash(),GDDefinePenDash()
- Line connection shape ...GDSetPenJoin(),
- Pen color ...GDSetPenColor()
- Raster operation ...GDSetDrawingMode()

Each parameter is described below.

Filling Pattern

The filling pattern is a binary pattern, ranging in eight dots vertically and in eight dots horizontally. It is expressed with "0" or "1" and it is used for filling. "0" indicates the background, and "1" indicates the foreground.

The patterns that can be specified includes the followings.

0 to 38	...Filling using the filling pattern corresponding to the designated No.
NO_DRAW(-1)	...No fill
FILL_BACK_COLOR(-2)	...Filling in background color
FILL_FORE_COLOR(-3)	...Filling in foreground color

The filling patterns designated with 0 to 38 can be specified by the user.

Foreground Color of Fill

Foreground color of filling pattern.
Use the GColor type to designate.
If NO_DRAW (-1) is designated, the foreground part (1) of the filling pattern is not filled.

Background Color of Fill

Background color of filling pattern.
Use the GColor type to designate.
If NO_DRAW (-1) is designated, the background part (0) of the filling pattern is not filled.

Brush of Fill

The brush is a structure set of the pattern, foreground color and background color that are necessary for filling.

```
typedef struct _GBrush{
        short       nFillPattern;     /* Filling pattern            */
        GColor      gcForeColor;      /* Foreground color of fill */
        GColor      gcBackColor;      /* Background color of fill */
} GBrush
```

Line Thickness

The line thickness is indicated with the X and Y dot sizes.
The line figures (line and arc) and the border of the figure are drawn in reference to this line thickness setting.

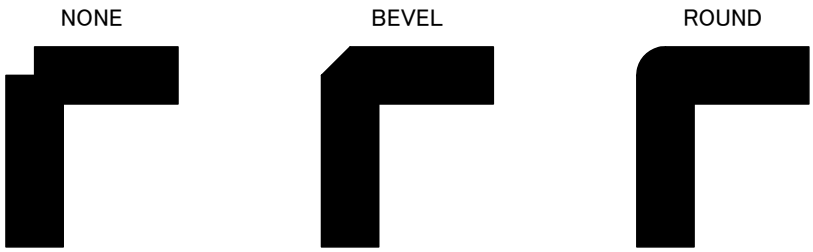
Line Pattern

The line pattern is a horizontal eight-dot binary pattern indicated with "0" and "1," and it is used for line drawing.
Designate the line pattern with a number between 0 and 8 among line patterns defined in the drawing handle structure. (Pattern 0 indicates the line.)
Line patterns 1 to 8 can be designed by the user. (Pattern 0 may not be designed.)

Line Connection Shape

The line connection shape indicates the shape of the joint of lines of the rectangle and polygon. One of three types can be selected among: NONE, BEVEL and ROUND.

GLINEJOIN_TYPE_NONE GLINEJOIN_TYPE_BEVEL GLINEJOIN_TYPE_ROUND



Pen Color

The pen color is used to draw the line, border of the figure and character.
If NO_DRAW (-1) is designated, the line is not drawn.

Raster Operation

This is designation of the raster operation for figure drawing. The raster operation includes the following types.

Raster operation	Description
GRASTER_OPERATION_REPLACE	The base is overwritten.
GRASTER_OPERATION_XOR	Exclusive OR. Write twice to restore the original.
GRASTER_OPERATION_AND	AND drawing with base
GRASTER_OPERATION_OR	OR drawing with base

Figure Operation

Drawing operation functions for GRect, GPoint, GPoly and GLine defined in NC Designer are prepared. The basic operations are shown below.

Operation	Description
Set	The value is set in the member of the structure.
Get	The value is acquired to the member of the structure.
Copy	The setting of a structure is duplicated to another structure.
Enlarge/reduce	The figure is converted from a rectangular area to another rectangular area at a certain ratio.
Offset	The figure is moved by the designated distance.
Virtual-to-physical coordinate conversion	The virtual coordinates of the figure are converted into physical coordinates.
Physical-to-virtual coordinate conversion	The physical coordinates of the figure are converted into virtual coordinates.

The following additional operations are prepared for GRect.

Operation	Description
Shift	Shift the designated rectangular area by the designated distance, and acquire the area indicating one or two areas emptied as a result of shifting.
Resize	Resize the designated rectangular area by the designated scale.
Sum	Calculate the minimum rectangular area including designated two or more rectangular areas.
Product	Calculate the overlapped area of the designated two rectangular areas.
Agreement	Agreement of the designated two rectangular areas is tested.
Inclusion	The coordinates of the designated point are checked to be in the rectangular area.

Image

NC Designer supports the following image formats.

GRF(GUI Raster Format)
This is the NC Designer's original raster image format.
The format of the header is indicated below.

```
typedef struct _GRFHeader{
        short           nType;      /* Type                                 */
        short           nWidth;     /* Number of pixels of horizontal width */
        short           nHeight;    /* Number of pixels of vertical width   */
        short           nBpp;       /* Number of bits per each pixel        */
        long            lSize;      /* Image size                           */
        unsigned char *pData;       /* Pointer of image data                */
}GRFHeader;
```

The type includes the following.

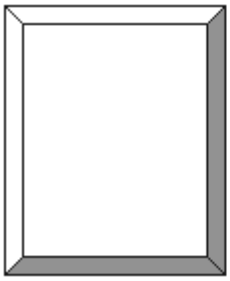
- GRF_TYPE_DDBRLE Device-dependent run-length compression type

3D Effect

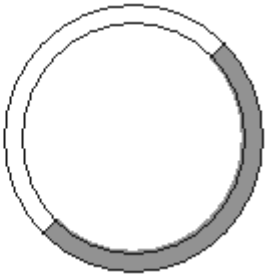
NC Designer supports the following functions to realize 3D effects.

Solid Frame Drawing

A drawing function for adding a solid frame easily to the figure is prepared.
The solid frame includes two types: rectangular solid frame and round solid frame.



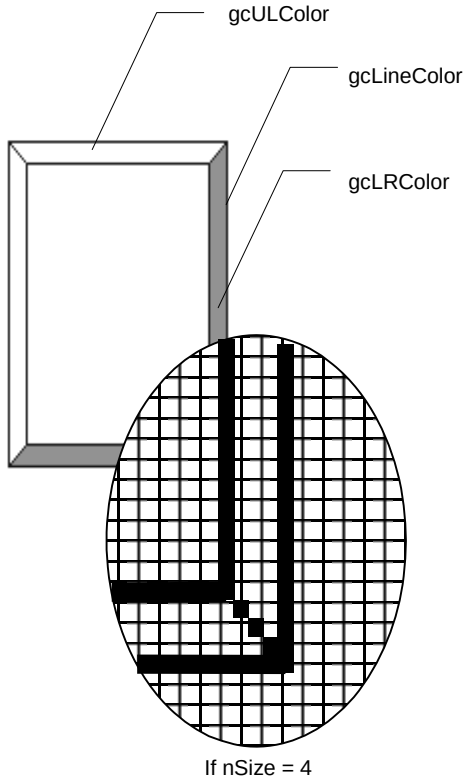
Rectangular solid



Round (oval) solid

The following structures are defined to realize a solid frame.

```
typedef struct _GBorder{
        unsigned char fBorder;          /* Presence of solid frame              */
        GColor        gcULColor;        /* Color of upper left part of frame    */
        GColor        gcLRColor;        /* Color of lower right part of frame   */
        GColor        gcLineColor;      /* Color of border                      */
        short         nSize;            /* Width of solid frame. 1: border only */
}GBorder;
```



If nSize = 4

Resource Support

NC Designer has a resource control mechanism called GResource. GDraw is provided with a mechanism for entering data and drawing in the font, character string, image and solid frame function by designating a resource ID.

20.2 Description of Function

Function List

Initialization and Termination Processes

Function name	Function
GDInitPlatform	The platform is initialized.
GDQuitPlatform	The platform is terminated.

Drawing environment settings

Function name	Function
GDOpenDrawHandle	Get drawing handle
GDCloseDrawHandle	Release drawing handle
GDBeginDraw	Begin drawing
GDEndDraw	End drawing
GDFlushScreen	Refresh screen
GDSetDisplay	Set graphics mode
GDSetClipRect	Define clipping area
GDInceptClipRect	Clip the area of overlap between the specified rectangle and the current clip area
GDSetDrawingMode	Set raster operations for drawing
GDSetLocalMode	Set to local (physical) coordinate system
GDSetVirtualMode	Set to virtual coordinate system
GDSetVirtualRect	Set rectangular area in virtual coordinates
GDOffsetLocalOrigin	Offset to local coordinate origin
GDSetPenColor	Set line and character color
GDSetPenSize	Set line width (pen size)
GDSetPenDash	Set line pattern
GDSetPenJoin	Specify joint for rectangles and polygons
GDDefinePenDash	Define line pattern
GDMoveTo	Move pen position
GDSetForeColor	Set foreground color
GDSetBackColor	Set background color
GDSetFillColor	Set colors for foreground and background
GDSetFillPattern	Set fill pattern
GDSetFillAttribute	Set fill attributes
GDSetBrush	Set brush
GDDefineFillPattern	Define fill pattern
GDCalcBorderRect	Calculate rectangular area based on 3D border settings
GDGetPalette	Get palette color code

Figure Drawing

Function name	Function
GDSetPixel	Draw pixels
GDGetPixel	Get color code for specified coordinates
GDLineTo	Draw line
GDDrawArc	Draw arc
GDDrawRect	Draw rectangle
GDDrawOval	Draw oval
GDDrawPie	Draw sector
GDDrawPoly	Draw polygon
GDDrawRRect	Draw rounded corner rectangle
GDDrawBezier	Draw Bezier curve

Font

Function name	Function
GFLoadFont	Load specified font
GFSetSize	Set character size
GFGetSize	Get font size
GFSetFontAttribute	Set font attributes
GFGetFontAttribute	Get font attributes
GFSetFontUserAttribute	Set user font attributes
GFGetFontUserAttribute	Get user font attributes
GFReleaseFont	Release font
GFGetFontCount	Get number of fonts that can be used
GFGetFontInformation	Get information for font of specified No.

Text

Function name	Function
GDSetFont	Set font
GDSetSimpleFont	Set simple font
GDDrawString	Draw character string
GDDrawCaptionString	Draw caption character string
GDDrawCaptionCursorString	Draw caption character string (with cursor display)
GDGetCharWidth	Get specified character width
GDGetCharHeight	Get specified character height
GDGetStringWidth	Get specified character string width
GDGetNextChar	Get next character
GDGetPrevChar	Get previous character
GDGetStringLength	Get character string length

Figure Operation

(1) Rectangle

Function name	Function
GRSetRect	Define rectangular area
GRGetRect	Get rectangle area
GRDuplicateRect	Copy rectangle
GRMapRect	Enlarge/reduce rectangular area
GROffsetRect	Rectangle offset processing
GRShiftRect	Rectangle shift processing
GRResizeRect	Resize rectangle
GRUnionRect	Join rectangle areas
GRInceptRect	Confirm rectangular area overlap
GREqualRect	Confirm that rectangle areas are the same
GRPointInRect	Check if point coordinates fall within rectangle coordinates
GRPhysicalToVirtualRect	Convert rectangle physical coordinates to virtual coordinates
GRVirtualToPhysicalRect	Convert rectangular area virtual coordinates to physical coordinates
GRMapToScreenRect	Convert rectangular area defined with pgrRect to screen coordinate system
GRMapToVirtualRect	Convert rectangular area defined with pgrRect to virtual screen coordinate system

(2) Point

Function name	Function
GPTSetPoint	Define point structure
GPTGetPoint	Get point structure content
GPTDuplicatePoint	Copy point structure
GPTMapPoint	Map rectangle defined with pgrFrom to rectangle defined with pgrTo
GPTOffsetPoint	Point structure offset processing
GPTPhysicalToVirtualPoint	Convert point structure physical coordinates to virtual coordinates
GPTVirtualToPhysicalPoint	Convert point structure virtual coordinates to physical coordinates
GPTMapToScreenPoint	Convert point structure to screen coordinates
GPTMapToVirtualPoint	Convert point structure physical coordinates to virtual coordinates

(3) Line

Function name	Function
GLSetLine	Define line structure
GLGetLine	Get line
GLDuplicateLine	Copy line structure
GLMapLine	Enlarge/reduce line coordinates
GLOffsetLine	Offset processing for line structure
GLPhysicalToVirtualLine	Convert line structure physical coordinates to virtual coordinates
GLVirtualToPhysicalLine	Convert line structure virtual coordinates to physical coordinates

(4) Polygon

Function name	Function
GPSetPoly	Define polygon
GPGetPoly	Get polygon settings
GPMapPoly	Enlarge/reduce polygon
GPOffsetPoly	Polygon offset processing
GPPhysicalToVirtualPoly	Convert polygon physical coordinates to virtual coordinates
GPVirtualToPhysicalPoly	Convert polygon virtual coordinates to physical coordinates

(5) Image

Function name	Function
GDDrawGrf	Draw BMP
GDDrawImage	Draw image

VRAM Operation

Function name	Function
GDSetVram	Set VRAM for drawing destination
GDBitBlt	Block transfer

Resource Operation

Function name	Function
GDSetSimpleFontResource	Set resource simple font
GDDrawImageResource	Draw resource image
GDDrawStringResource	Draw resource character string
GDDrawCaptionStringResource	Draw caption based on character string resources
GDDrawRectBorderResource	Draw rectangular resource 3D border
GDDrawOvalBorderResource	Draw (oval) circular resource 3D border

3D Effect

Function name	Function
GDDrawRectBorder	Draw rectangular 3D border
GDDrawOvalBorder	Draw (oval) circular 3D border

Memory management

Function name	Function
GMSetMemorySpace	Register memory space information
GMDeleteMemorySpace	Delete registered memory space
GMMemoryAlloc	Allocate memory space
GMMemoryFree	Release memory space

Other

Function name	Function
GDQueryError	Get final error code

Macro Definition

No Drawing

```
#define NO_DRAW          -1          /* No drawing (used for filling pattern and line color) */
```

Filling Pattern

```
#define FILL_BACK_COLOR  -2          /* Background color of fill                             */
#define FILL_FORE_COLOR  -3          /* Foreground color of fill                             */
```

Definition of Line Connection

```
#define NONE             0           /* No treatment                                         */
#define BEVEL            1           /* Line                                                 */
#define ROUND            2           /* Rounded corner                                       */
```

Horizontal Position of GCaption Nameplate

```
#define GCAPTION_POS_LEFT    0       /* Align left                                           */
#define                      1       /* Center                                               */
GCAPTION_POS_CENTER
#define GCAPTION_POS_RIGHT   2       /* Align right                                          */
```

Vertical Position of GCaption Nameplate

```
#define GCAPTION_POS_TOP     0       /* Align top                                            */
#define                      1       /* Center                                               */
GCAPTION_POS_CENTER
#define                      2       /* Align bottom                                         */
GCAPTION_POS_BOTTOM
```

Drawing Handle

```
#define HGDRAW           GDraw*      /* Drawing handle                                       */
```

VRAM Handle

```
#define HGVRAM           GVram*      /* VRAM handle                                          */
```

Type Definition

Rectangle

```
typedef struct _GRect{
        short       nXmin;      /* X coordinate of upper left of rectangle    */
        short       nYmin;      /* Y coordinate of upper left of rectangle    */
        short       nXmax;      /* X coordinate of lower right of rectangle   */
        short       nYmax;      /* Y coordinate of lower right of rectangle   */
}GRect;
```

Point Coordinate

```
typedef struct _GPoint{
        short       nX;         /* X coordinate                               */
        short       nY;         /* Y coordinate                               */
}GPoint;
```

Line

```
typedef struct _GLine{
        short       nX1;        /* Starting point (X coordinate)              */
        short       nY1;        /* Starting point (Y coordinate)              */
        short       nX2;        /* End point (X coordinate)                   */
        short       nY2;        /* End point (Y coordinate)                   */
}GLine;
```

Polygon

```
typedef struct _GPoly{
        short       nNumPoints; /* Number of vertices                         */
        GPoint      *pgptPoints;/* Coordinates of vertex                      */
        GRect       grBounds;   /* Circumscribed rectangle of polygon         */
}GPoly;
```

Graphic Structure

```
typedef struct _GDraw{
        GRect           grLocalRect;            /* Physical coordinate area                    */
        GPoint          gptLocalOrigin;         /* Origin of physical coordinate               */
        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 connection shape                       */
        short           nPenDash;               /* Line type No.                               */
        short           nDashOffset;            /* Line pattern offset                         */
        short           nTextMode;              /* Text mode                                   */
        GColor          gcForeColor;            /* Foreground color of fill                    */
        GColor          gcBackColor;            /* Background color of fill                    */
        short           nFillPattern;           /* Filling pattern No.                         */
        short           nDrawCondition;         /* Drawing condition                           */
        short           nDrawingMode;           /* Raster operation                            */
        HGVRAM          hSystemVram;            /* Original VRAM                               */
        HGVRAM          hDrawVram;              /* Drawing VRAM                                */
        HGFONT          hFont;                  /* Font                                        */
        void            *vgdftDraw;             /* Drawing function table                      */
        char            cDashPatterns[NUM_DASH_PATS][DASH_PAT_SIZE];
                                                /* Line pattern                                */
        char            cFillPatterns[NUM_FILL_PATS][FILL_PAT_SIZE];
                                                /* Filling pattern                             */
        GDrawParamPoly gpPoly;                  /* Polygon drawing parameter                   */
        GDrawParamWideLine gpwWideLine;         /* Thick line drawing parameter                */
        GDrawParamOval gpwOval;                 /* Circle, arc and sector drawing parameter    */
        unsigned short usErrorCode;             /* Previous error code                         */
}GDraw;
```

VRAM structure

```
typedef struct _GVram{
        short           nType;                  /* VRAM type                                   */
        short           nWidth;                 /* Width                                       */
        short           nHeight;                /* Height                                      */
        short           nBitsPerPixel;          /* Number of bits per each pixel               */
        short           nBytesPerLine;          /* Number of bytes per line                    */
        char            *pcFrameBuffer;         /* Frame buffer                                */
        short           nType;                  /* VRAM type                                   */
        void            *pvData;                /* User extension area                         */
}GVram;
```

21. GResource (resource control)

This section describes GResource that handles character strings, images and other resources.

21.1 Outline

21.1.1 Font

The font data is controlled as the GSimpleFont array. The data at array 0 is disabled and resource data is stored at array 1 and later arrays.
Font control variable name ...GSimpleFont pFonts[];
An example of font resource generation is shown below.

```
/* Font */
const GSimpleFont pFonts00[]={
    {"",0,0,0,0,0},
    {"MS Gothic",16,1,1,0,0},
    {"MS Mincho",16,1,1,0,0},
};
```

21.1.2 Character String

The character string data is controlled as a char* array. The data at array 0 is disabled and resource data is stored at array 1 and later arrays.
Character string control variable name: char *pStrings[];
An example of character string resource generation is shown below.

```
/* String */
const GTCHAR *pStrings00[]={
    "",
    "abcde",
    "12345",
};
```

21.1.3 Palette

The palette used in applications using NC Designer is controlled as the GPalletHeader structure array. The data at array 0 is disabled and resource data is stored at array 1 and later arrays.
An example of palette resource generation is shown below.

```
/* Palette */
const GColor pcData_ID_PALETTE00006[]={
    RGB16(0,0,0),RGB16(0,0,128), ...(omitted)
};

const GPaletteHeader GPHeader_ID_PALETTE00006={256,pcData_ID_PALETTE00006};

const GPaletteHeader* pPalette[]={
    NULL,
    &GPHeader_ID_PALETTE00006,
};
```

21.1.4 Image

The image data is controlled as the GImage structure array. The data at array 0 is disabled and resource data is stored at array 1 and later arrays.
Image control variable name ... GImage pImages[];
An example of character string resource generation is shown below.

```
/* Image */
const unsigned char pcID_IMAGE00001[]={
0xEE,0x86,0x0C,0x44,0xF0,0x3D,0x7B,0x6F, 0xF7,0x5E,0x4C,0x29,0xDB,0x86, ...(omitted)
};
const GRFHeader GRFID_IMAGE00001={GRF_TYPE_DDBRLE,32,32,16,653,(unsigned char
*)pcID_IMAGE00001};

const GImage pImages[]={
    {NULL},
    {(GRFHeader*)&GRFID_IMAGE00001},
};
```

21.1.5 Solid Frame

The solid frame data is controlled as the GBorder structure array. Data at array 0 is disabled and resource data is stored at array 1 and later arrays.
Solid frame control variable name ...GBorder pBorders[];
An example of solid frame resource generation is shown below.

```
/* Border */
const GBorder pBorders[]={
    {0,0,0,0,0},
    {1,RGB16(255,255,255),RGB16(128,128,128),RGB16(0,0,0),3},
    {1,RGB16(0,0,0),RGB16(255,255,255),RGB16(180,180,180),3},
};
```

21.1.6 HTML File Name

The HTML file name data is controlled as the GFileName array. Data at array 0 is disabled and resource data is stored at array 1 and later arrays.
HTML file name control variable name ...GFileName *pHtmls[];
An example of HTML file name resource generation is shown below.

```
/* HtmlBrowser */
const GFileName pHtmls[]={
    "",
    "index.htm",
};
```

21.1.7 Language Identification Character String

The language identification character string data is controlled as a char* array.
Language identification character string control variable name ... GTCHAR *pLanguageStr[];
An example of language identification character string resource generation is shown below.

```
/* Language Discrimanation String */
extern GTCHAR pLanguageStr00[];

/* Language Discrimanation String */
GTCHAR *pLanguageStr[]={
    pLanguageStr00,
};
```

21.2 Description of Function

21.2.1 Function List

Name	Outline
GRCLoadFont	Get specified ID font resource
GRCLoadString	Get specified ID character string resource
GRCLoadImage	Get specified ID image resource
GRCLoadBorder	Get specified ID 3D border resource
GRCLoadHtmlFileName	Get HTML file name for specified ID
GRCLoadLanguageStr	Get specified locale language-identifying character strings resource

22. GEvent (event control)

This section describes GEvent that controls events.

