

MITSUBISHI CNC

MELDASMAGIC 64

CUSTOM APPLICATION INTERFACE LIBRARY GUIDE (FUNCTION SECTION)

MELDASMAGIC is a registered trademark of Mitsubishi Electric Corporation.
Microsoft and Win32 are registered trademarks of Microsoft Corporation in the USA.

Windows is a trademark of Microsoft Corporation in the USA.

The other company names and product names are trademarks or registered trademarks of the respective companies.

Introduction

This instruction manual describes the Custom Application Interface (Custom API) Library used for developing the custom applications of MELDASMAGIC Series. The Custom API Functions used when developing the custom application are described in this manual, so please read this manual and the manuals below before programming.

Custom Application Interface Library Guide	
(Programming Section)	BNP-B2197
(Variable Section)	BNP-B2199

Please read the following "Precautions for Safety" to ensure safe use of the MELDASMAGIC Series.

Precautions for Safety

Always read the specifications issued by the machine maker, this manual, related manuals and enclosed documents before starting installation, operation, programming, maintenance or inspections to ensure correct use. Thoroughly understand the basics, safety information and precautions of this numerical controller before using the unit.

The safety precautions are ranked as "DANGER", "WARNING" and "CAUTION" in this manual.

 **DANGER**

When there is a great risk that the user could be subject to fatalities or serious injuries if handling is mistaken.

 **WARNING**

When the user could be subject to fatalities or serious injuries if handling is mistaken.

 **CAUTION**

When the user could be subject to injuries or when physical damage could occur if handling is mistaken.

Note that even if the item is ranked as "  CAUTION", incorrect handling could lead to serious results. Important information is described in all cases, so please observe the items.

 **DANGER**

Not applicable in this manual.

 **WARNING**

Not applicable in this manual.

 **CAUTION**

Items related to product and manual

 For items described as "Restrictions" and "Usable State" etc., the instruction manual issued by the maker takes precedence over this manual.

 Items not described in this manual must be interpreted as "Not Possible".

 This instruction manual has been written on the assumption that all options are provided. Check the specifications issued by the machine maker before starting use.

 Some screens and functions may differ or may not be used depending on the NC system version.

Outline

This is a reference manual for the Custom Application Interface (Custom API) Library, one of the PC softwares for MELDASMAGIC Series.

The Custom API Library is an interface library for the NC Card and applications developed by the user. In the Custom Application, the setting of and reference to data inside the NC Card as well as the operation of NC are possible through the API Functions described in this manual.

This library is provided in the format of DLL (Dynamic Link Library) for Windows.

Installation of Custom API Function

The Runtime Library of Custom API Library is necessary to use the Custom API function. Refer to the "MELDASMAGIC64 Custom Application Interface Library Guide (Programming Section) (BNP-B2191)" for an outline of the Runtime Library. Refer to "MELDASMAGIC64 Setup Instruction Manual (BNP-B2191)" for the installation method.

The files below, which are on the "Custom API Library" floppy disk, are necessary to create applications using the Custom API function.

- | | |
|--------------|---|
| • nmcapi.h | Custom API function definition file |
| • melncapi.h | Custom API function macro definition file |
| • meltype.h | Custom API data type definition file |
| • melerr.h | Custom API error code definition file |
| • melsberr.h | Custom API sub-error code definition file |

Include the above mentioned header files into the source file. Refer to "MELDASMAGIC64 Custom Application Interface Library Guide (Programming Section) (BNP-B2191)" for the "Custom API Library" installation method.