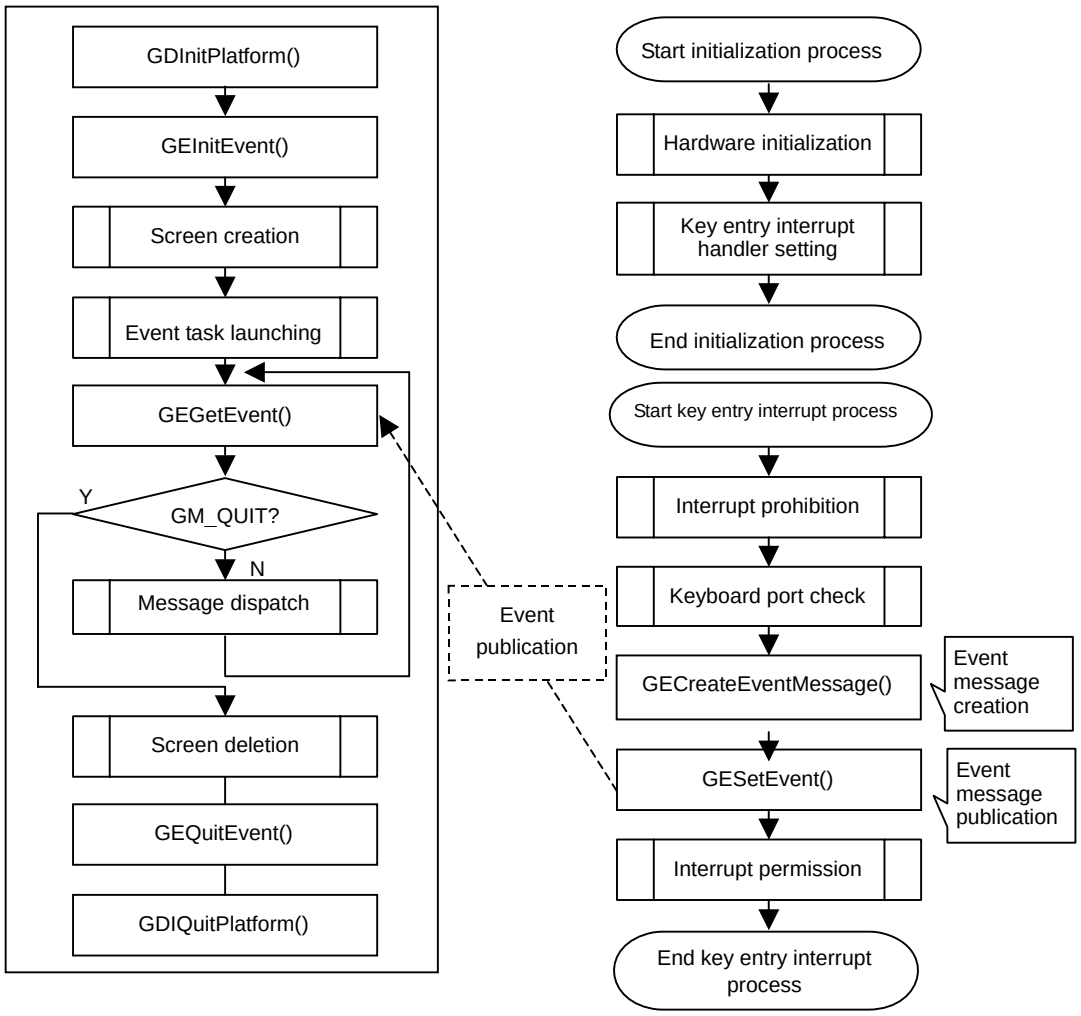
22.1 Outline

NC Designer processes the following events. When an event occurs, a message is specified for the event, and the message is sent to the main processor of the GUI task.

Event type	Event ID	Description
System event	0x000 to 0x0FF	Common system event (incl. timer event)
Mouse event	0x100 to 0x1FF	Events covering general pointing devices
Key event	0x200 to 0x2FF	Key entry event
Window event	0x1000 to 0x1FFF	Window system event
User-defined event	0xA000 to 0xAFFF	Original event defined by user

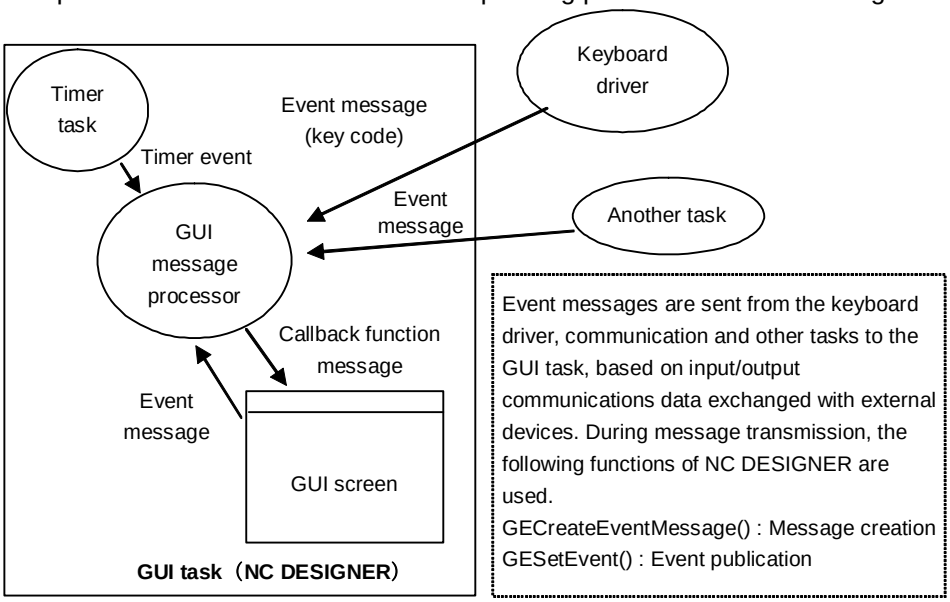
22.2 Flow of Event Processing

The flow of GUI tasks and interrupt handler processes (shown below is an example of key entry interrupt) that use NC Designer is shown below. The key entry interrupt process is registered in the system initialization process, and an entry key code is acquired in the key entry interrupt process (interrupt handler) each time key entry interrupt is generated, and the key entry code is sent as an event message to the main GUI task.



22.3 Event Function

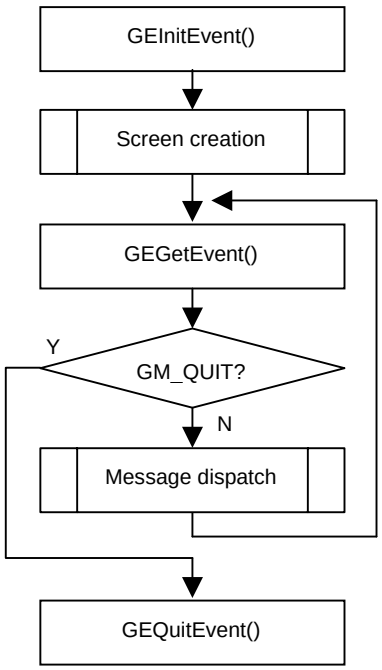
Controls and so on arranged on the panel send messages through the application developed with NC Designer, to the GUI message processor. The message received at the GUI message processor is dispatched via the screen to the corresponding panel and controls arranged on the panel.



Event function has the following functions.

Name	Outline
GEInitEvent	Initialize event system
GEQuitEvent	Process event system termination
GECreateEventMessage	Define event structure
GESetEvent	Register event
GEGetEvent	Acquire event
GESetSystemTime	Set system time
GEGetSystemTime	Get system time

These functions are combined and the received event message is processed in a flow shown below.



Initialize event processing

Initialize physical screen and display

Receive event message

Check the received event and dispatch the messages other than the termination message (GM_QUIT) to the screen.

End event processing

GUI task

22.4 Event Message

The message used in event processes is configured from the following pieces of data.

```
typedef struct GEventMessage{
        unsigned short usMessage;       /* Message type                    */
        void *pvTarget;                 /* Destination of transmission     */
        long lLParam;                   /* Message parameter (high order)  */
        long lUParam;                   /* Message parameter (low order)   */
        unsigned long ulLTime;          /* Generation time (low order)     */
        unsigned long ulUTime;          /* Generation time (high order)    */
}GEventMessage;
```

Specify the desired event message at the message type (usMessage) in the data component shown above and designate the destination of transmission.
Specify the following data at the destination of transmission (pvTarget).

NULL	:	The message is sent to the screen.
Object pointer	:	The message is sent to the control. GetChild() is used for acquisition of the control object.

The message parameter (ulLParam and u1UParam) is used as a parameter for each event message.
Specify the generation time (u1LTime and u1UTime) for identification when necessary. The time acquired with GEGetSystemTime() is used to specify the generation time.

22.5 Event Message Type

A message list that can be specified as a message type is shown below.

Definition	Value	Description
GM_QUIT	0x0000	End of system
GM_TIMER	0x0001	Timer event
GM_CHAR	0x0002	Key event
GM_LBUTTONPRESS	0x0101	Press of left mouse button
GM_LBUTTONRELEASE	0x0102	Release of left mouse button
GM_RBUTTONPRESS	0x0103	Press of right mouse button
GM_RBUTTONRELEASE	0x0104	Release of right mouse button
GM_MBUTTONPRESS	0x0105	Press of middle mouse button
GM_MBUTTONRELEASE	0x0106	Release of middle mouse button
GM_KEYPRESS	0x0200	Press of hardware key
GM_KEYRELEASE	0x0201	Release of hardware key
GM_USER	0x01006	User-defined event
GM_SHOWPANEL	0x01100	Panel switching event

Each message is described below.

22.5.1 GM_QUIT

This message terminates the GUI task.

Variable	Description
usMessage	GM_QUIT
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used: Specify "0."
lUParam	Not used: Specify "0."

22.5.2 GM_TIMER

This message notifies of a timer event. During the procedure, the timer event ID specified with GESetTimer() is specified.

Variable	Description
usMessage	GM_TIMER
pvTarget	Destination object pointer (NULL: screen)
lLParam	Timer event ID specified with SEGetTimer()
lUParam	Not used: Specify "0."

22.5.3 GM_CHAR

This message sends character codes.

Variable	Description
usMessage	GM_CHAR
pvTarget	Destination object pointer (NULL: screen)
lLParam	Character code
lUParam	Not used (keyboard shift status): Specify "0."

22.5.4 GM_LBUTTONPRESS

This message sends the event issued when the left mouse button is pressed, together with the mouse cursor position.

Variable	Description
usMessage	GM_LBUTTONPRESS
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.5 GM_LBUTTONRELEASE

This message sends the event issued when the left mouse button is pressed, together with the mouse cursor position.

Variable	Description
usMessage	GM_LBUTTONRELEASE
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.6 GM_RBUTTONPRESS

This message sends the event issued when the right mouse button is pressed, together with the mouse cursor position.

Variable	Description
usMessage	GM_RBUTTONPRESS
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.7 GM_RBUTTONRELEASE

This message sends the event issued when the right mouse button is pressed, together with the mouse cursor position.

Variable	Description
usMessage	GM_RBUTTONRELEASE
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.8 GM_MBUTTONPRESS

This message sends the event issued when the middle mouse button is pressed, together with the mouse cursor position.

Variable	Description
usMessage	GM_MBUTTONPRESS
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.9 GM_MBUTTONRELEASE

This message sends the event issued when the middle mouse button is released, together with the mouse cursor position.

Variable	Description
usMessage	GM_MBUTTONRELEASE
pvTarget	Destination object pointer (NULL: screen)
lLParam	Not used (state of mouse button): Specify "0."
lUParam	Mouse cursor coordinates (GPoint)

22.5.10 GM_KEYPRESS

This message sends the event issued when the keyboard is pressed, together with the virtual key code.

Variable	Description
usMessage	GM_KEYPRESS
pvTarget	Destination object pointer (NULL: screen)
lLParam	Virtual key code (defined by user). 0x00 to 0x2F: Used by NC DESIGNER
lUParam	Not used (key state): Specify "0."

The following controls and functions use the virtual key code.

- List : GK_UP/GK_DOWN
- Text box : GK_LEFT/GK_RIGHT/GK_BACKSPACE/GK_DELETE
- Focus setting : GK_LEFT/GK_UP/GK_RIGHT/GK_DOWN
- Edit control : GK_LEFT/GK_RIGHT/GK_UP/GK_DOWN /GK_BACKSPACE/GK_DELETE/GK_RETURN

22.5.11 GM_KEYRELEASE

This message sends the event issued when the keyboard is released, together with the virtual key bode.

Variable	Description
usMessage	GM_KEYRELEASE
pvTarget	Destination object pointer (NULL: screen)
lLParam	Virtual key code (defined by user). 0x00 to 0x2F: Used by NC DESIGNER
LUParam	Not used (key status): Specify "0".

22.5.12 GM_USER

This message sends the user-defined event. The user defines the parameter to be sent.

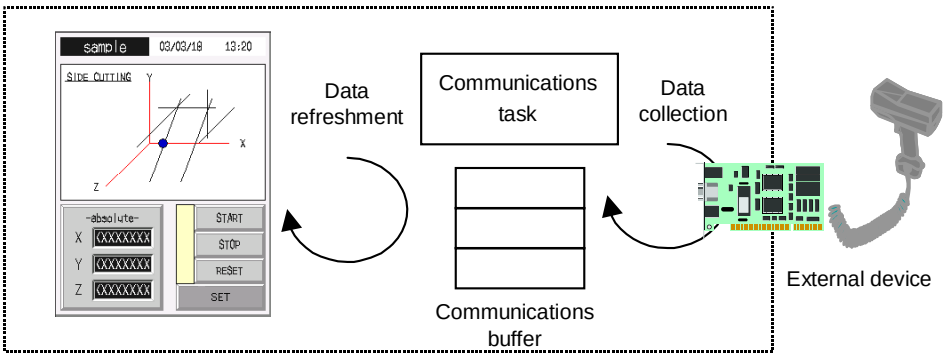
Variable	Description
usMessage	GM_USER
pvTarget	Destination object pointer (NULL: screen)
lLParam	Defined by user
lUParam	Defined by user

22.5.13 GM_SHOWPANEL

Variable	Description
usMessage	GM_SHOWPANEL
pvTarget	Destination object pointer (NULL: screen)
lLParam	Page No. (0 to 255)
lUParam	Not used: Specify "0".

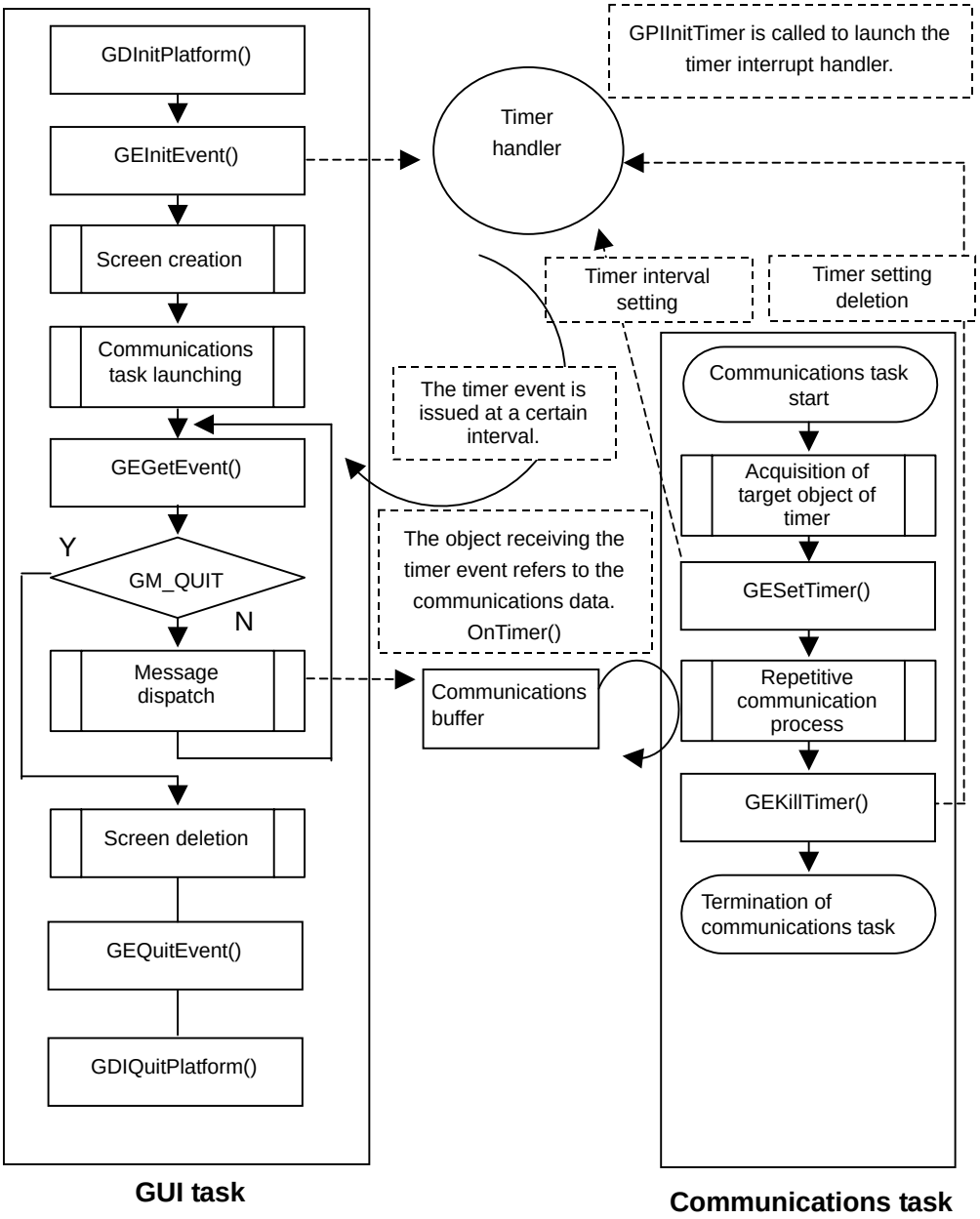
22.6 Timer Event

Description is given here about the timer event process among other events, which is sent from another task. The timer event uses timer interrupt, which is an OS-dependent part of the GPI. In the following description, a communications task is launched as another task and the timer event is used when communication data is acquired at a certain interval from the designated object (control). The timer handler is generated when GEInitEvent() is launched. The communications task specifies the timer event (GESetTimer()) for the target object so that the applicable object keeps receiving the timer event at a preset interval until the timer event is terminated (GEKillTimer()).



NOTE

- Terminate the timer event created by GetSetTimer() without fail, using GEKillTimer(), before the object of the applicable page is deleted.


GDInitPlatform()
GEInitEvent()
Screen creation
Communications task launching
GEGetEvent()
Y
GM_QUIT
N
Message dispatch
Screen deletion
GEQuitEvent()
GDIQuitPlatform()
GUI task
GPIInitTimer is called to launch the timer interrupt handler.
Timer handler
Timer interval setting
Timer setting deletion
The timer event is issued at a certain interval.
The object receiving the timer event refers to the communications data.
OnTimer()
Communications buffer
Communications task start
Acquisition of target object of timer
GESetTimer()
Repetitive communication process
GEKillTimer()
Termination of communications task
Communications task

22.7 User Event

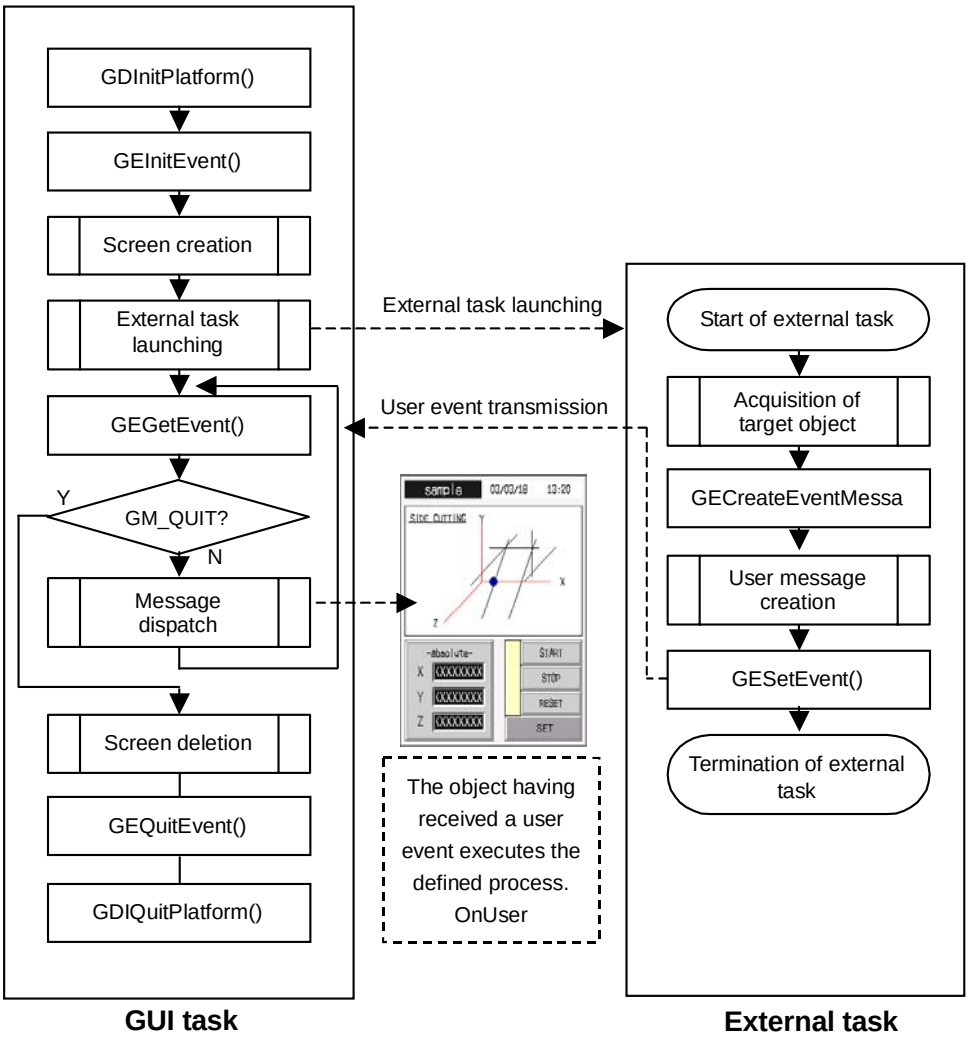
The user event is an original event message defined by the user so that it is received in the application. The user event is received by the callback function (OnUser()) of the application. The following message is sent as a user event.

22.7.1 GM_USER

Variable	Description
usMessage	GM_USER
pvTarget	Destination object pointer (NULL: screen)
lLParam	Defined by user
LUParam	Defined by user

A unique parameter is created so that the GUI task processes the user event from an external task or the like. As a user event message, specify unique IDs in 1LParam and 1UParam and send a message to the GUI task. The GUI task does not send a response message back to the external task.
(Shown below is an example of the user event.)

lUParam = GM_USER_SETVALUE
lUParam = GM_USER_RSTVALUE
lLParam = Setting (value setting)
lLParam = None (value clear)


GDInitPlatform()
GEInitEvent()
Screen creation
External task launching
GEGetEvent()
GM_QUIT?
Y
N
Message dispatch
Screen deletion
GEQuitEvent()
GDIQuitPlatform()
GUI task
External task launching
User event transmission
sample
START
STOP
RESET
SET
The object having received a user event executes the defined process.
OnUser
Start of external task
Acquisition of target object
GECreateEventMessa
User message creation
GESetEvent()
Termination of external task
External task

Appendix

Reference items for the operation of NC Designer are described here. Refer to the description given here when necessary.

Appendix 1. Error Message List

The error messages displayed with NC Designer and remedies are described below.
(A to Z)

	Message	Cause and remedy
A	An invalid character is in a locale name. Please set up an effective character.	This message is displayed if a wrong character is designated during registration of the locale. Use letters, numbers or underscores (_) for the locale name.
	As for Panel/Window information, other users are performing read-out or preservation. Please perform … again after waiting for a while.	This message is displayed if a file has been opened by another user when you read or write the file.
	As for project common information, other users are performing read-out or preservation. Please perform … again after waiting for a while.	This message is displayed if a file has been opened by another user when you read or write the file.
C	Can not create a frame inside a frame.	This message is displayed if a view frame is created inside another view frame. Arrange controls and figures in the view frame.
	Can not create any more frames.	This message is displayed if 10 or more view frames are created in a page. Contain within 10 view frames.
	Can not create any more new screens.	This message is displayed if a new page is created beyond the maximum number of pages at [File] - [New Panel] / [New Window]. Delete unnecessary pages.
	Can not create any more screens.	This message is displayed if a new page is created beyond the maximum number of pages at [Tools] - [Screen Maintenance]. Delete unnecessary pages.
	Can not delete any more nodes.	This message is displayed if the number of vertices of connected lines or a polygon is "3" or fewer at [Layout] - [Modify] - [Delete Node].
	Can not open any more screens. Close other screen to open new screen.	This message is displayed if a new page is opened beyond the number of simultaneously edited pages permitted to NC Designer. The number of pages that can be edited simultaneously is 16 or fewer. Close unnecessary pages.
	Character string is not set.	The message is displayed if no search string is specified at [Edit] - [Find]. Specify the desired search string in the "Find What" field of the [Find] dialog box.
E	End page No. is out of range. Set a number from 0 to 255.	This message is displayed if the last page No. is specified beyond the range when setting the screen range of a process at [Tools] - [Functional Object List] or the like. Specify the screen page number in the range between 0 and 255.
	End page No. is smaller than the start page No.	This message is displayed if the first page number is larger than the last page No. when setting the screen range of a process at [Tools] - [Functional Object List] or the like. Specify a number smaller than the last page number.

	Message	Cause and remedy
F	Failed to do this operation because you are not authorized to edit all of the files in this project. Make sure if there are any other peoples editing these files now.	The general editing right is necessary to execute [Save Project], [Save Project As], [Screen Maintenance] and [Source Code Generation] functions. This message is displayed if acquisition of the general right fails.
I	If you don't save the project data, functional object property may break because the project data contains resource data. Do you want to continue?	This message is displayed if the project is terminated without saving the project data after resource data is changed. Execute [File] - [Save Project], [Save Project As] or [Save All] before terminating the project.
	It is not a numerical value.	This message is displayed if other than a value is entered in the value entry area. Enter an integer within the permissible range in the area.
M	Memory for undoing/redoing operation is insufficient. Increase the free memory by closing other screens or exiting other applications etc.	This message is displayed upon memory shortage. Close other screens or terminate other applications to increase the free memory, then execute again.
	Memory is insufficient. Increase the free memory by closing other screens or exiting other applications etc.	This message is displayed upon memory shortage during source code generation. Close unnecessary applications or take other measures to increase the memory and execute source code generation again.
N	No items are selected.	This message is displayed if no check item is specified at [Tools] - [Error Check]. Select at least one check item and operate again.
O	Other project exists in this directory. Can not create new one.	This message is displayed if an existing project is designated at [File] - [New Project]. Designate another project name.
P	Please set an alphabetic character (a-z, A-Z) to the head of a locale name.	This message is displayed if a wrong character is used during registration of the locale. Specify a one-byte letter (A to Z or a to z) at the top of the locale name.
R	Reading of a file went wrong during import.	The message is displayed if file reading fails during importation of pages. Make sure the project at the import source can be correctly opened.
	Reading of a file went wrong during import. Processing is interrupted.	The message is displayed if the network is shut off during importation of a character string resource from a network file. Restore network connection and import the character string resource again.
	Reading of a file went wrong. The cause below can be considered. 1. The specified file is editing. 2. Network area was specified to be a reading place and the network connection was cut.	This message is displayed if file reading fails during export of a character string resource or import of pages. Refer to the message and remove the cause, then export the character string resource again, or import the pages.
S	Start page No. is out of range. Set a number from 0 to 255.	This message is displayed if the starting page No. is specified beyond the range when setting the screen range of a process at [Tools] - [Functional Object List] or the like. Specify the screen page number in the range from 0 to 255.
T	The following character cannot be used for a file name. \n\\/:;*?"<>|	This message is displayed if a wrong character is designated in the [File Name] at the [Import of a Character Sequence Resource] dialog box. Designate allowable characters for the file name.
	The following character cannot be used for a folder name. \n/:,;*?"<>|	This message is displayed if a wrong character is designated in the [Folder] at the [Import of a Character Sequence Resource] dialog box. Designate allowable characters for the folder name.

	Message	Cause and remedy
T	The input resource data name is already used.	This message is displayed if the registered resource name is already used. Designate another resource data name.
	The number of characters of a folder and a file is to sum total %d character.	Contain the sum of the folder name and file name characters within 200 at the [Export of a Character Sequence Resource] dialog box.
	The number of functional object will exceed the maximum limit. Operation cancelled.	This message is displayed if the object is created beyond the maximum number of objects that can be created in a single page of the page (excl. that in view frame)/view frame Delete unnecessary objects.
	The numerical range is from XXX to XXX.	This message is displayed if an excessive value is entered or the field is left blank at a value entry area. Enter an integer within the permissible range in the area.
	The project already exists. Choose another project name please.	This message is displayed if an existing project is designated at [File] - [New Project]. Designate another project name.
	The project which is going to open cannot be read because of the project edited by NC Designer of a version newer than NC Designer under execution.	Use NC Designer of the version used to create the project.
	The registered resources have exceeded the maximum limit of 5000. Change the import settings.	This message is displayed if the resource is created beyond the maximum number of resources that can be registered in one project. Delete unnecessary resources.
	The specified resource data name already exists. Please enter a unique resource data name.	The multiple resource data under the same name can not exist in one project. Enter the resource data name which does not overlap with the other names.
	The specified file name is unusual.	This message is displayed if there is no drive specification for the folder specified at the [New Project Wizard] or [Import of a Character Sequence Resource] dialog box. Designate the folder together with the full path or, to designate a network folder, designate the drive, too.
	The specified locale name is already used.	This message is displayed if the registered locale has been registered. Specify an unused locale name.
	The specified title already exists. Please enter a unique title.	The multiple titles under the same name can not exist in one project. Enter the resource data name which does not overlap with the other names.
	The setting range of X coordinate is between 0 and 2559 and Y coordinate is between 0 and 1919.	This message is displayed if the X coordinate or the Y coordinate is located out side of the setting range when setting the start position of the cursor. Set the coordinates within the setting range.
	The writing of a file went wrong. The following causes can be considered. 1. The specified file is read-only 2. The free memory of a disk is insufficient. 3. Network area was specified to be a writing place and the network connection was cut.	This message is displayed if file writing fails during exportation of a character string resource. Refer to the message and remove the cause, then export the character string resource again.
	The writing of a property went wrong.	The message is displayed if file reading fails during import of pages. Make sure the file at the import destination is writable.
	The X coordinate is not set.	This message is displayed if the X coordinate has not been set when designating the display start position of the control in which the sub cursor will be displayed. Set the X coordinate.

	Message	Cause and remedy
	The Y coordinate is not set.	This message is displayed if the Y coordinate has not been set when designating the display start position of the control in which the sub cursor will be displayed. Set the Y coordinate.
	Total number of functional object will exceed the limit. Can not continue operation.	This message is displayed if the object is created beyond the maximum number of objects that can be created in a single page (incl. that in view frame). Delete unnecessary objects.

Appendix 2. Shortcut Key List

The shortcut keys that can be used with NC Designer are shown below.

Menu	Function	Shortcut key
File	New Panel	Ctrl + N
	Open Panel/Window	Ctrl + O
	New Window	Ctrl + Shift + N
	Save Panel/Window	Ctrl + S
Edit	Undo	Ctrl + Z
	Redo	Ctrl + Y
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	Delete
	Find	Ctrl + F
	Edit of a caption	Space (for control selection only)
	All Objects	Ctrl + A
	Same Object Type	Ctrl + D (for control/ figure selection only)
View	Previous Page	Shift + PageUp
	Next Page	Shift + PageDown
	Previous Frame Page	PageUp (for frame selection only)
	Next Frame Page	PageDown (for frame selection only)
	Refresh	F9
Layout	Up	↑ (for control selection only)
	Down	↓ (for control selection only)
	Left	← (for control selection only)
	Right	→ (for control selection only)
	Group	Ctrl + G (for selection of multiple controls only)
	Ungroup	Ctrl + U(for group selection only)
Tools	Error check	Ctrl + E
	Functional Object List	Ctrl + L
	Test	Ctrl + T

Appendix 3. About NC Designer

The version of NC Designer is displayed.

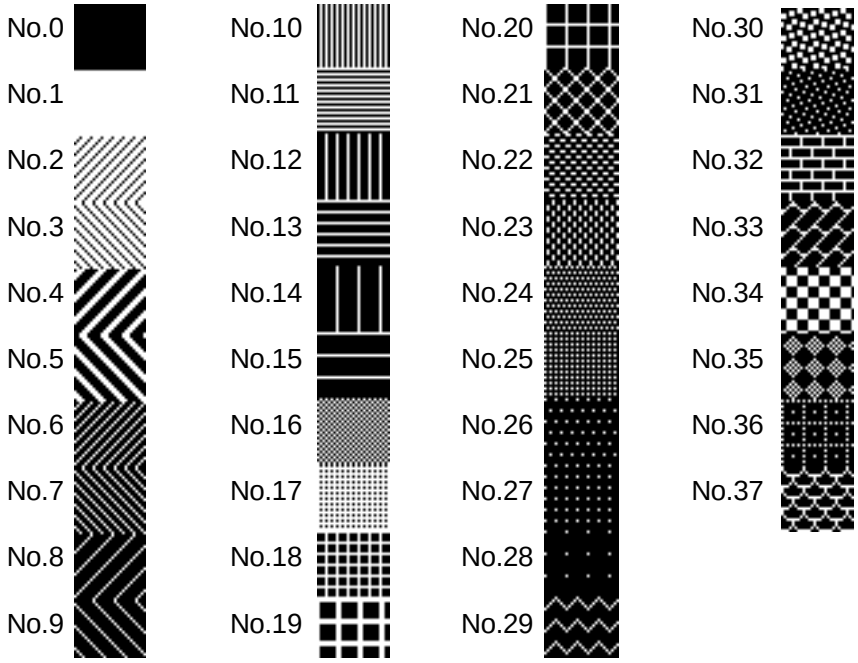
1. Select [About NC Designer] from the [Help] menu.

2. The [About NC Designer] dialog box is displayed.
   Click on the [OK] button to close the dialog box.

Appendix 4. Pattern List

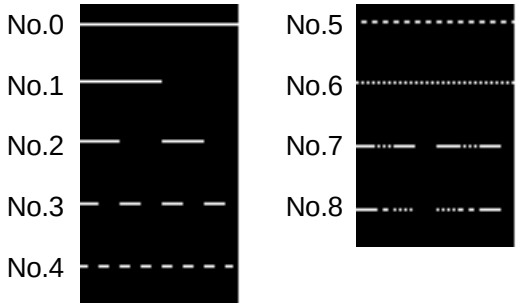
Appendix 4.1 Filling Pattern List

The standard version of NC Designer supports the following 38 filling patterns.



Appendix 4.2 Line Pattern List

The standard version of NC Designer supports the following 9 line patterns.



Appendix 5. Default Palette Color

The standard version of NC Designer supports the following palette.

Color No.	RGB	Color No.	RGB	Color No.	RGB	Color No.	RGB
0	(0,0,0)	32	(44,0,0)	64	(0,0,44)	96	(86,255,44)
1	(0,0,128)	33	(86,0,0)	65	(44,0,44)	97	(135,255,44)
2	(0,128,0)	34	(135,0,0)	66	(86,0,44)	98	(192,255,44)
3	(0,128,128)	35	(192,0,0)	67	(135,0,44)	99	(255,255,44)
4	(128,0,0)	36	(0,44,0)	68	(192,0,44)	100	(0,0,86)
5	(128,0,128)	37	(44,44,0)	69	(255,0,44)	101	(44,0,86)
6	(128,128,0)	38	(86,44,0)	70	(0,44,44)	102	(86,0,86)
7	(192,192,192)	39	(135,44,0)	71	(44,44,44)	103	(135,0,86)
8	(128,128,128)	40	(192,44,0)	72	(86,44,44)	104	(192,0,86)
9	(0,0,255)	41	(255,44,0)	73	(135,44,44)	105	(255,0,86)
10	(0,255,0)	42	(0,86,0)	74	(192,44,44)	106	(0,44,86)
11	(0,255,255)	43	(44,86,0)	75	(255,44,44)	107	(44,44,86)
12	(255,0,0)	44	(86,86,0)	76	(0,86,44)	108	(86,44,86)
13	(255,0,255)	45	(135,86,0)	77	(44,86,44)	109	(135,44,86)
14	(255,255,0)	46	(192,86,0)	78	(86,86,44)	110	(192,44,86)
15	(255,255,255)	47	(255,86,0)	79	(135,86,44)	111	(255,44,86)
16	(192,220,192)	48	(0,135,0)	80	(192,86,44)	112	(0,86,86)
17	(166,202,240)	49	(44,135,0)	81	(255,86,44)	113	(44,86,86)
18	(255,251,240)	50	(86,135,0)	82	(0,135,44)	114	(86,86,86)
19	(160,160,164)	51	(135,135,0)	83	(44,135,44)	115	(135,86,86)
20	(32,192,192)	52	(192,135,0)	84	(86,135,44)	116	(192,86,86)
21	(64,192,192)	53	(255,135,0)	85	(135,135,44)	117	(255,86,86)
22	(96,192,192)	54	(0,192,0)	86	(192,135,44)	118	(0,135,86)
23	(128,192,192)	55	(44,192,0)	87	(255,135,44)	119	(44,135,86)
24	(192,32,192)	56	(86,192,0)	88	(0,192,44)	120	(86,135,86)
25	(192,64,192)	57	(135,192,0)	89	(44,192,44)	121	(135,135,86)
26	(192,96,192)	58	(192,192,0)	90	(86,192,44)	122	(192,135,86)
27	(192,128,192)	59	(255,192,0)	91	(135,192,44)	123	(255,135,86)
28	(192,192,32)	60	(44,255,0)	92	(192,192,44)	124	(0,192,86)
29	(192,192,64)	61	(86,255,0)	93	(255,192,44)	125	(44,192,86)
30	(192,192,96)	62	(135,255,0)	94	(0,255,44)	126	(86,192,86)
31	(192,192,128)	63	(192,255,0)	95	(44,255,44)	127	(135,192,86)

Color No.	RGB	Color No.	RGB	Color No.	RGB	Color No.	RGB
128	(192,192,86)	160	(0,192,135)	192	(86,135,192)	224	(44,135,255)
129	(255,192,86)	161	(44,192,135)	193	(135,135,192)	225	(86,135,255)
130	(0,255,86)	162	(86,192,135)	194	(192,135,192)	226	(135,135,255)
131	(44,255,86)	163	(135,192,135)	195	(255,135,192)	227	(192,135,255)
132	(86,255,86)	164	(192,192,135)	196	(0,192,192)	228	(255,135,255)
133	(135,255,86)	165	(255,192,135)	197	(44,192,192)	229	(0,192,255)
134	(192,255,86)	166	(0,255,135)	198	(86,192,192)	230	(44,192,255)
135	(255,255,86)	167	(44,255,135)	199	(135,192,192)	231	(86,192,255)
136	(0,0,135)	168	(86,255,135)	200	(255,192,192)	232	(135,192,255)
137	(44,0,135)	169	(135,255,135)	201	(0,255,192)	233	(192,192,255)
138	(86,0,135)	170	(192,255,135)	202	(44,255,192)	234	(255,192,255)
139	(135,0,135)	171	(255,255,135)	203	(86,255,192)	235	(44,255,255)
140	(192,0,135)	172	(0,0,192)	204	(135,255,192)	236	(86,255,255)
141	(255,0,135)	173	(44,0,192)	205	(192,255,192)	237	(135,255,255)
142	(0,44,135)	174	(86,0,192)	206	(255,255,192)	238	(192,255,255)
143	(44,44,135)	175	(135,0,192)	207	(44,0,255)	239	(24,24,24)
144	(86,44,135)	176	(192,0,192)	208	(86,0,255)	240	(37,37,37)
145	(135,44,135)	177	(255,0,192)	209	(135,0,255)	241	(52,52,52)
146	(192,44,135)	178	(0,44,192)	210	(192,0,255)	242	(68,68,68)
147	(255,44,135)	179	(44,44,192)	211	(0,44,255)	243	(77,77,77)
148	(0,86,135)	180	(86,44,192)	212	(44,44,255)	244	(95,95,95)
149	(44,86,135)	181	(135,44,192)	213	(86,44,255)	245	(105,105,105)
150	(86,86,135)	182	(192,44,192)	214	(135,44,255)	246	(114,114,114)
151	(135,86,135)	183	(255,44,192)	215	(192,44,255)	247	(125,125,125)
152	(192,86,135)	184	(0,86,192)	216	(255,44,255)	248	(146,146,146)
153	(255,86,135)	185	(44,86,192)	217	(0,86,255)	249	(157,157,157)
154	(0,135,135)	186	(86,86,192)	218	(44,86,255)	250	(168,168,168)
155	(44,135,135)	187	(135,86,192)	219	(86,86,255)	251	(180,180,180)
156	(86,135,135)	188	(192,86,192)	220	(135,86,255)	252	(204,204,204)
157	(135,135,135)	189	(255,86,192)	221	(192,86,255)	253	(216,216,216)
158	(192,135,135)	190	(0,135,192)	222	(255,86,255)	254	(229,229,229)
159	(255,135,135)	191	(44,135,192)	223	(0,135,255)	255	(242,242,242)

Appendix 6. 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 No.                                          */
                                                        /* 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 No.                                                 */
                                                        /* 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 7. Addition of Original User Event

Change the configuration file corresponding to each control to change presence/absence of the callback function of each control. In addition, you can add original callback functions.

The configuration file is stored in the following folder.
\(folder where NC DESIGNER is installed)\BLDPARTS
If NC Designer has been installed by default by system environment, it will be stored in the following folder:
For Windows 2000/XP
C:\Program Files\NC Designer\BLDPARTS
For Windows Vista/7
C: \MITSUBISHI CNC\MELSOFT\NC Designer\BLDPARTS

The configuration file of each control is specified below.

Control	File name
Basic control	GBasicControl.ini
Button	GButton.ini
Text box	GTextBox.ini
Label	GLabel.ini
List	GList.ini
Picture	GPicture.ini
Check box	GCheckBox.ini
Radio button	GRadioButton.ini
Progress bar	GProgressBar.ini
HTML browser	GHtmlBrowser.ini
Vertical scroll bar	GVScrollBar.ini
Horizontal scroll bar	GHScrollBar.ini
Edit control	GEdit.ini
Input box	GInputBox.ini
Ten-key	GSoftKey.ini
NC data text box	GNCDataTextBox.ini
PLC button	GNCPLCButton.ini
PLC extension button	GNCPLCExButton.ini
PLC message	GNCPLCMessage.ini
Table	GNCTable.ini
Counter	GNXCounter.ini
CycleTime	GNXCycleTime.ini
Feedrate	GNXFeedrate.ini
GModal M	GNXGModal.ini
GModal L	GNXGModal_L.ini
GModal Simple	GNXGModalSimple.ini
LoadMeter	GNXLoadMeter.ini
MSTB	GNXMSTB.ini
ONB	GNXONB.ini
ProgramBuffer	GNXPrgBuff.ini
SPCommand	GNXSPCommand.ini
Menu	GNXMenu.ini
FileInOut	GNXFileInOut.ini
AlarmMessage	GNXAlarmMessage.ini
Monitor	GNXMonitorStatus.ini
Time	GNXTime.ini

The setting of each file is shown below.

Key	Section	Description
OnKeyPress	DefaultCallBack	Specify whether OnKeyPress callback is provided or not. 0: Not provided. Other than 0: Provided.
OnKeyRelease	DefaultCallBack	Specify whether OnKeyRelease callback is provided or not. 0: Not provided. Other than 0: Provided.
OnPress	DefaultCallBack	Specify whether OnPress callback is provided or not. 0: Not provided. Other than 0: Provided.
OnRelease	DefaultCallBack	Specify whether OnRelease callback is provided or not. 0: Not provided. Other than 0: Provided.
OnClick	DefaultCallBack	Specify whether OnClick callback is provided or not. 0: Not provided. Other than 0: Provided.
OnDraw	DefaultCallBack	Specify whether OnDraw callback is provided or not. 0: Not provided. Other than 0: Provided.
OnTimer	DefaultCallBack	Specify whether OnTimer callback is provided or not. 0: Not provided. Other than 0: Provided.
OnSetFocus	DefaultCallBack	Specify whether OnSetFocus callback is provided or not. 0: Not provided. Other than 0: Provided.
OnKillFocus	DefaultCallBack	Specify whether OnKillFocus callback is provided or not. 0: Not provided. Other than 0: Provided.
OnCreate	DefaultCallBack	Specify whether OnCreate callback is provided or not. 0: Not provided. Other than 0: Provided.
OnDelete	DefaultCallBack	Specify whether OnDelete callback is provided or not. 0: Not provided. Other than 0: Provided.
OnUser	DefaultCallBack	Specify whether OnUser callback is provided or not. 0: Not provided. Other than 0: Provided.
OnScroll	DefaultCallBack	Specify whether OnScroll callback is provided or not. 0: Not provided. Other than 0: Provided.

Key	Section	Description
OnScrollFinish	DefaultCallBack	Specify whether OnScrollFinish callback is provided or not. 0: Not provided. Other than 0: Provided.
OnSelectChange	DefaultCallBack	Specify whether OnSelectChange callback is provided or not. 0: Not provided. Other than 0: Provided.
CallBackEventNum	ExtendCallBack	Specify the number of callback events to be additionally registererd. (0 to 32)
CallBackEventNameXXX (XXX=000 to 032)	ExtendCallBack	Specify the callback event name to be added (within 32 characters). The first character must be a single byte letter (A to Z or a to z). The second and later characters must be single byte letters, numbers or underscores (_).
CallBackEventIDXXX (XXX=000 to 032)	ExtendCallBack	Specify the event functioning as a distribution key of the callback event.
CallBackPropertyIDXXX (XXX=000 to 032)	ExtendCallBack	Specify the identifier of the callback event property (within 32 characters). The first character must be a single byte letter (A to Z or a to z). The second and later characters must be single byte letters, numbers or underscores (_) or space.

NOTE

- The callback functions, for which "not provided" is specified in the property window during selection of each control at NC Designer, are not displayed.
- If a wrong character or an excessive number is specified at "CallBackEventNum," setting "0" is assumed.
- If no specification is given at "CallBackEventNameXXX," "CallBackEventIDXXX" or "CallBackPropertyIDXXX," the callback function is not displayed in the property window.
- Specify "CallBackEventIDXX" so that there is no duplicate ID.

Create a user original callback function in the following way.

In case of C++ language
Function name:Control name (all in upper case) + callback event name
Argument:unsigned short usMessage; //Callback event message
unsigned long lUParam; //Additional data
unsigned long lLParam; //Return data
Return value:long

Appendix 8. HTML Tag List

NC Designer supports the following HTML tags.

	Function	Tag name
Document	Document structure definition	<html> - </html>, <head> - </head>,<body> - </body>
TEXT	Title setting	<hxxx> - </hxxx>
	Paragraph setting	-
	Carriage return	
	Long sentence quotation	<blockquote> - </blockquote>
	Emphasis	- , -
	Superscript and subscript designation	⁻, ₋
	Text direction designation	<bdo dir="xxx"> - </bdo>
PAGE	Background color designation	<body bgcolor="xxx"> - </body>
	Background image designation	<body background="xxx "bgproperties="fixed"> - </body>
	Text color designation	<body text="xxx"> - </body>, <body link="xxx"> - </body>
	Partial text color designation	 -
	Title position designation	<hxxx align="yyy"> - </hxxx>
	Paragraph position designation	-
	Designation of position of designated range	-
	Centering	-
	Horizontal ruler display	<hr>,<hr xxx>
	Horizontal ruler color designation	<hr color="xxx">
	Page margin designation	<body xxx> - </body>
FONT	Absolute font size designation	 -
	Relative font size designation, pattern 1	 -
	Relative font size designation, pattern 2	<basefont size="xxx">, -
	Font type designation	 -
	Font style designation, pattern 1	- ,<I> - </I>, <strike> - </strike>, <s> - </s>,<tt> - </tt>,<u> - </u>
	Font style designation, pattern 2	<big> - </big>, -
	Special character display	& number;, & key word
LIST	Numbered list creation	-
	List mark change	<ul type="xxx"> - , <li type="xxx"> -
	Numbered list mark change	<ol type="xxx"> - , <li type="xxx"> -
	List starting number change	<ol start="xxx"> -
	Serial list number change	<li value="xxx"> -
	Term definition list display	<dl><dt> - </dl><dd> - </dd></dl>, <dl compact><dt> - </dl><dd> - </dd></dl>

	Function	Tag name
LIST	List mark change	<ul type="xxx"> - , <li type="xxx"> -
	Numbered list mark change	<ol type="xxx"> - , <li type="xxx"> -
IMAGE	Image display	<img src="xxx">
	Image size designation	<img src="xxx" width="yyy" height="zzz">
	Designation of arrangement with text	<img src="xxx" align="yyy">
	Arrangement of text around image	<img src="xxx" align="yyy">
	Resetting of arrangement around image	<br clear="xxx">
	Image-to-text gap designation	<img src="xxx" vspace="yyy" hspace="zzz">
TABLE	Table creation	<table "xxx"> - </table>,<tr> - </tr>, <td> - </td>
	Table title creation	<th> - </th>
	Designation of table position to text	<table align="xxx"> - </table>
	Resetting of arrangement around table	<br clear="xxx">
	Table size designation	<table width="xxx" height="yyy"> - </table>
	Cell size designation	<th width="xxx" height="yyy"> - </th>, <td width="xxx" height="yyy"> - </td>
	Caption	<caption> - </caption>, <caption align="xxx"> - </caption>
	Vertical cell merge	<th rowspan="xxx"> - </th>, <td rowspan="xxx"> - </td>
	Horizontal cell merge	<th colspan="xxx"> - </th>

* "xxx", " yyy" and "zzz" are parameters specified as a tag.

NOTE

- Tags not found in the list are not supported. Unsupported tags are ignored when the file is displayed.
- The supported image format is jpeg only; other image file formats are not displayed.
- The upper limit of the HTML file size is 100K bytes.

Appendix 9. Executing File Registration Method

Appendix 9.1 F0 Release

To register the executing file data to the function key, it is necessary to edit melAppCtrl.ini.
Create the registered executing file by full-screen as much as possible.

Appendix 9.1.1 melAppCtrl.ini

Example of setting

```
; When the Shift+F10 key (F0 key) is pressed, the calculator is started.
[Program00]
VirtualKey=VK_F10
KeyData=VK_SHIFT
Command00=Execute,C:\WINDOWS\system32\calc.exe,the calculator,,0,0
```

Refer to Appendix 9.1.2 for details of items.

Appendix 9.1.2 Details of melAppCtrl.ini

The configuration file (melAppCtrl.ini) conforms to the description format of the Windows.INI file in principle, and the upper bound of the maximum INI file size is 32KByte.

The description format of Windows.INI

```
[(Section)]
(Key) = (Value of key)
        :
[(Section)]
(Key) = (Value of key)
        :
```

Edit the following item of the [General] section.

Section name	Details	Initial value (when undefined)
GENERAL	Specify general system requirements. A set number of the following "StartUp" section and "Program" section is set.	-

Section name	Details	Initial value (when undefined)
StartUpCount	Specify the set number of the [StartUp] section. Setting range : 0 to 5	0
ProgramCount	Specify the set number of the [Program] section. Setting range : 0 to 10	0

Edit the following item of the [StartUp] section.

Section name	Details	Initial value (when undefined)
StartUp**	Specify the executing file information when the power supply is turned ON. To ** of section name, specify the sequential No. (0 to) until the values in which 1 is subtracted from the [StartUp] key setting value of [General] section.	No setting

Section name	Details	Initial value (when undefined)
CommandCount	Specify the number of executing commands. Setting range : 0 to 10	1
Command**	Specify the executing commands. The command is executed in ascending order of the No. set to **. To **, specify the sequential No. (0 to) until the values in which 1 is subtracted from the [CommandCount] key setting value. Refer to "Command list" for the command which can be specified. Setting range : Number of commands of maximum characters = 256 byte	NULL (No commands)

Edit the following item of the [Program] section.

Section name	Details	Initial value (when undefined)
ControlParam	Designate a parameter to control the execution of commands. Setting range :1(Standard screen), 100 to 199(Custom screen)	1：Standard screen
Program**	Specify the executing file corresponding to the input key. To ** of section name, specify the sequential No. (0 to) until the values in which 1 is subtracted from the [Program] key setting value of [General] section.	No setting

Section name	Details	Initial value (when undefined)
VirtualKey	Specify the key code of operation board to register the execution file. The key code which can be set is as follows. **Key name** / **Virtual key code** F1 to F12 / VK_F1 to F12 '0' to '9' / VK_0 to VK_9 'A' to 'Z' / VK_A to VK_Z	No setting
KeyData	Specify the Shift/Ctrl/Alt key that combines with above-mentioned "VirtualKey" and is input. This key can be defined by combining two or more key codes. (In that case, insert a space between the key codes.) Ex.) The Shift/Ctrl key was pressed at the same time KeyData = VK_SHIFT VK_CONTROL The key code which can be set is as follows. **Key name** / **Virtual key code** Shift / VK_SHIFT Ctrl / VK_CONTROL Alt / VK_MENU	No setting
CommandCount	Specify the number of executing commands. Setting range : 1 to 10	1

Section name	Details	Initial value (when undefined)
Command**	Specify the executing commands. The command is executed in ascending order of the No. set to **. To **, specify the sequential No. (0 to) until the values in which 1 is subtracted from the [CommandCount] key setting value. Refer to "Command list" for the command which can be specified. Setting range : Number of commands of maximum characters = 256 byte	NULL (No commands)

Command list

The command set with the "Command**" key is set as a comma-deliminated character string (command character string).

Format for command character string is shown below.

(Command name), (1st argument), (2nd argument), (3rd argument)…

When the numerical value is set to the argument, the value is processed **as a hexadecimal number when the character of "0x" is added to the head.** Other numerical values are processed as a decimal number.

<table>
<tr><th>Command name</th><th>Details</th></tr>
<tr><td>Execute</td><td>Start the designated executing file.

The starting status of the executing file can be judged by setting the title bar character string and the class name of the window. As a result, a multiple start of the execution file can be controlled.
If both the title bar character string and the class names of the window are set, it is judged "The execution file is starting" when each requirement is satisfied at the same time.

When the executing file has already started, the focus is set to the corresponding executing file, and the order of displaying the window is changed to most significant. This can be invalidated by the setting of the 4th argument and the 5th argument.)

The detail of argument is as shown below.
<table>
<tr><th>Argument</th><th>Details</th></tr>
<tr><td>1st argument</td><td>The file name of the starting executing file (including the folder name)</td></tr>
<tr><td>2nd argument</td><td>The title bar character string of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged as the unsetting.</td></tr>
<tr><td>3rd argument</td><td>The class name of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged as the unsetting.</td></tr>
<tr><td>4th argument</td><td>Presence of focus control

0 : Move focus to the executing file.
-1 : Do not control the focus.

When the value is not specified, it is judged the setting value is 0.</td></tr>
<tr><td>5th argument</td><td>Presence of window position control

0 : Display the window in most significant.
1 : Display the window in least significant.
-1 : Do not control the window position.</td></tr>
</table>
[Restrictions]
When two or more executing files matched to the condition exist, the executing file found first is operated.</td></tr>
</table>

<table>
<tr><th>Command name</th><th>Details</th></tr>
<tr><td>Sleep</td><td>Stop the command execution only at specified time (ms).

The detail of argument is as shown below.
<table>
<tr><th>Argument</th><th>Details</th></tr>
<tr><td>1st argument</td><td>Stop time (ms)</td></tr>
</table>
</td></tr>
<tr><td>Exit</td><td>End the designated executing file.

The ending executing file can be selected by setting the title bar character string and the class name of the window.

If both the title bar character string and the class names of the window are set, the executing file which satisfies each requirement at the same time is ended.

Only when the executing file has already started, the corresponding executing file is ended.

The detail of argument is as shown below.
<table>
<tr><th>Argument</th><th>Details</th></tr>
<tr><td>1st argument</td><td>The title bar character string of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged as the unsetting.</td></tr>
<tr><td>2nd argument</td><td>The class name of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged as the unsetting.</td></tr>
</table>
[Restrictions]]
When two or more executing files matched to the condition exist, the executing file found first is operated.</td></tr>
</table>

<table>
<tr><th>Command name</th><th>Details</th></tr>
<tr><td>PostMessage</td><td>Send the Windows messages to the designated executing file.

The executing file sent the Windows messages can be selected by setting the title bar character string and the class name of the window.

If both the title bar character string and the class names of the window are set, the Windows messages are send to the executing file which meets each requirement at the same time.

Only when the executing file has already started, the Windows messages specified for the corresponding executing file are sent.

The detail of argument is as shown below.
<table>
<tr><th>Argument</th><th>Details</th></tr>
<tr><td>1st argument</td><td>The title bar character string of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged the unsetting.</td></tr>
<tr><td>2nd argument</td><td>The class name of window referred to confirm the starting status of the executing file

Wild-card (*) can be specified in the character string.

When the character string is not specified, it is judged the unsetting.</td></tr>
<tr><td>3rd argument</td><td>Message ID of sent Windows messages</td></tr>
<tr><td>4th argument</td><td>Argument 1 (wParam) of sent Windows message</td></tr>
<tr><td>5th argument</td><td>Argument 2 (lParam) of sent Windows message</td></tr>
</table>
[Restrictions]
When two or more executing files matched to the condition exist, the executing file found first is operated.</td></tr>
</table>

Appendix 9.1.3 Function of Update Cycle Setting

When the executing file registered by F0 release is started, update cycle of the standard screen which is operated on the background can be set by melAppCtrl.ini. As a result, the time that the starting execution file occupies CPU can be extended.
When a standard screen moves to an active screen, the setting of the update cycle (sleep time) is released.

<The argument of sleep time setting PostMessage>

Argument	Setting value
1st argument	Mitsubishi CNC HMI (fixed)
2nd argument	Mitsubishi CNC HMI Class (fixed)
3rd argument	0x500(fixed)
4th argument	Specify the update cycle time of a standard screen when the custom application is started. When the registered custom application is started, a standard screen is updated at the set update cycle only. It is possible to set "Do not update", "Release setting" or "until 0 to 1 second by each millisecond unit". Setting rang : -1 to 1000 -1 : Do not update. 0 : Release the setting. 1 to 1000 : Changes to the set update cycle. Out of range : Do not change the update cycle.
5th argument	0 (fixed)

Refer to Appendix 9.1.2 for details of PostMessage.

Setting example of melAppCtrl

```
; The calculator is started when the F0 key is pressed, and the update cycle of a standard screen is set to 500ms.
[Program00]
ControlParam=1
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=2
Command00=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI Class,0x500,500,0
Command01=Execute,C:\WINDOWS\SYSTEM32\calc.exe,calculator,,
```

Update at 500ms cycle

NOTE

- When an application is switched by Alt + Tab, the update cycle changing function is not applied.
- Whether the start of the execution file succeeded is not checked. Even if the start of the executing file is failed, the update cycle of a standard screen is changed.
- The start time of the registered executing file can be shortened by executing the PostMessage command ahead of the Execute command.
- The update cycle setting is released by moving the standard screen to the background screen once, and displaying the standard screen on an active screen again. Note that the setting is not released in the display on an active screen.

Appendix 9.2 Menu Release

An arbitrary executing file can be registered in the main menu.
A standard screen can be operated with the executing file started.
To register the executing file registration method data to the main menu, it is necessary to edit Config.ini, and to prepare the icon image and the executing file for registration.
The customdef.ini has to be described by UNICODE text.
Create the registered executing file by full-screen as much as possible.

Appendix 9.2.1 customdef.ini

Refer to 15.6.5.1.2.

Appendix 9.2.2 Icon Image

Refer to 15.6.5.1.3.

Appendix 9.3 Using a function key (screen switching key) in the custom screen

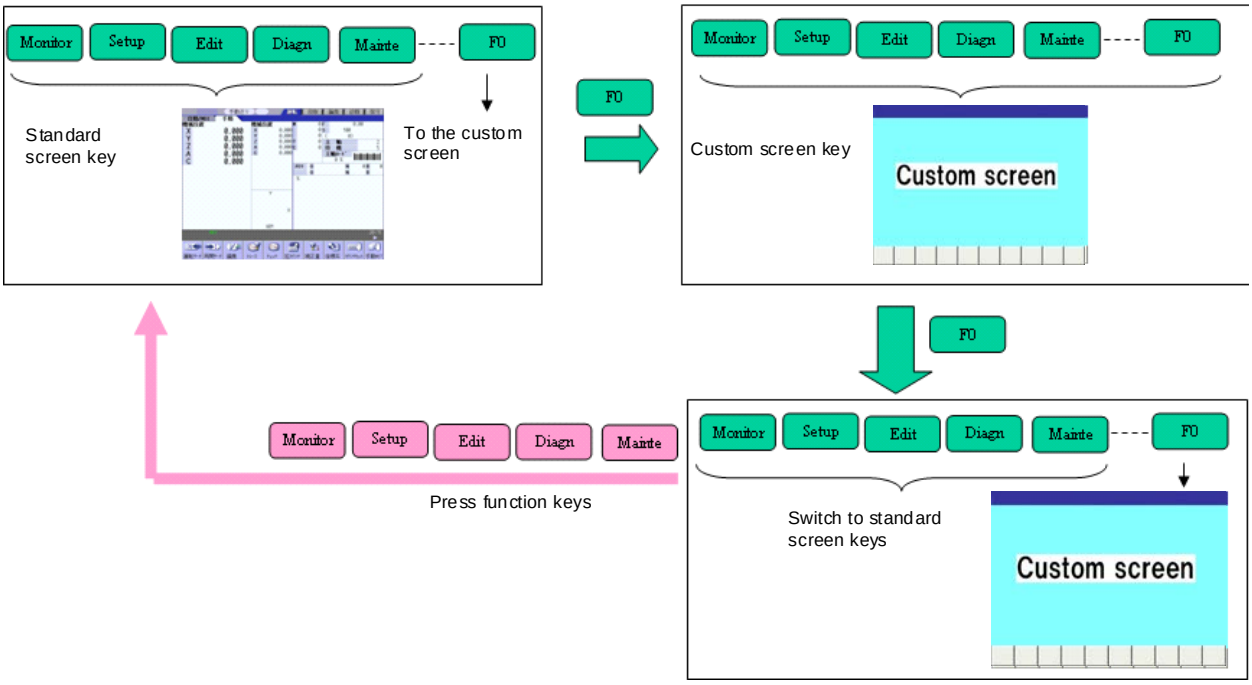
The function keys for switching the standard screen, such as [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE], can be used as a screen switching key and operation key in the custom screen for the executing file registration method custom screen.
To use the function keys mentioned above as keys in the custom screen, execute the setting to switch the CotrolParam key in the Program section. There are three specific methods to switch the ControlParam key:
1) Utilization method 1 by adding a definition to melAppCtrl.ini (only the meaning of function key can be changed with "F0" key)
2) Utilization method 2 by adding a definition to melAppCtrl.ini (changing to a specific standard screen with "F0" key)
3) Utilization method 3 by adding a definition to melAppCtrl.ini (changing to the specific standard screen with "Menu" key)
By switching the ControlParam key, the custom screen can receive a function key without a definition in "melAppCtrl.ini". The key allocations are as follows:

Function key	Key that the custom screen receives
MONITOR	SHIFT+F1
SETUP	SHIFT+F2
EDIT	SHIFT+F3
DIAGN	SHIFT+F4
MAINTE	SHIFT+F5

（Compliment）The explanations for specific methods for switching the CotrolParam key are based on when the custom screen of the executing file registration method is allocated to "F0" key.
The same applies to SFP key, Screen display key(), and Screen selection key ().

Appendix 9.3.1 Utilization method 1 by adding a definition to melAppCtrl.ini (only the meaning of function key can be changed with "F0" key)

The below shows the setting example of melAppCtrl.ini in order that, after displaying the custom screen by pressing "F0", the allocated function keys become transition keys to the standard screen while the custom screen is maintained when "F0" is pressed again.
When "F0" is pressed while the standard screen is displayed, the screen changes to the custom screen and then [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become the keys to be used in the custom screen. When "F0" is pressed again, [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become transition keys to the standard screen even the custom screen is still valid. After that, it goes back to the standard screen by pressing [MONITOR], [SETUP], [EDIT], [DIAGN] or [MAINTE] key.



Setting example of "melAppCtrl.ini"

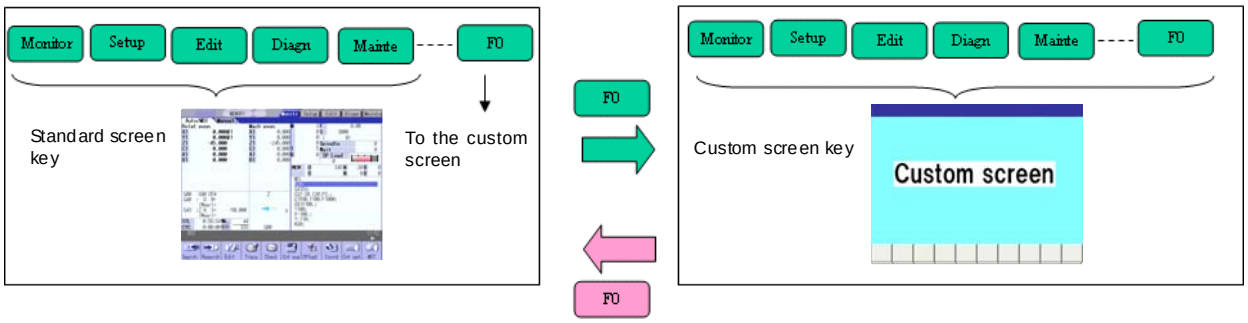
```
[GENERAL]
;Program      The number of sections
ProgramCount=2

;Startup the custom screen
[Program00]
ControlParam=1
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=2
Command00=Execute,d:\Ccustoum\custom.exe,Custom Ttitle,Custom Class,0,0
Command01=PostMessage,Mitsubishi CNC Application Control Task,Mitsubishi CNC Application Control Class,0x400,1,100

;Change the function keys back to the keys for the standard screen (F0 key)
[Program01]
ControlParam=100
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=1
Command00=PostMessage,Mitsubishi CNC Application Control Task,Mitsubishi CNC Application Control Class,0x400,1,1
```

Change "ControlParam" to "100"
Setting range : 100 to 199

Change "ControlParam" back to "1".
* "1" is the standard screen.

Appendix 9.3.2 Utilization method 2 by adding a definition to melAppCtrl.ini (changing to a specific standard screen with "F0" key)

The below shows the setting example of melAppCtrl.ini to change to a specific standard screen by pressing "F0" again after the custom screen is displayed by pressing "F0" key.
When "F0" is pressed while the standard screen is displayed, it changes to the custom screen and then [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become the keys to be used in the custom screen. When "F0" is pressed again, it changes to a specific standard screen (it is "monitor screen" in this example) and then [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become transition keys to the standard screen.



Setting example of "melAppCtrl.ini"

```
[GENERAL]
;Program The number of sections
ProgramCount=2

;Startup the custom screen (F0 key)
[Program00]
ControlParam=1
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=2
Command00=Execute,d:\Ccustoum\custom.exe,Custom Ttitle,Custom Class,0,0
Command01=PostMessage,Mitsubishi CNC Application Control Task,Mitsubishi CNC Application Control Class,0x400,1,100

;Change the function keys back to the keys for the standard screen (F0 key)
[Program01]
ControlParam=100
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=4
Command00=PostMessage,Mitsubishi CNC Application Control Task,Mitsubishi CNC Application Control Class,0x400,1,1
Command01=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI Class,0x500,0,0
Command02=Execute,c:\ncsys\melhmi.exe,Mitsubishi CNC HMI,Mitsubishi CNC HMI Class,0,0
Command03=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI Class,0x401,0x70,0x00000001
```

Change "ControlParam" to "100"
Setting range : 100 to 199

Set ControlParam back to "1".
* "1" is the standard screen.

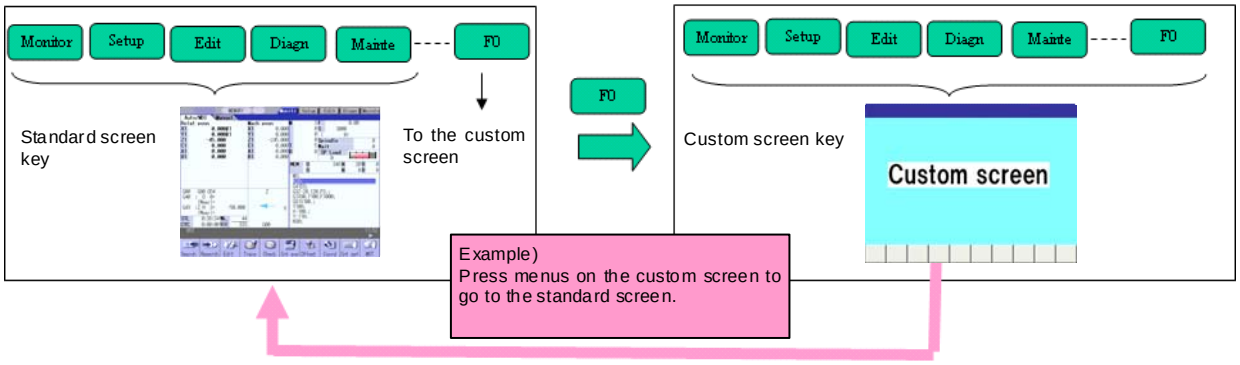
Display the "monitor screen"

Setting for displaying the screen other than the operation screen

```
//Changes to the setup screen
Command03=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI
Class,0x401,0x71,0x00000001
//Changes to the edit screen
Command03=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI
Class,0x401,0x72,0x00000001
//Changes to the diagnosis screen
Command03=PostMessage,Mitsubishi CNC HMI,Mitsubishi CNC HMI
Class,0x401,0x73,0x00000001
```

Appendix 9.3.3 Utilization method 3 by adding a definition to melAppCtrl.ini (changing to the specific standard screen with "Menu" key)

The below shows the setting example of melAppCtrl.ini to change to a specific standard screen by pressing "F0" again after the custom screen is displayed by pressing "F0" key, and the sample source.
When "F0" is pressed while the standard screen is displayed, it changes to the custom screen and then [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become the keys to be used in the custom screen. When Menu is then pressed, it changes to a specific standard screen (it is "monitor screen" in this example) and then [MONITOR], [SETUP], [EDIT], [DIAGN] and [MAINTE] become transition keys to the standard screen. To use this method, it needs to incorporate the processes of switching the ControlParam key from the menu in the custom screen and of changing to the standard screen into the source code for custom screen application.



Setting example of "melAppCtrl.ini"

```
[GENERAL]
;Program The number of sections
ProgramCount=1

;Startup the custom screen (F0 key)
[Program00]
ControlParam=1
VirtualKey=VK_F10
KeyData=VK_SHIFT
CommandCount=2
Command00=Execute,d:\Ccustoum\custom.exe,Custom Ttitle,Custom Class,0,0
Command01=PostMessage,Mitsubishi CNC Application Control Task,Mitsubishi CNC Application Control Class,0x400,1,100
```

Change "ControlParam" to "100"
Setting range：100 to 199

Sample source code（processing example of changing to the operation screen from the menu of the custom application）

```
// Return the operation of function keys to the standard screen (return "ControlParam" to "1")
CWnd* _hmiApphwd = FindWindow("Mitsubishi CNC Application Control Class",
"Mitsubishi CNC Application Control Task");
_hmiApphwd->PostMessage(WM_USER, (WPARAM)1, (LPARAM)1);

// Display the standard screen in the most front
CWnd* _hmihwd = FindWindow("Mitsubishi CNC HMI Class","Mitsubishi CNC HMI");
_hmihwd->SetForegroundWindow();

// Display the monitor screen (Monitor) of the standard screen
LPARAM _lKeyStatus = 0;
_lKeyStatus |= 0x00000001;
_hmihwd->PostMessage((WM_USER+0x0001), VK_F1, lKeyStatus);
```

Appendix 9.3.4 The standard screen after changing

The table below explains the screen display of the standard screen when the custom screen is changed to the standard screen.
If the window is being displayed, it will be closed.
All the menus will return to the main menu. The menu of the displayed screen will be highlighted.

The standard screen display before the custom screen was displayed			The standard screen display after changing from the custom screen
Monitor	Common in all screens		Each screen will be closed
Setup	Compensation amount		Compensation amount
	Tool measurement		Tool measurement
	Tool registration		Tool register
	Tool life		Tool life
	Coordinate system		Coordinate system
	Workpiece measurement		Workpiece measurement
	Pallet		Pallet
	User parameter		Each screen will be closed. Either the compensation amount, tool measurement, tool registration, tool life, coordinate system, workpiece measurement or pallet, whichever was displayed before displaying these screens, will be displayed.
	MDI Editing		
	Counter set		
	Manual operation MST		
	T-list search		
	Tool management		
Edit	Edit		Edit
	Check		Check
	Input/output		Input/Output screen will be closed. Either Edit or Check, whichever was displayed before displaying the input/output screen, will be displayed.
Diagnosis	H/W S/W configuration		H/W S/W configuration
	Option display		Option display
	I/F diagnosis		I/F diagnosis
	Drive monitor		Drive monitor
	NC memory diagnosis		NC memory diagnosis
	Alarm message		Alarm message
	Self diagnosis		Self diagnosis
	NC sampling		NC sampling
	Anshin-net		Anshin-net
	MTB net		MTB net
Maintenance	Maintenance	All backup	All backup
		System setup	System setup
		S-analog adjustment	S-analog adjustment
		Absolute position	Absolute position
		Auxiliary axis test operation	Auxiliary axis test operation
		Servo diagnosis	Servo diagnosis
		Collection setting	Collection setting
		Option setting	Option setting
		Device open parameter	Device open parameter
		SRAM open parameter	SRAM open parameter
	Parameter		Parameter
	Input/output		Input/output

Appendix 10. HMI Integrated Installer

Appendix 10.1 Outline

"HMI integrated installer" is a function to install and upgrade "Application of custom release" and "Application of HMI related" with the memory card and data in the card.

For integrated installer, two methods are prepared to install and upgrade the application.

[Methods of installing and upgrading application]

(1) Method of starting installer (EXE file) for each application prepared beforehand, and installing application
(Thereafter, this method is described "Installer start method".)

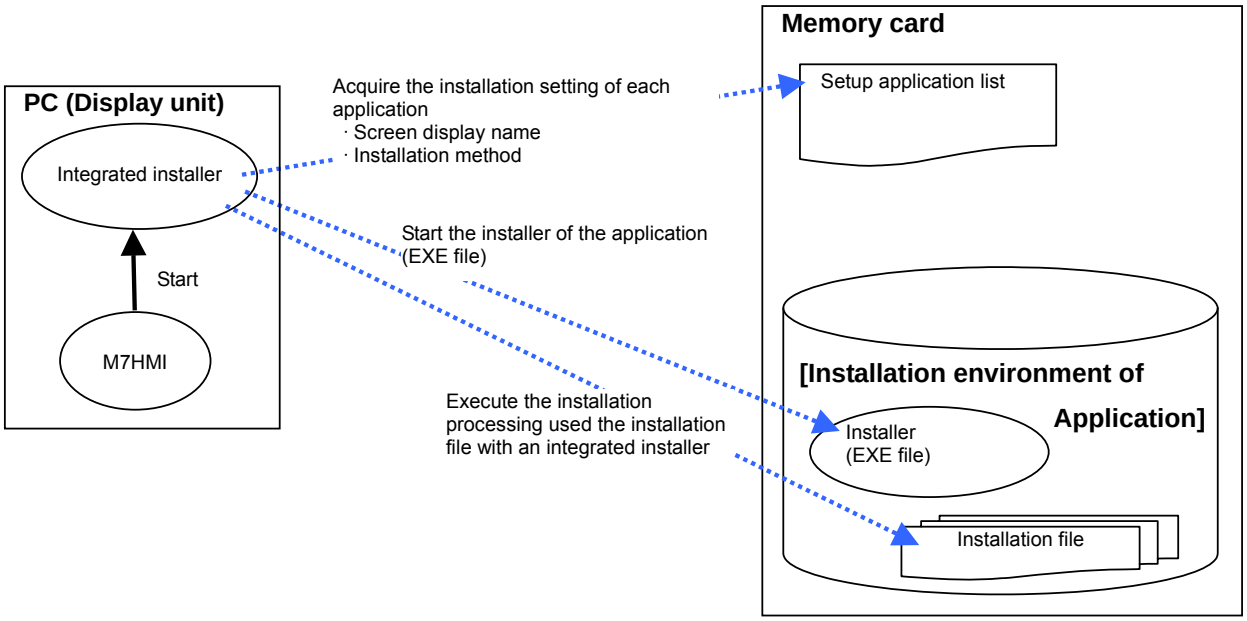
(2) Method of copying, deleting specified file (directory) with integrated installer, setting registry, and installing application
(Thereafter, this method is described "File copy method".)

The following are prepared in the memory card even if which method is used, and then each application is installed and upgraded:

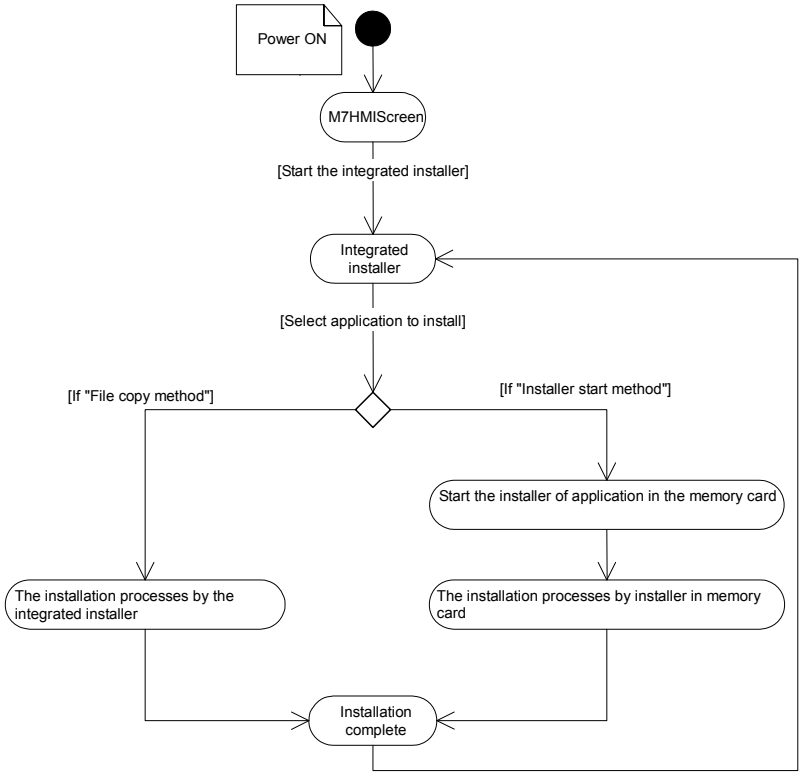
- Installation environment of each application (EXE file and a file to copy, etc.)
- Setting file which collects information on installation of each application
(Thereafter, this setting file is described "setup application list".)

Appendix 10.2 Configuration

The entire configuration of the HMI integrated installer function is as follows.

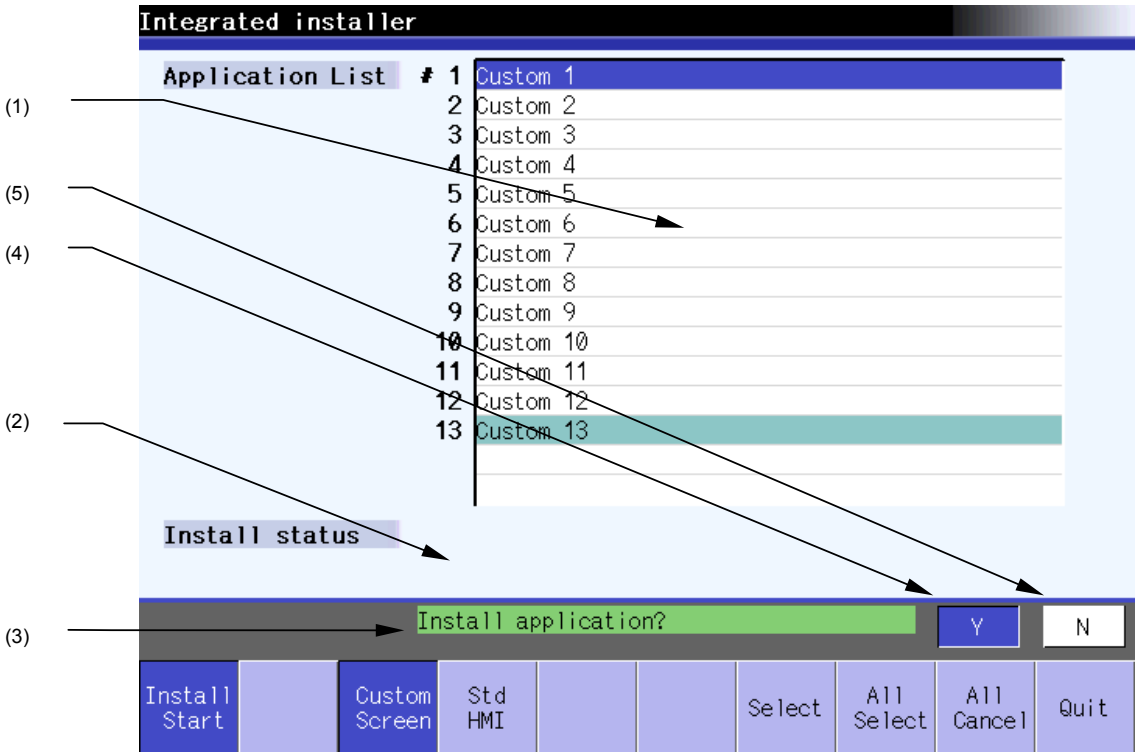


The flow until installation completion of the application is as follows.

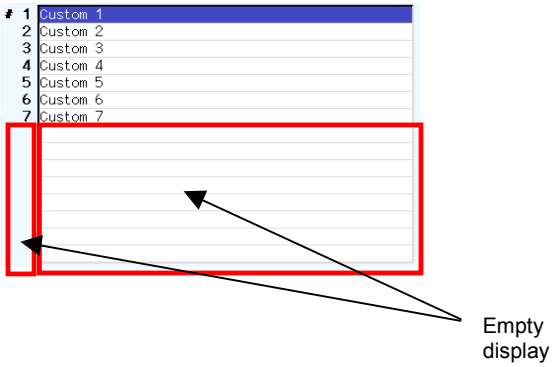


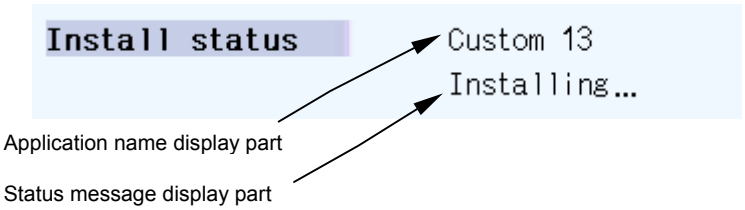
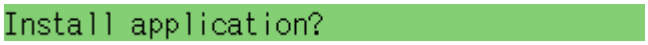
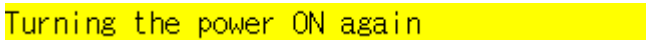
Appendix 10.3 Screen Configuration

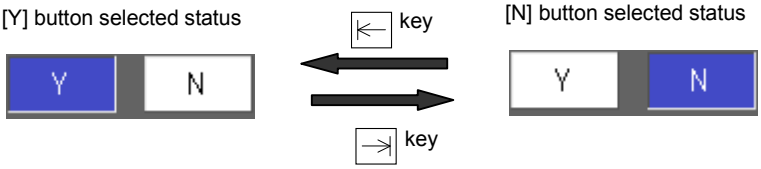
Screen image



Display items

Display item	Details
(1) Application list	This displays the list of the application name which can be installed. The display content of the table changes depending on the selected menu. "Custom screen menu is selected" -> "Application of custom release" list which can be installed "Standard HMI menu is selected" -> "Application of HMI related" list which can be installed "Application of custom release" is displayed as a default. The name set to "Setup application list" is displayed as application name. The cursor is displayed, and the installed application can be selected. The selected application name is highlighted as follows. Two or more applications can be selected. "Application name which has not been selected"  "Application name which has been selected"  "Application name displaying cursor"  The cursor can be moved by ↑ ↓ PAGE PAGE keys. **(Note)** After the line of the final data display, an empty column is displayed. The line No. is not displayed, too. **[Example of display (when number of application is 7)]** 

	Display item	Details
(2)	Installation status display part	This displays the installation status of the installing application.  **[Display examples]** [Before installation]  [After installation]  **"Application name display part"** The name of currently installing application is displayed. **"Status message display part"** The current installation status is displayed as a message. The message "Installing…" is blinked every second while installing the application. When an error will occur during installation, an error message is displayed. **(Note)** When the installation method is "Installer start method", nothing is displayed in all the display parts (The application installation status is not displayed).
(3)	Operation/Power supply restart message display part	This displays the operation messages and the power supply restart messages. The background color is different depending on the kind of the displaying message. "Operation message"  "Power supply restart message" 

Display item	Details
(4) [Y] button (5) [N] button	There are buttons to decide whether to execute or cancel the operation when a confirmation message is displayed in "Operation/Power supply restart message display part" and then the status is changed to the waiting. The meaning of each button is as follows. ● [Y] button : Execute the operation ● [N] button : Cancel the operation Usually (the confirmation message is not displayed), each button is "Invalid status", and cannot be selected. "Button invalid status (grayout)"  When the confirmation waiting status, either button is "Selection status", and the other is "Normal status". (The [Y] button turns to "Selection status" when the status is changed to the waiting.) "Button selection status (highlight)"  "Button normal status (normal display)"  The selected button can be switched with the ->|, |<- key at the confirmation waiting status. [Y] button selected status  [N] button selected status

Menus

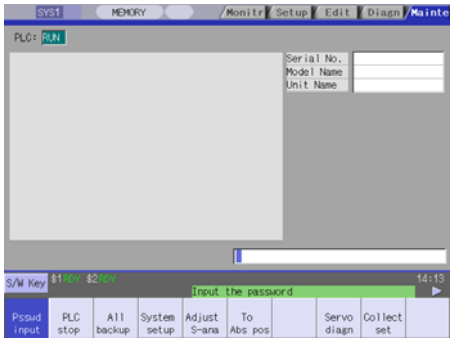
Menu	Details
Install Start	This starts the installation of the selected application.
Custom Screen	This displays the "Application of custom release" name which can be installed in the application list.
Std HMI	This displays the "Application of HMI related" name which can be installed in the application list.
Select	This selects the application at the cursor, and cancels the selection status.
All Select	This selects all applications in the application list.
All Cancel	This cancels the selected status of all applications in the application list.
Quit	This quits the integrated installer. After selected, the end confirmation message is displayed.

Appendix 10.4 Operation Methods

Appendix 10.4.1 Installing the Application

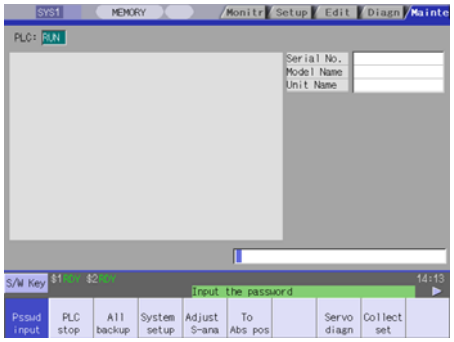
(1) The memory card where either of installation environment of "Application of custom release" or "Application of HMI related" is stored is prepared, and the memory card is inserted in the memory card interface in front of the display.

(2) Select Mainte - Password input menu on the Maintenance screen.



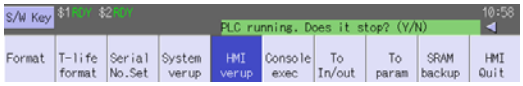
(3) Input the password, and select the INPUT key.

The password is set, and the HMI integrated installer can be started.



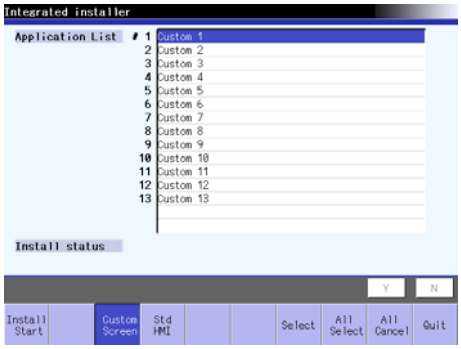
(4) Select HMI verup menu.

The HMI verup menu is highlighted, and the message to confirm the stop of PLC is displayed.

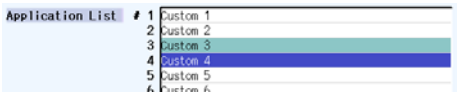


(5) Select the Y or INPUT key.

After PLC is stopped, HMI integrated installer is started, and then the screen is displayed.
When the installation environment of "Application of custom release" in memory card, the "Application of custom release" name which can be installed in the application list is displayed.

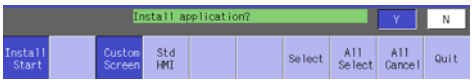


(6) Select one application to install from the displaying application list.



(7) Select Install start menu.

The Install start menu is highlighted, and a message to confirm installing application is displayed in "Operation/Power supply restart message display part". The [Y] button turns to "Selection status".



(Note 1) If any application has not been selected when Install start menu is selected, an error message is displayed and the Install start menu is unhighlighted.

(8) Select the INPUT key.
Or touch the selection status [Y] button.

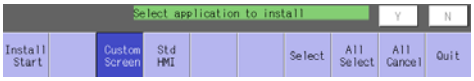
The installation of the selected application is started. The screen display is different depending on the installation method.

(Note 1) When the selection status [N] button or the menu is touched, the Install start menu is unhighlighted and the application is not installed.

(Note 2) While installing, all menus are invalid menus (grayout), and cannot be selected.

[Invalid menu display]

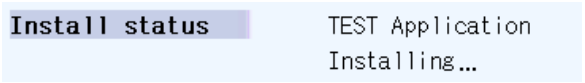
(Note 3) All tasks except the related system are quitted immediately before installing the selected application. Quit all unnecessary tasks before an integrated installer starts.



<<When the application of "File copy method" is selected>>

The application is installed according to the setting of the setup application list. A current installation situation is displayed in "Installation status display part". The message "Installing…" is blinked every second while installing the application.

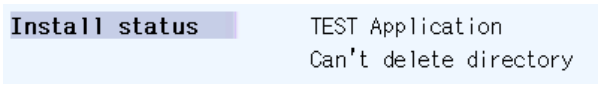
[Example of display 1]
Application name (registered name) -> TEST Application



When an error will occur during installing, an application is not installed, and an error message is displayed in "Installation status display part". In that case, quit the integrated installer once, and review the installation environment and the setup application list, etc. in the memory card.

[Example of display 1]
Application name (registered name) -> Custom 13
To copy the designated file failed.

[Example of display 2]
Application name (registered name) -> Test Application
To delete the designated directory failed.

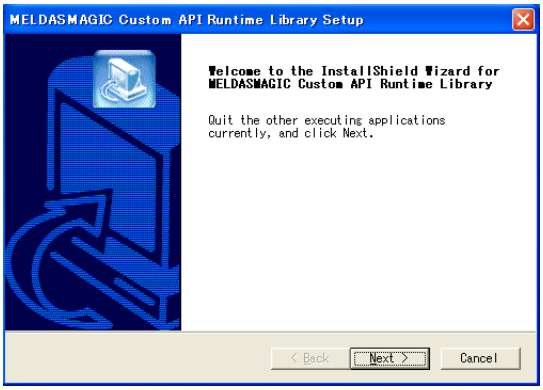


(Note 1) When the installation is interrupted by error, the file installed until interruption does not return to the origin.

(Note 2) The message other than "Installing…" displayed while installing is not blinked ("Finish installed" "Can't copy file", etc.).

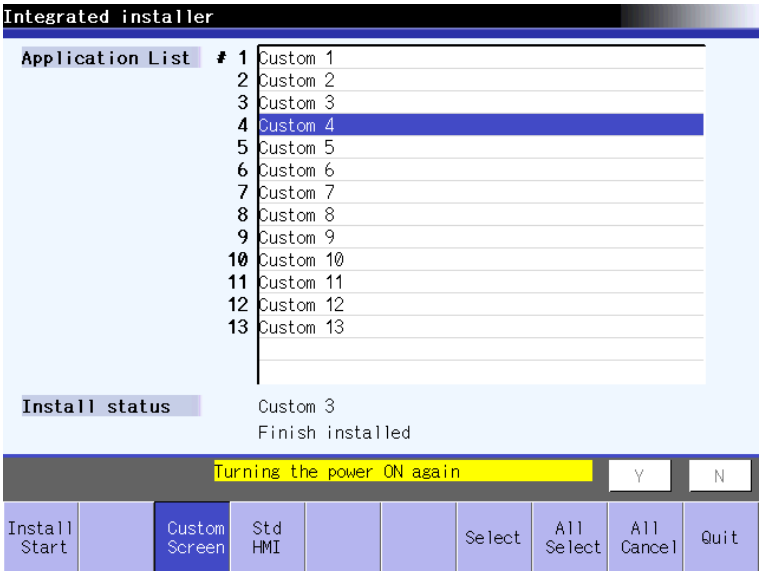
<<When the application of "Installer start method" is selected>>
The installer set to the setup application list is started, and displayed in front of the screen. Install the application according to the guidance displayed on the installer screen.

[Example of display]



(9) The application installation is completed.

<<When the installation of "File copy method" application is completed>>
The installation completion message is displayed in "Installation status display part", and the Install start menu highlighting and the selection status of the application are canceled. The power supply-restart message is displayed in "Operation/Power supply restart message display part".



<<When the installation of "Installer start method" application is completed>>
When returning to the integrated installer screen after the installation screen was quitted, the Install start menu highlighting and the selection status of the application are canceled. The power supply-restart message is displayed in "Operation/Power supply restart message display part".

(Note 1) For "Installer start method", even if canceling the installation on the started installer screen and then returning to the integrated installer screen, the power supply-restart message is displayed in "Operation/Power supply restart message display part". (When the registered installer is started the power supply-restart message is displayed in "Operation/Power supply restart message display part".)

Appendix 10.5 Details for Functions

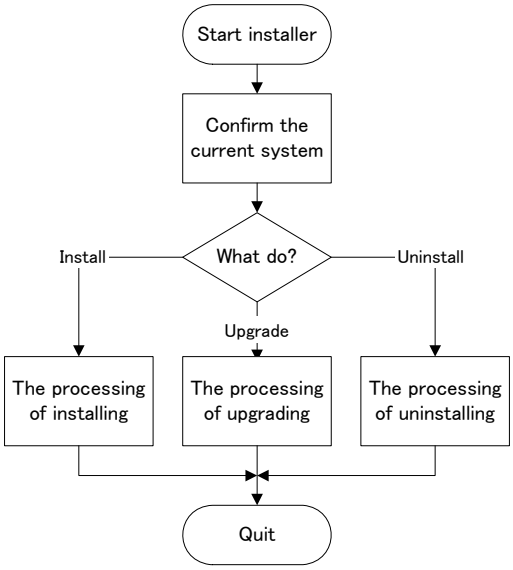
Appendix 10.5.1 Installation Method

For integrated installer, two methods are prepared to install and upgrade the application according to the description of "Appendix 10.1 Outline": "Installer start method" and "File copy method".

Installer start method

The installer for the application (EXE file) prepared beforehand in the memory card is started from an integrated installer, and installed by the installer for the application.
In this method, the setup application list setting is easy, however, it is necessary to create the installer for the application (EXE file). The process outline of the prepared installer is as follows. Create the created installer file by full-screen as much as possible.

<<The process outline of the installer for the application>>



File copy method

The application is installed by "Installation file" in the memory card and an integrated installer.
In this method, it is not necessary to create the installer for the application (EXE file), however, the setup application list setting is complex.
The possible installation processes by an integrated installer are as follows.

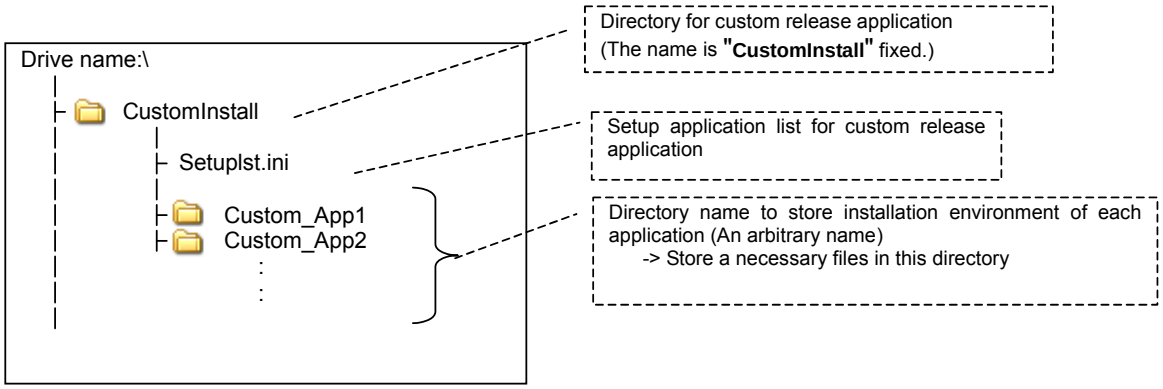
<<The possible installation processes>>

- Overwrite the designated file to an arbitrary directory.
- Copy a new file after deleting the file in an arbitrary directory when the designated file exists in an arbitrary directory.
- Overwrite the designated directory to an arbitrary directory.
- Copy a new file after deleting the directory in an arbitrary directory when the designated directory exists in an arbitrary directory.
- Delete the designated file.
- Delete the designated directory.
- Add the registry key to the registry newly.
- Add the new registry entry to the registry key.
- Change the registry entry data.
  (Only following three types can be changed and added: "32-bit value (DWORD value)", "Character string value", "Binary value".)

Appendix 10.5.2 About the Memory Card for Upgrade

Directory configuration

The directory configuration of the memory card for the upgrade is as follows.



Create for the root directory of the memory card as follows:

- Directory for custom release application (directory name : CustomInstall)

Create the followings in the created directory:

- Setup application list (file name : Setuplst.ini)
- Directory to store installation environment of each application

Setup application list (Setuplst.ini)

"Setup application list" is a file to set the installation settings of each application. The integrated installer installs the application according to the setup application list setting.

(1) File name

Setup application list file name is "Setuplst.ini". Do not apply other file name. Apply the same file name to "For application of custom release" and "For application of HMI related".

(2) Description format

Setup application list (Setuplst.ini) conforms to the description format of the Windows.INI file as a rule, and the maximum size of the INI file is 32KByte. Apply the same file name to "For application of custom release" and "For application of HMI related".

The description format of Windows.INI

```
[(Section)]
(Key) = (Value of key)
        :
[(Section)]
(Key) = (Value of key)
        :
        :
```

(3) About each section and key

[APPLIST] Section

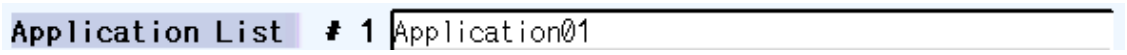
Section name	APPLIST
Details	
Specify the input password when application is installed and the set number of the [APP] section.	
Initial value (when undefined)	
No setting	

1	**Key name**	PASSWORD
Details		
Specify the input password when application installation is started. ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 15 one-byte characters		
Initial value (when undefined)		
No setting		

2	Key name	APPCOUNT
Details		
Specify the number of applications (number of the [APP] section setting) registered in the integrated installer. ● Setting range : 1 to 15 (15 applications or less can be registered.))		
Initial value (when undefined)		
0		

[APP] Section

Section name	APP**
Details	
Specify the information of each application installation. Specify the sequential No. (1 to 15) to the set value [APPCOUNT] of [APPLIST] section to ** of section name. **(Note 1)** Number from 1 sequentially. **(Note 2)** [APP] section to which larger No. than the value set to [APPCOUNT] key of [APPLIST] section is set is ignored.	
Initial value (when undefined)	
No setting	

1	Key name	NAME
Details		
Specify the name displayed in "Application list". ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 45 one-byte characters		
Initial value (when undefined)		
"Application" + "No. set to [APP] section (** part)" [Example of display] For example of [APP01] section, "Application01" is displayed in "Application list". 		

2	Key name	DIR
Details		
Specify the "Directory name to store installation environment" of each application in memory card. ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 100 one-byte characters **(Note 1)** The following directory cannot be selected: the directory name of 100 or more characters, or the directory name used with two-byte character.		
Initial value (when undefined)		
No setting		

3	Key name	INSTALLTYPE

Details

Specify the application installation method. The key which needs setting is changed depending on this key setting value.

- Setting range : 1 to 2
  1 : Installer start method
  2 : File copy method

Initial value (when undefined)

1 (Installer start method)

4	Key name	INSTALLER

Details

Specify the file name (include the extension (.exe)) of the installer (EXE file) which is started when the installation method is "Installer start method ([INSTALLTYPE] key is 1.)".

- Character type which can be set : Only the alphanumeric character
- Number of character which can be set : Within 50 one-byte characters

(Note 1) Specify the installer (EXE file) which exists in the directory specified with the [DIR] key. The file name of the installer which exists in other directories cannot be specified.

(Note 2) When the installation method is "File copy method ([INSTALLTYPE] key is 2)", the key is ignored. It is not necessary to set.

Initial value (when undefined)

setup.exe

5	Key name	FILE

Details

Specify the setting of operation to the file by comma-deliminated character when the installation method is "File copy method ([INSTALLTYPE] key is 2)". The format of character string is as follows.

"The format of character string"

(Action), (FileName), (ActionDir)

	Name	Detail
1	Action	Designate the operation to the file (directory) specified by "FileName". ● Setting range : 1 to 3 1 : Overwrite the file (directory) to the copy destination. 2 : Copy after the file (directory) to the copy destination is deleted. 3 : Delete the file (directory).
2	FileName	Specify the operation file (directory) name designated by "Action". ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 50 one-byte characters (Note 1) Specification with wild-card (“*") is also possible. **["*." + extension]** [Setting example] *.txt -> All files with "txt" as extension **["*"]** -> All files and directory (Note 2) When the file is copied ("Action" is "1" or "2"), specify the file (directory) which exists in the directory specified with the [DIR] key. The file name (directory) which exists in other directories cannot be specified.
3	ActionDir	Specify the directory path which relates to the operation designated by "Action". ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 100 characters <<When copying the file ("Action" is "1" or "2")>> Specify the directory path of the file (directory) to the copy destination specified by “FileName". <<When deleting the file ("Action" is "3")>> Specify the directory path in which the file (directory) specified by “FileName" exists. (Note 1) In both cases, the path is set by full path.

(Note 1) The number of [FILE] keys which can be set to one [APP**] section is up to 100. All [FILE] keys over 100 are disregarded.
(Note 2) When the application is installed, the file designated with each [FILE] key is operated by the setup application list sequence.
(Note 3) When the installation method is "Installer start method ([INSTALLTYPE] key is 1)", all [FILE] key is ignored. It is not necessary to set.
(Note 4) When the "Action" is "2" or "3" and the deleted file (directory) does not exist, the deletion is not executed. An error will not occur during installation.
(Note 5) The read-only file cannot be rewritten and deleted.

Initial value (when undefined)

No setting

6	**Key name**	REGKEY

Details

Specify the name of registry key which is created newly when the installation method is "Installer start method ([INSTALLTYPE] key is 2.)". Set the key name including all the parents keys. Specify the separation of each key name "\".

- Character type which can be set : Only the alphanumeric character
- Number of character which can be set : Within 150 one-byte characters (including the parents key name and each key name separation "\")

(Note 1) The number of [REGKEY] keys which can be set to one [APP**] section is up to 100. All [REGKEY] keys over 100 are disregarded.
(Note 2) When the application is installed, the registry key designated with each [REGKEY] key is created by the setup application list sequence.
(Note 3) When the installation method is "Installer start method ([INSTALLTYPE] key is 1)", all [REGKEY] key is ignored. It is not necessary to set.
(Note 4) When the set registry key has already existed, nothing is executed. An existing registry key cannot be deleted. An error will not occur during installation.

Initial value (when undefined)

No setting

7	Key name	REGKEYVALUE

Details

Specify the registry entry set to the registry key by comma-deliminated character when the installation method is "File copy method ([INSTALLTYPE] key is 2)". The format of character string is as follows.

"The format of character string"

(RegistryKeyName), (ValueName), (ValueType), (Value)

	Name	Details
1	RegistryKeyName	Specify "Registry key name" which newly sets the registry entry. Set the key name including all the parents keys. Specify the separation of each key name "\". ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 150 one-byte characters (including the parents key name and each key name separation "\")
2	ValueName	Specify the name of the registry entry set to registry key designated by "RegistryKeyName". ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 50 one-byte characters
3	ValueType	Specify the data type of the registry entry set to registry key designated by "RegistryKeyName". ● Setting range : 1 to 3 1 : 32-bit value (DWORD value) 2 : Character string value 3 : Binary value

4	Value	Specify the value of the registry entry set to registry key designated by "RegistryKeyName". The setting range is different depending on the setting value of "ValueType". Setting range [When ValueType = 1 (32-bit value)] ● Notation : Either decimal or hexadecimal notation <<Example of setting (when the value is "10000")>> Value=10000 (decimal notation) Value=0x2710 (hexadecimal notation) (Note 1) Add "0x" on the head of the value when the notation is hexadecimal. [When ValueType = 2 (character string value)] ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 50 one-byte characters <<Example of setting (when the value is "Custom1")>> Value="Custom1" (Note 2) Enclose the character string with " (double quotation mark). [When ValueType = 3 (binary value)] ● Character type which can be set : Only the alphanumeric character ● Number of character which can be set : Within 50 one-byte characters

(Note 1) The number of [REGKEYVALUE] keys which can be set to one [APP**] section is up to 100. All [REGKEYVALUE] keys over 100 are disregarded.
(Note 2) When the application is installed, the registry entry designated with each [REGKEYVALUE] key is set by the setup application list sequence.
(Note 3) When the installation method is "Installer start method ([INSTALLTYPE] key is 1)", all [REGKEYVALUE] key is ignored. It is not necessary to set.
(Note 4) When the designated registry entry has already existed, the existing registry entry is changed.
(Note 5) An existing registry entry cannot be deleted.

Initial value (when undefined)

No setting

(Note 1) When the application is installed, the installation is operated by the following sequence.
(1) [FILE] key : (Designated file operation)
(2) [REGKEY] key : (Add registry key)
(3) [REGKEYVALUE] key : (Add registry entry)

This sequence cannot be changed.

(Note 2) The key which needs settings are different depending on the method of installing the application.

Installer start method

	Key name	Comment
1	NAME	
2	DIR	Indispensability
3	INSTALLTYPE	
4	INSTALLER	Indispensability

File copy method

	Key name	Comment
1	NAME	
2	DIR	Indispensability
3	INSTALLTYPE	Indispensability
5	FILE	
6	REGKEY	
7	REGVALUE	

(Note 3) Define the each section key to the setup application list in following the sequence.

[APPLIST] section

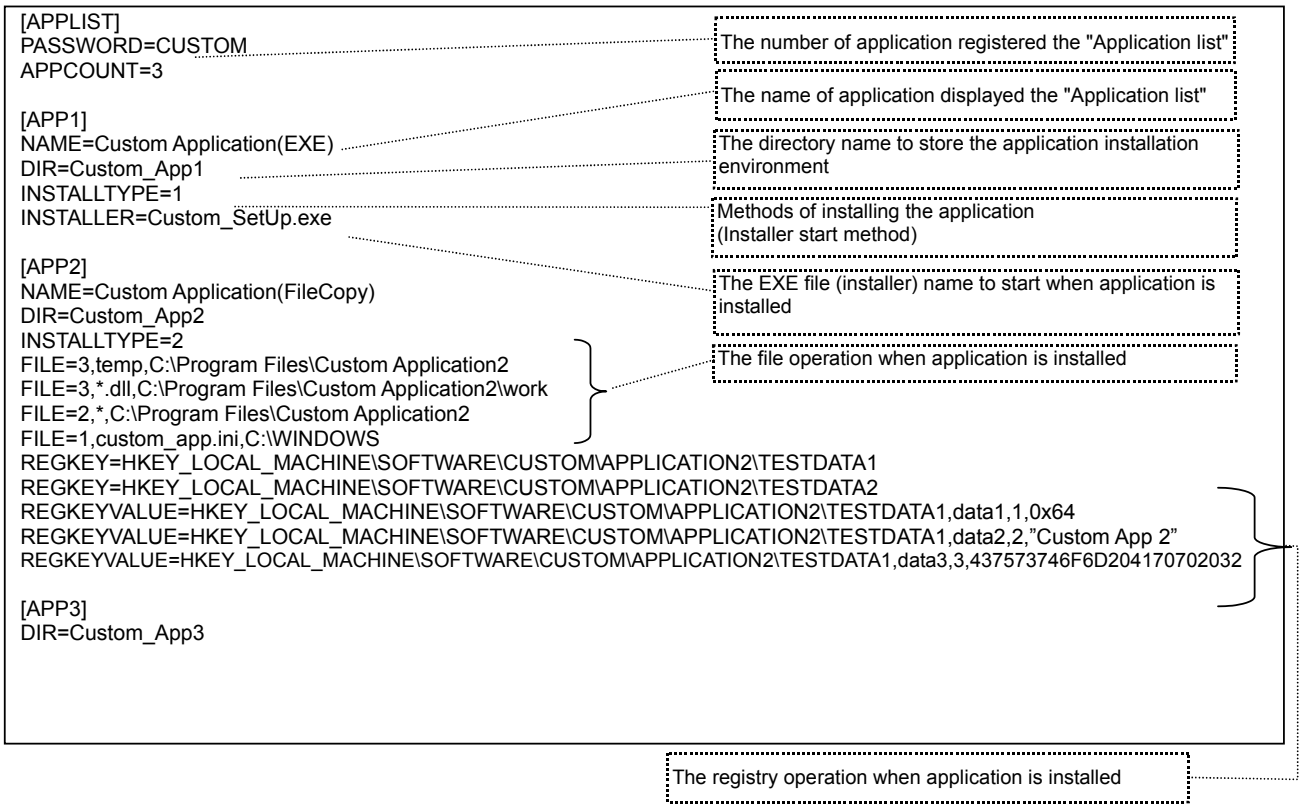
	Key name	Comment
1	PASSWORD	
2	APPCOUNT	Indispensability

[APP] section

	Key name	Comment
1	NAME	
2	DIR	Indispensability
3	INSTALLTYPE	Indispensability
4 or later	Any of the following: INSTALLER FILE REGKEY REGKEYVALUE	Indispensability

(4) Example of setting ("For application of custom release")

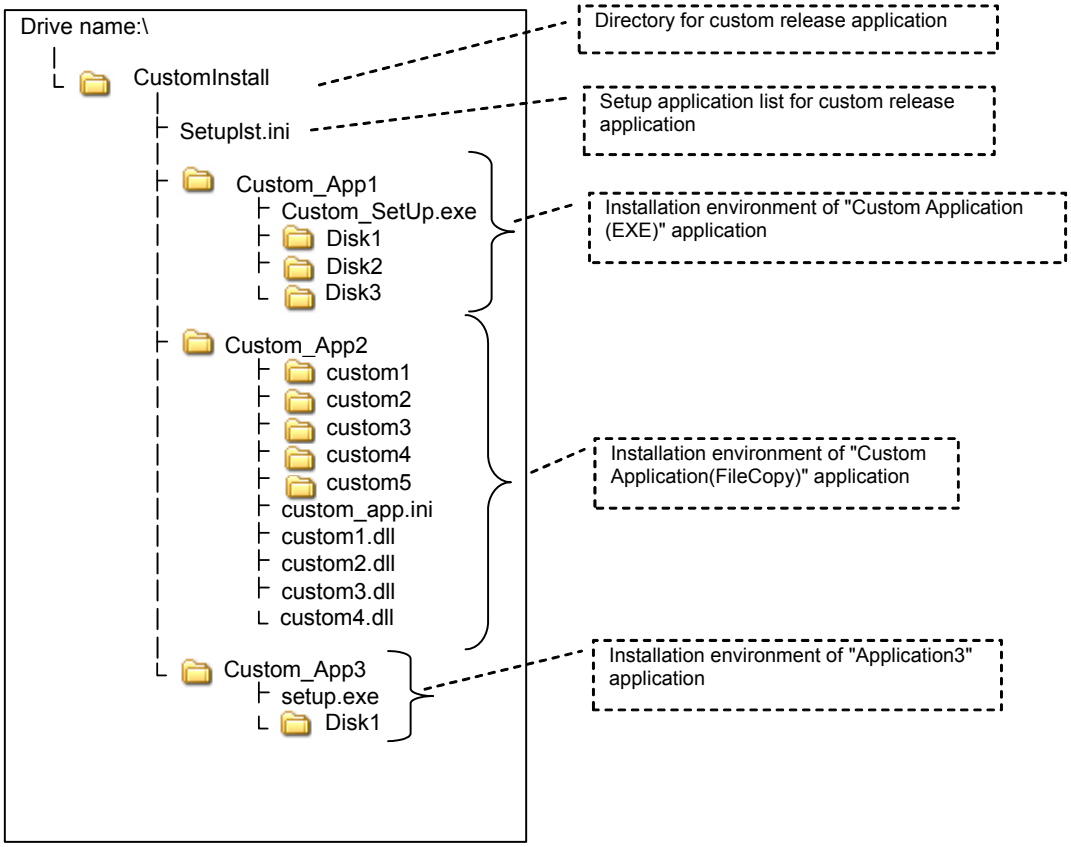
Setup application list



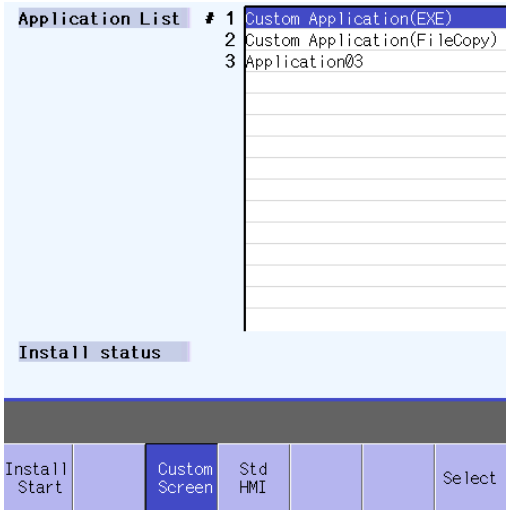
NOTE

- APP1] is the installer start method, and [APP2] is the file copy method.

File configuration in memory card



"Application list" when Custom screen menu is selected



The flow of each application installation

● "Custom Application (EXE)" application
1) Select "Custom Application (EXE)" from the application list, and select [Install start] menu.
2) The installer (Custom_SetUp.exe) which exists in the "Custom_App1" directory (the installation environment of "Custom Application (EXE)" application) is started, and the installer screen is displayed in front of the screen.
3) Install the application according to the guidance displayed on the installer screen.
4) After the installation is completed, the installer screen is quitted, and return to the integrated installer screen.

● "Custom Application (FileCopy)" application
1) Select "Custom Application (FileCopy)" from the application list, and select [Install start] menu.
2) Delete the "temp" directory including file if the "temp" directory exists in the "C:\Program Files\Custom Application2" directory.
3) Delete the file with ".dll" as extension if the file with ".dll" as extension exists in the "C:\Program Files\Custom Application2" directory.
4) Copy all files and directories which exist in the "Custom_App2" directory (the installation environment of "Custom Application (FileCopy)" application) after all files and directories in "C:\Program Files\Custom Application2" directory are deleted.
5) Overwrite "custom_app.ini" file which exists in "Custom_App2" directory (the installation environment of "Custom Application (FileCopy)" application) to "C:\WINDOWS" directory.
6) Create the "TESTDATA1" key as a subkey of "HKEY_LOCAL_MACHINE\SOFTWARE\CUSTOM\APPLICATION2" key to the registry.
7) Create the "TESTDATA2" key as a subkey of "HKEY_LOCAL_MACHINE\SOFTWARE\CUSTOM\APPLICATION2" key to the registry.
8) Add the entry (entry name "data1", data type "32-bit value", data "0x00000064") to the key created in "6)".
9) Add the entry (entry name "data2", data type "character string value", data "Custom App 2") to the key created in "6)".
10) Add the entry (entry name "data3", data type "binary value", data "437573746F6D204170702032") to the key created in "6)".
11) The installation is completed.

● "Application3" application
1) Select "Application3" from the application list, and select [Install start] menu.
2) The installer (setup.exe) which exists in the "Custom_App3" directory (the installation environment of "Application3" application) is started, and the installer screen is displayed in front of the screen.
3) Install the application according to the guidance displayed on the installer screen.
4) After the installation is completed, the installer screen is quitted, and return to the integrated installer screen.

Appendix 10.6 Parameter

The following table shows the related parameters.

[Parameter list]

No.	Name	Detail	Range
#1043	lang (Select language displayed)	Specify the display language. 0: English display (Standard) 1: Japanese display (Standard) 3: Third language 4: Fourth language 11: Display in German (Option) 12: Display in French (Option) 13: Display in Italian (Option) 14: Display in Spanish (Option) 15: Display in Chinese (Option) (traditional Chinese) 16: Display in Korean (Option) 17: Display in Portuguese (Option) 18: Display in Dutch (Option) 19: Display in Swedish (Option) 20: Display in Hungarian (Option) 21: Display in Polish (Option) 22: Display in Chinese (Option) (simplified Chinese) 23: Display in Russian (Option) 24: Display in Turkish (Option) 25: Display in Czech (Option)	0 to 3 11 to 25

NOTE

- On HMI screen, start the integrated installer after the related parameters is set.
  The integrated installer cannot set the related parameters.
- The integrated installer refers the parameter setting value when the integrated installer has started.
  Even if the parameter is changed after starting, the change is invalid.

Appendix 10.7 Operation/Alarm Messages

The message displayed in each display part is as follows.

[Message list]
<<Operation/Power supply restart message display part>>

Message	Details
Operation message	
Install application?	It is confirmed whether the installation of the application is started. ● [Y] : Start the installation of the application. ● [N] : Do not start the installation of the application.
Select application to install	An application to install has not been selected from "Application list". Select an application.
Can't install application	The selected application cannot be installed. Confirm the mounting status of the memory card and the installation environment in the memory card.
Quit ?	It is confirmed whether the integrated installer is quitted. ● [Y] : Quit the integrated installer. ● [N] : Do not quit the integrated installer.
Power supply restart message	
Turning the power ON again	It is necessary to turn the power supply ON again because the application was installed. Turn the power supply ON again.

<<Installation status display part>>

Message	Details
Installing…	The application is installing.
Finish installed	The application installation was completed.
Can't copy file	To copy the specified file failed while installing. Confirm the setting of the setup application list and the installation environment in the memory card.
Can't delete file	To delete the specified file failed while installing. Confirm the setting of the setup application list and the file to delete.
Can't copy directory	To copy the specified directory failed while installing. Confirm the setting of the setup application list and the installation environment in the memory card.
Can't delete directory	To delete the specified directory failed while installing. Confirm the setting of the setup application list and the directory to delete.
Can't add registry key	To add the specified registry key failed while installing. Confirm the setting of the setup application list.
Can't add registry entry	The registry entry could not be added to the specified registry key while installing. Confirm the setting of the setup application list.
Can't copy	To copy the specified file or directory failed while installing. Confirm the setting of the setup application list and the installation environment in the memory card.
Can't delete	To delete the specified file or directory failed while installing. Confirm the setting of the setup application list and the file/directory to delete.

Appendix 11. Installing Custom Data (M70/M70V/M700VS/E70)

For M70/M70V/M700VS display unit

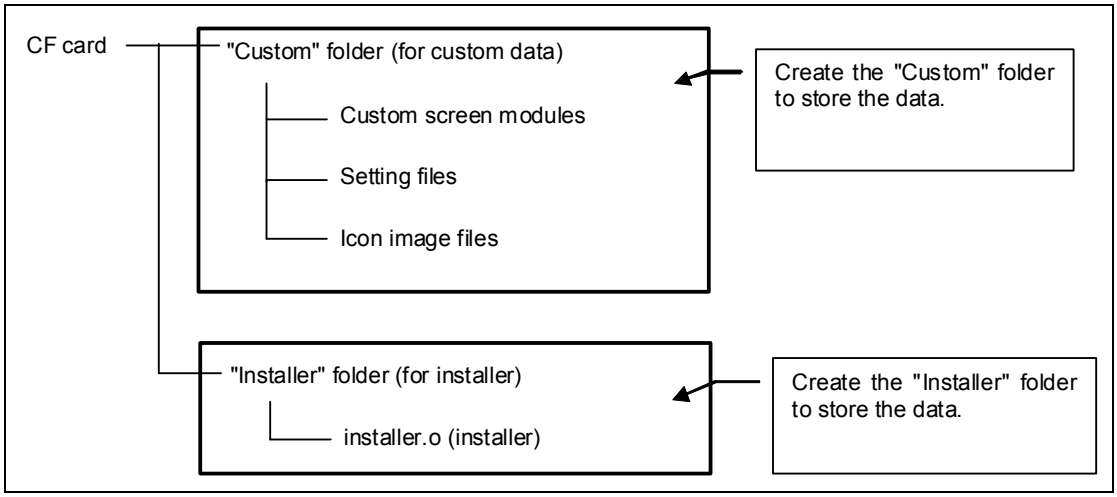
For M70/M70V/M700VS/E70 unit, use M70/M700 SETUP INSTALLER to install the custom screen data. Use a CF card to install.

(1) Data for M70/M700 SETUP INSTALLER

Type	Data	Contents	Remarks
Custom data	Custom screen module	Interpreter data and object data	
	Config.ini	Setting file for the assignment of the custom screen	
	customdef.ini	Setting file for the assignment of the custom screen to a menu or function button on the standard screen.	
	customload.txt	Setting file for registering the object data name and the load order	
	*.jpg	File for the icon image	

(2) Folder configuration in CF card

Make the following folder configuration in a CF card to store the data for M70/M700 SETUP INSTALLER.



(Note 1) "installer.o" is provided by Mitsubishi Electric Corporation. Create the "Installer" folder in a CF card to store it.

Starting up M70/M700 SETUP INSTALLER

(1) Insert a CF card for M70/M700 SETUP INSTALLER into the front panel CF.

(2) Turn the power ON while pressing the ◁ menu.

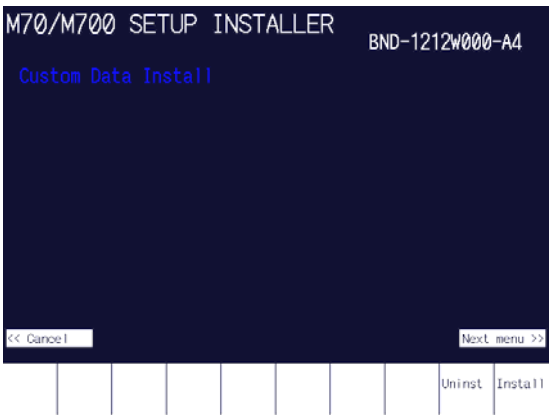
Startup screen appears. A bleep sounds after a while.
Then the mode selection screen for M70/M700 SETUP INSTALLER appears.

(Note 1) Keep pressing the ◁ key until the Mode Select screen appears.

Installing custom data

(1) Press the [Custom Data] menu key on the mode selection screen.

Custom data installation screen appears.



(2) Press the [Install] menu key.

A comfirmation message appears.

(3) Press the [Yes] menu key.

A message appears after the installation has completed.

- Pressing the [No] menu key returns to the first menu.
- Do not turn the power OFF during the installation of custom data.

NC Matching List【NC Designer full-function】

NC Designer must be combined with a matching NC as listed below to provide all its functions.

<table>
<tr><td rowspan="2">NC Designer Version</td><td colspan="6">Type</td></tr>
<tr><td>M700</td><td>M70</td><td>M700VS</td><td>M700VW</td><td>M70V</td><td>E70</td></tr>
<tr><td>A0</td><td>B5</td><td rowspan="3">E0</td><td rowspan="3">F0</td><td rowspan="3">F2</td><td rowspan="4">G0</td><td rowspan="8">J0 or later</td></tr>
<tr><td>A1</td><td rowspan="2">C0</td></tr>
<tr><td>A2</td></tr>
<tr><td>A3</td><td colspan="2">E1</td></tr>
<tr><td>A4</td><td colspan="4">F4</td></tr>
<tr><td>A5</td><td colspan="2">FB</td><td colspan="3">G4</td></tr>
<tr><td>A6</td><td colspan="2">FD</td><td colspan="3">HO</td></tr>
<tr><td>A7</td><td colspan="2">FE</td><td colspan="3">H3</td></tr>
</table>

(Note1) Before upgrading the NC Designer, be sure to uninstall the old version.

NC Matching List【Additional controls】

NC Designer must be combined with a matching NC as listed below to provide additional controls.

<table>
<tr><td rowspan="2">Additional control</td><td colspan="6">Type</td></tr>
<tr><td>M700</td><td>M70</td><td>M700VS</td><td>M700VW</td><td>M70V</td><td>E70</td></tr>
<tr><td>NC data text box</td><td colspan="2">E1 or later</td><td>F0 or later</td><td>F2 or later</td><td rowspan="7">G0 or later</td><td rowspan="9">J0 or later</td></tr>
<tr><td>I/O control</td><td colspan="4" rowspan="6">F4 or later</td></tr>
<tr><td>PLC extension button control</td></tr>
<tr><td>PLC message control</td></tr>
<tr><td>Operation status display</td></tr>
<tr><td>Alarm display</td></tr>
<tr><td>Time display</td></tr>
<tr><td>Input box</td><td>-</td><td>-</td><td colspan="3" rowspan="2">G4 or later</td></tr>
<tr><td>Ten-key</td><td>-</td><td>-</td></tr>
</table>

NC Matching List【Additional functions/variables】

NC Designer must be combined with a matching NC as listed below to provide additional functions and variables.

<table>
<tr><td rowspan="2"></td><td colspan="6">Type</td></tr>
<tr><td>M700</td><td>M70</td><td>M700VS</td><td>M700VW</td><td>M70V</td><td>E70</td></tr>
<tr><td>Macro functions of NC control</td><td colspan="2">E2 or later</td><td>F0 or later</td><td>F2 or later</td><td>G0 or later</td><td rowspan="4">J0 or later</td></tr>
<tr><td>Functions to set/get input character string of NC data text box</td><td>-</td><td>-</td><td colspan="3">G3 or later</td></tr>
<tr><td>Read and write common variables with macro program (#100-, #500-)</td><td>-</td><td>-</td><td colspan="3" rowspan="2">G4 or later</td></tr>
<tr><td>Read and write macro reserved word
variables (@1-@999, #1-#99)</td><td>-</td><td>-</td></tr>
<tr><td>Read and write the common variables shared among the part systems (#900000-#907399) with macro program</td><td>-</td><td>-</td><td colspan="3">H0 or later</td></tr>
</table>

Revision History

Date of revision	Manual No.	Revision details
Nov. 2004	IB(NA)1500040-B	First edition created.
Feb. 2007	IB(NA)1500040-C	• The following sections were added. 1.1.3 Interpreter Method and Compilation Method 7.2.13 Table Object (GNCTable) 7.3 NC Control Object 15.3.3 Method of Compilation Environment Establishment in Windows CE 15.5.2.1 Changing From the Custom Screen to 700 Series Standard Screen (F0 Release) 15.2.2.2 Changing the Menu Name While Displaying Custom Screen (Menu Release) 15.5.2.3 Closing the Custom Screen (Menu Release) 15.6 Custom Release 16.5.14 NCPLCButton 16.5.15 NCPLCTextbox Appendix 9. Executing File Registration Method Appendix 10. HMI Integrated Installer • The following sections were deleted. 13.3 Source File Format (C language version) 13.5 User Program Generation 15.6 Config.ini File Setting • "7.2 Normal Control" was added, and the chapter composition of 7.2 to 7.13 was changed. • Mistakes were corrected.
Jul. 2010	IB(NA)1500040-D	• The following sections were added. 7.3.14 NC Data Textbox (GNCDataTextBox) 7.3.15 PLC extension button (GNCPLCExButton) 7.3.16 PLC Message (GNCPLCMessage) 7.3.17 Menu (GNXMenu) ; Menu display part 7.3.18 FileInOut (GNXFileTransfer) ; Input/Output Control 7.3.19 AlarmMessage (GNXAlarmMessage) ; Alarm Display Part 7.3.20 MonitorStatus (GNXMonitorStatus) ; Operation Status Display Part 7.3.21 Time (GCNXTime) ; Time Display Part 15.5.5 Displaying Previously Displayed Custom Screen 15.7 M70 S/W Keyboard 16.5.16 NCDataTextBox 16.5.17 Menu 16.5.18 FileInOut 16.5.19 AlarmMessage 16.5.20 MonitorStatus 16.5.21 Counter 16.5.22 CycleTime 16.5.23 LoadMeter 16.5.24 SPCommand 16.6 NC Data Access Function 18.14 GCNCControl (NC Control Management) Appendix 11. M70/M700VS SETUP INSTALLER (Continue to the next page)

Date of revision	Manual No.	Revision details
		(Continued) • The following contents were added. 6.3 Locale The number of resources which can be registered was raised to 32. Czech 7.3.9 ONB (GNXONB) ; ONB display part Percentage display property 7.3.12 PLC Button Object (GNCPLCButton) Update condition property 13.1 Generating Screen Data Source Codes Added the following information; “The binary conversion type of M70, M70V, M700 and M700V is a little-endian format.” 15.6 Custom Release Added M70 series PANEL_HOLD 16.5.1 Normal Command and 16.5.2 Control Common Command Macro Function 17.1 Outline GCNCControl Class, NC Control Class Changed the Class Diagram Changed Class Definition Appendix 6. Data Type Definitions NC Data Access Related Structures • The following contents were deleted. The overlapping explanations of the properties of common functions of controls in section 7.2 and 7.3. • Mistakes were corrected.
Feb. 2011	IB(NA)1500040-E	• The following contents were added. 7.2.14 Input box(GInputBox) 7.2.15 Numerical keypad(GSoftKey) 9.7 Sub cursor setting 16.5.25 NCTable 16.5.26 InputBox 16.5.27 SoftKey 18.15 GCInputBox(Input box) 18.16 GCSoftKey(Numerical keypad) Appendix.9.3 Using a function key (screen switching key) in the custom screen (Continue to the next page)

Date of revision	Manual No.	Revision details
		(Continued) • The following contents were added. 7.3.1 Counter（GNXCounter） Part system designation property AxisCross property Details of axis status display Related NC parameters 7.3.2 CycleTime（GNXCycleTime） Part system designation property Time type property Normal bold font 7.3.3 Feedrate（GNXFeedrate） Part system designation property 7.3.4 GModal M（GNXGModal） Part system designation property MSTBVisible property 7.3.5 GModal L（GNXGModal_L） Part system designation property 7.3.6 GModal Simple（GNXGModalSimple） Part system designation property 7.3.7 LoadMeter（GNXLoadMeter） Part system designation property 7.3.8 MSTB（GNXMSTB） Part system designation property Item “ScrollBarVisible” was added to Display property Item “LineNumber” was added to Display property 7.3.9 ONB（GNXONB） Part system designation property 7.3.10 ProgramBuffer（GNXPrgBuff） Part system designation property Character attribute property 7.3.11 SPCommand（GNXSPCommand） Axis designation property Display digit property Three columns display property Middle-sized font 16.4.3 Programming Language Macro reserved word variable NC system variable Operator (added MOD, OR, AND, XOR) Operation function Type conversion Condition Expression (added EQ, GT, GE, LT, LE, NE) Program Branch (added WHILE, GOTO, sequence No.) 16.5.16 NCDataTextBox The following two functions were added. ・Set input character string (GCSNCDataTextboxSetStringBuffer) ・Get input character string (GCSNCDataTextboxGetStringBuffer) 16.7 Error Message List Error messages between error code 20 to 23 18.14.1 GNCDataTextBox（NC data text box） The following two functions were added. ・Set input character string (GCSNCDataTextBoxSetStringBuffer) ・Get input character string (GCSNCDataTextBoxGetStringBuffer) (Continue to the next page)

Date of revision	Manual No.	Revision details
		(Continued) Appendix.1 Error Message List The following three messages were added. ・The X coordinate is not set ・The Y coordinate is not set ・The setting range of X coordinate is between 0 and 2559 and Y coordinate is between 0 and 1919. Appendix.9.1.2 Details of melAppCtrl.ini ControlParam key was added. ●The following contents were changed. 6.11 HTML File Name Resource HTML File Name Resource → File Name Resource 7.2.10 HTML Browser Object Storage location of HTML file 13.1 Generating Screen Data Source Codes Resource-related code file Protected → Not protected 15.6.2.2 Necessary Files M70/M700VS unit storage folder /custom/img/ → /custom/ ●The following sections were deleted. 14.3 Folder Configuration Folder storing manuals, Folder storing library files, Folder storing samples ● Mistakes were corrected.
Nov.2011	IB(NA)1500040-F	●The following chapters were added. 15.3.2.1 Operation Procedure with Visual C++6.0 15.3.2.2 Operation Procedure with Visual Studio2010 ●The following contents were added. 2.2.4 Operating Environment of NC Designer ・Windows Vista/7 were added to the system environment. ・The limitation of installation folder due to the UAC function. 4.2 Creating a New Project/4.6 Saving the Project as .../ 5.4 Entering Panel Properties/5.5 Entering Window Properties/ 5.11 Creating View Frame The names that cannot be used for the project name, panel name, window name, and view frame name in the compilation method. 4.6 Saving the Project as ... The limitation of the saving destination of the project for Windows Vista/7 due to the UAC function. 7.2.3.1 Property Settings Notes on the display format setting. 8. Figure Notes on the setting for the start point angle and end point angle from the sector/arc's property. 9.4 Focus Setup The tab key to the focus movement direction. (Continue to the next page)

Date of revision	Manual No.	Revision details
		(Continued) 12.2 File Configuration ・config.ini setting when using the project macro. ・Notes on when using the project macro. ・Limitation about the screen macro. 15.4.2 Launching the Application Window Install destination folder for Windows 2000/XP, and Windows Vista/7. 15.6.2.1 Necessary Applications Visual Studio 2005/2008/2010 are now usable for M700/M700VW. 15.6.4.1.1 ・Limitation when using the project macro. ・Setting for [PROJECT] section when using the project macro. 15.6.5.1.2 customdef.ini Explanations of the focus when the instance is held. 16.3.2 Macro Editing Dialog Box Notes on creating the event. 16.4.3 Programming Language Added the common variables shared among the part systems #900000 to #907399 to "NC system variable" and "(2) Common variable". 16.5 Function Details Added GMEMDelete() corresponding to the macro example GMEMCreate() to avoid memory leak. 16.5.13 TextBox Notes on setting the character string type. NC Matching List【NC Designer full-function】 Notes on upgrading from A5 to A6. •Mistakes were corrected.
Mar.2013	IB(NA)1500040-G	The following chapters were added. 3.4 Installing NC Designer 5.12 Importing a Page · Added GCSResourceLoadString to display menu controls in multilingual environment to the character string setting function of the macro example. · Additional functions/variables were added to the NC Matching List. · Corresponded to MITSUBISHI CNC E70 series. · Mistakes were corrected.

Global Service Network

AMERICA

MITSUBISHI ELECTRIC AUTOMATION INC. (AMERICA FA CENTER)

Central Region Service Center
500 CORPORATE WOODS PARKWAY, VERNON HILLS, ILLINOIS 60061, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-847-478-2650

Michigan Service Satellite
ALLEGAN, MICHIGAN 49010, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-269-673-4092

Ohio Service Satellite
LIMA, OHIO 45801, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-847-478-2650
CLEVELAND, OHIO 44114, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-847-478-2650

Minnesota Service Satellite
ROGERS, MINNESOTA 55374, U.S.A.
TEL: +1-847-478-2500 / FAX: +1-847-478-2650

West Region Service Center
16900 VALLEY VIEW AVE., LAMIRADA, CALIFORNIA 90638, U.S.A.
TEL: +1-714-699-2625 / FAX: +1-847-478-2650

Northern CA Satellite
SARATOGA, CALIFORNIA 95070, U.S.A.
TEL: +1-714-699-2625 / FAX: +1-847-478-2650

East Region Service Center
200 COTTONTAIL LANE SOMERSET, NEW JERSEY 08873, U.S.A.
TEL: +1-732-560-4500 / FAX: +1-732-560-4531

Pennsylvania Service Satellite
ERIE, PENNSYLVANIA 16510, U.S.A.
TEL: +1-814-897-7820 / FAX: +1-814-987-7820

South Region Service Center
1845 SATTELITE BOULEVARD STE. 450, DULUTH, GEORGIA 30097, U.S.A.
TEL +1-678-985-4529 / FAX +1-678-258-4519

Texas Service Satellites
GRAPEVINE, TEXAS 76051, U.S.A.
TEL: +1-817-251-7468 / FAX: +1-817-416-5000
HOUSTON, TEXAS 77001, U.S.A.
TEL: +1-678-258-4529 / FAX: +1-678-258-4519

Florida Service Satellite
WEST MELBOURNE, FLORIDA 32904, U.S.A.
TEL: +1-321-610-4436 / FAX: +1-321-610-4437

Canada Region Service Center
4299 14TH AVENUE MARKHAM, ONTARIO L3R OJ2, CANADA
TEL: +1-905-475-7728 / FAX: +1-905-475-7935

Canada Service Satellite
EDMONTON, ALBERTA T5A 0A1, CANADA
TEL: +1-905-475-7728 / FAX: +1-905-475-7935

Mexico City Service Center
MARIANO ESCOBEDO 69 TLALNEPANTLA, 54030 EDO. DE MEXICO
TEL: +52-55-9171-7662 / FAX: +52-55-9171-7649

Monterrey Service Satellite
MONTERREY, N.L., 64720, MEXICO
TEL: +52-81-8365-4171 / FAX: +52-81-8365-4171

BRAZIL

MELCO CNC do Brasil Comércio e Serviços S.A

Brazil Region Service Center
ACESSO JOSE SARTORELLI, KM 2.1 CEP 18550-000, BOITUVA-SP, BRAZIL
TEL: +55-15-3363-9900 / FAX: +55-15-3363-9911

EUROPE

MITSUBISHI ELECTRIC EUROPE B.V. (EUROPE FA CENTER)
GOTHAER STRASSE 10, 40880 RATINGEN, GERMANY
TEL: +49-2102-486-0 / FAX: +49-2102-486-5910

Germany Service Center
KURZE STRASSE. 40, 70794 FILDERSTADT-BONLANDEN, GERMANY
TEL: + 49-711-770598-121 / FAX: +49-711-770598-141

France Service Center
25, BOULEVARD DES BOUVETS, 92741 NANTERRE CEDEX FRANCE
TEL: +33-1-41-02-83-13 / FAX: +33-1-49-01-07-25

France (Lyon) Service Satellite
120, ALLEE JACQUES MONOD 69800 SAINT PRIEST FRANCE
TEL: +33-1-41-02-83-13 / FAX: +33-1-49-01-07-25

Italy Service Center
VIALE COLLEONI 7-PALAZZO SIRIO CENTRO DIREZIONALE COLLEONI,
20864 AGRATE BRIANZA MILANO ITALY
TEL: +39-039-6053-342 / FAX: +39-039-6053-206

Italy (Padova) Service Satellite
VIA 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: +44-1707-282-846 / FAX: +44-1707-27-8992

Spain Service Center
CTRA. DE RUBI, 76-80-APDO. 420
08190 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: +48-12-630-4700 / FAX: +48-12-630-4701

Turkey Service Center
ŞERIFALI MAH. NUTUK SOK. NO.5 34775
ÜMRANIYE / ISTANBUL, TURKEY
TEL: +90-216-526-3990 / FAX: +90-216-526-3995

Czech Republic Service Center
TECHNOLOGICKA 374/6,708 00 OSTRAVA-PUSTKOVEC, CZECH REPUBLIC
TEL: +420-59-5691-185 / FAX: +420-59-5691-199

Russia Service Center
213, B.NOVODMITROVSKAYA STR., 14/2, 127015 MOSCOW, RUSSIA
TEL: +7-495-748-0191 / FAX: +7-495-748-0192

Sweden Service Center
STRANDKULLEN, 718 91 FRÖVI , SWEDEN
TEL: +46-581-700-20 / FAX: +46-581-700-75

Bulgaria Service Center
4 ANDREJ LJAPCHEV BLVD. POB 21, BG-1756 SOFIA, BULGARIA
TEL: +359-2-8176009 / FAX: +359-2-9744061

Ukraine (Kharkov) Service Center
APTEKARSKIY LANE 9-A, OFFICE 3, 61001 KHARKOV, UKRAINE
TEL: +380-57-732-7774 / FAX: +380-57-731-8721

Ukraine (Kiev) Service Center
4-B, M. RASKOVOYI STR., 02660 KIEV, UKRAINE
TEL: +380-44-494-3355 / FAX: +380-44-494-3366

Belarus Service Center
Nezavisimosti pr.177, 220125 Minsk, Belarus
TEL: +375-17-393-1177 / FAX: +375-17-393-0081

South Africa Service Center
P.O. BOX 9234, EDLEEN, KEMPTON PARK GAUTENG, 1625 SOUTH AFRICA
TEL: +27-11-394-8512 / FAX: +27-11-394-8513

ASEAN

MITSUBISHI ELECTRIC ASIA PTE. LTD. (ASEAN FA CENTER)

Singapore Service Center
307 ALEXANDRA ROAD #05-01/02 MITSUBISHI ELECTRIC BUILDING SINGAPORE 159943
TEL: +65-6473-2308 / FAX: +65-6476-7439

Indonesia Service Center
THE PLAZZA OFFICE TOWER, 28TH FLOOR JL.M.H. THAMRIN KAV.28-30, JAKARTA, INDONESIA
TEL: +62-21-2992-2333 / FAX: +62-21-2992-2555

Malaysia (KL) Service Center
60, JALAN USJ 10 /1B 47620 UEP SUBANG JAYA SELANGOR DARUL EHSAN, MALAYSIA
TEL: +60-3-5631-7605 / FAX: +60-3-5631-7636

Malaysia (Johor Baru) Service Center
NO. 16, JALAN SHAH BANDAR 1, TAMAN UNGKU TUN AMINAH, 81300 SKUDAI, JOHOR MALAYSIA
TEL: +60-7-557-8218 / FAX: +60-7-557-3404

Vietnam (Ho Chi Minh) Service Center
UNIT 2408~11, 24TH FLOOR, SAIGON TRADE CENTER, 37 TON DUC THANG STREET,
DISTRICT 1, HO CHI MINH CITY, VIETNAM
TEL: +84-8-3910 5945 / FAX: +84-8-3910 5947

Vietnam (Hanoi) Service Center
SUITE 9-05, 9TH FLOOR, HANOI CENTRAL OFFICE BUILDING, 44B LY THUONG KIET STREET,
HOAN KIEM DISTRICT, HANOI CITY, VIETNAM
TEL: +84-4-3937-8075 / FAX: +84-4-3937-8076

Philippines Service Center
UNIT NO.411, ALABAMG CORPORATE CENTER KM 25. WEST SERVICE ROAD
SOUTH SUPERHIGHWAY, ALABAMG MUNTINLUPA METRO MANILA, PHILIPPINES 1771
TEL: +63-2-807-2416 / FAX: +63-2-807-2417

MITSUBISHI ELECTRIC AUTOMATION (THAILAND) CO., LTD. (THAILAND FA CENTER)

BANG-CHAN INDUSTRIAL ESTATE NO.111 SOI SERITHAI 54
T.KANNAYAO, A.KANNAYAO, BANGKOK 10230, THAILAND
TEL: +66-2906-8255 / FAX: +66-2906-3239

Thailand Service Center
898/19,20,21,22 S.V. CITY BUILDING OFFICE TOWER 1, FLOOR 7
RAMA III RD., BANGPONGPANG, YANNAWA, BANGKOK 10120, THAILAND
TEL: +66-2-682-6522 / FAX: +66-2-682-9750

INDIA

MITSUBISHI ELECTRIC INDIA PVT. LTD.

India Service Center
2nd FLOOR, TOWER A & B, DLF CYBER GREENS, DLF CYBER CITY,
DLF PHASE-III, GURGAON 122 002, HARYANA, INDIA
TEL: +91-124-4630 300 / FAX: +91-124-4630 399
Ludhiana satellite office
Jamshedpur satellite office

India (Pune) Service Center
EMERALD HOUSE, EL-3, J-BLOCK, MIDC BHOSARI. PUNE – 411 026, MAHARASHTRA, INDIA
TEL: +91-20-2710 2000 / FAX: +91-20-2710 2100
Baroda satellite office
Mumbai satellite office

India (Bangalore) Service Center
PRESTIGE EMERALD, 6TH FLOOR, MUNICIPAL NO. 2,
LAVELLE ROAD, BANGALORE - 560 043, KAMATAKA, INDIA
TEL: +91-80-4020-1600 / FAX: +91-80-4020-1699
Chennai satellite office
Coimbatore satellite office

OCEANIA

MITSUBISHI ELECTRIC AUSTRALIA LTD.

Australia Service Center
348 VICTORIA ROAD, RYDALMERE, N.S.W. 2116 AUSTRALIA
TEL: +61-2-9684-7269 / FAX: +61-2-9684-7245

CHINA

MITSUBISHI ELECTRIC AUTOMATION (CHINA) LTD. (CHINA FA CENTER)

China (Shanghai) Service Center
1-3,5-10,18-23/F, NO.1386 HONG QIAO ROAD, CHANG NING QU,
SHANGHAI 200336, CHINA
TEL: +86-21-2322-3030 / FAX: +86-21-2308-2830
China (Ningbo) Service Dealer
China (Wuxi) Service Dealer
China (Jinan) Service Dealer
China (Hangzhou) Service Dealer
China (Wuhan) Service Satellite

China (Beijing) Service Center
9/F, OFFICE TOWER 1, HENDERSON CENTER, 18 JIANGUOMENNEI DAJIE,
DONGCHENG DISTRICT, BEIJING 100005, CHINA
TEL: +86-10-6518-8830 / FAX: +86-10-6518-3907
China (Beijing) Service Dealer

China (Tianjin) Service Center
B-2 801/802, YOUYI BUILDING, NO.50 YOUYI ROAD, HEXI DISTRICT,
TIANJIN 300061, CHINA
TEL: +86-22-2813-1015 / FAX: +86-22-2813-1017
China (Shenyang) Service Satellite
China (Changchun) Service Satellite

China (Chengdu) Service Center
ROOM 407-408, OFFICE TOWER AT SHANGRI-LA CENTER, NO. 9 BINJIANG DONG ROAD,
JINJIANG DISTRICT, CHENGDU, SICHUAN 610021, CHINA
TEL: +86-28-8446-8030 / FAX: +86-28-8446-8630

China (Shenzhen) Service Center
ROOM 2512-2516, 25/F., GREAT CHINA INTERNATIONAL EXCHANGE SQUARE, JINTIAN RD.S.,
FUTIAN DISTRICT, SHENZHEN 518034, CHINA
TEL: +86-755-2399-8272 / FAX: +86-755-8218-4776
China (Xiamen) Service Dealer
China (Dongguan) Service Dealer

KOREA

MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD. (KOREA FA CENTER)

Korea Service Center
1480-6, GAYANG-DONG, GANGSEO-GU, SEOUL 157-200, KOREA
TEL: +82-2-3660-9602 / FAX: +82-2-3664-8668

Korea Taegu Service Satellite
4F KT BUILDING, 1630 SANGYEOK-DONG, BUK-KU, DAEGU 702-835, KOREA
TEL: +82-53-382-7400 / FAX: +82-53-382-7411

TAIWAN

MITSUBISHI ELECTRIC TAIWAN CO., LTD. (TAIWAN FA CENTER)

Taiwan (Taichung) Service Center
NO.8-1, GONG YEH 16TH RD., TAICHUNG INDUSTRIAL PARK, SITUN DIST.,
TAICHUNG CITY 407, TAIWAN R.O.C.
TEL: +886-4-2359-0688 / FAX: +886-4-2359-0689

Taiwan (Taipei) Service Center
10F, NO.88, SEC.6, CHUNG-SHAN N. RD., SHI LIN DIST., TAIPEI CITY 111, TAIWAN R.O.C.
TEL: +886-2-2833-5430 / FAX: +886-2-2833-5433

Taiwan (Tainan) Service Center
11F-1., NO.30, ZHONGZHENG S. ROAD, YONGKANG DISTRICT, TAINAN CITY 710, TAIWAN, R.O.C.
TEL: +886-6-252-5030 / FAX: +886-6-252-5031

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 CNC

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : TOKYO BUILDING,2-7-3 MARUNOUCHI,CHIYODA-KU,TOKYO 100-8310,JAPAN

MODEL	NC Designer
MODEL CODE	008-339
Manual No.	IB-1500040

IB(NA)1500040(ENG)-G(1303)MEE
Made in JAPAN