How to Read the Manual

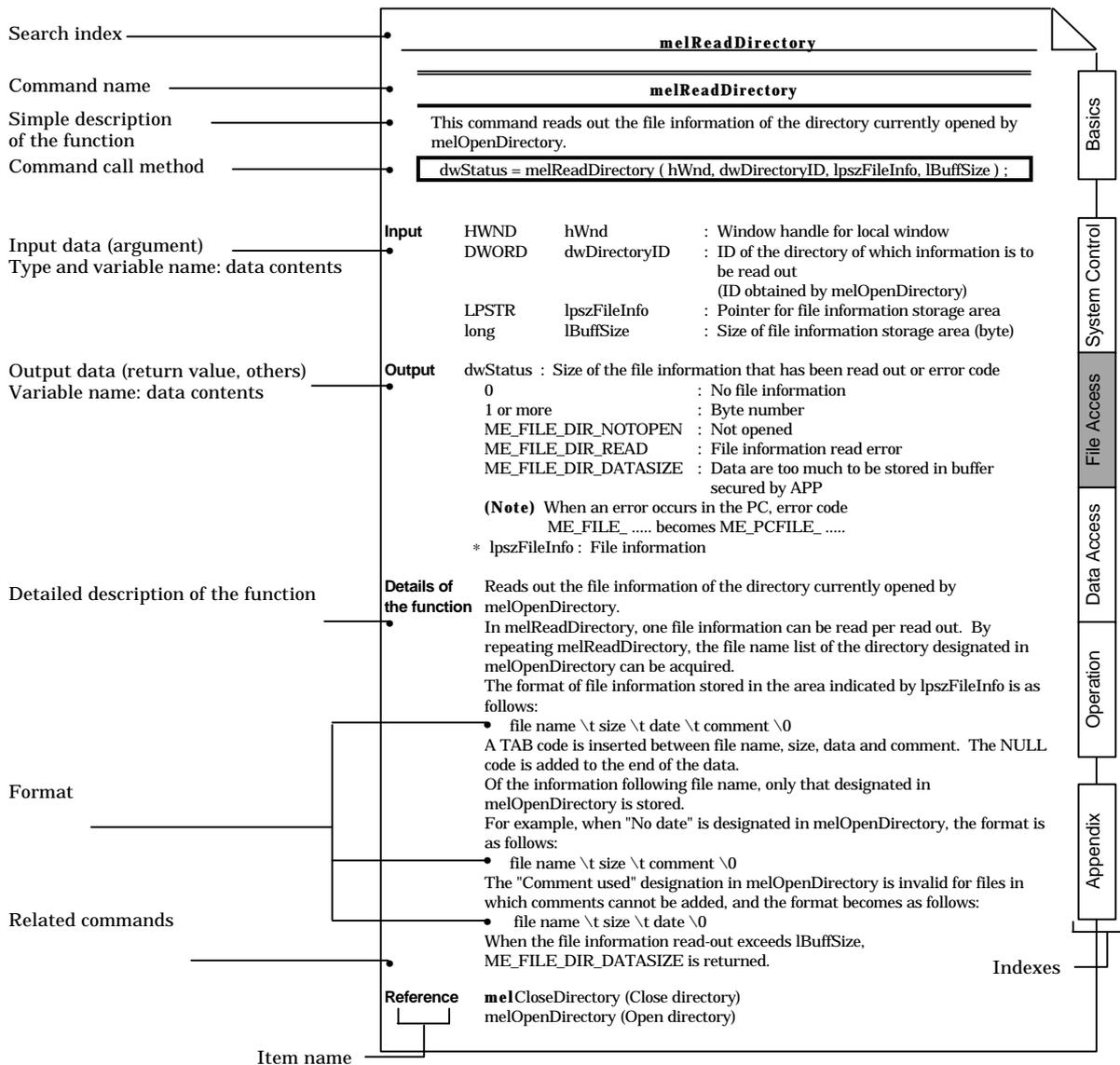
Organization

First, "1. API Function Basics" describes the basics of programming using the API Function. Next, "2. Commands" describes the commands of each command classification in alphabetical order. "3. Appendix" provides tables of all the command names and error codes listed in this manual. Indexes for "Basics", "Command classification" for command description and "Appendix" are indicated on the the right-hand edge of each right page. When searching for specific items, refer to the upper corner of each page for the basic information item, command name or table name provided in each page.

How the commands are described

Each command description includes the following:

1. Simple description of the command following the command name
2. Command call method framed in bold line, with the variable name indicated in italics
3. Input, output data and details of function described, with the Item name indicated in Extra boldface letters Format indicated in Boldface letters
4. "Reference" presenting related commands



1. API Function Basics

1.1 Designation of File Name

In Custom API Library, the NC Card is considered as a single drive, and various data on NC Card (machining program, tool offset etc.) are handled as files.

When accessing a NC Card file with the Custom API Library, designate the file name with an absolute path as:

drive name + ":" + directory name + file name

Example) Designate the 123.PRG file in the \PRG\USER\ directory of the NC Card with the NC Card No. "01".

(1) Designation of Drive Name

Communication in the Custom API Library is carried out with NC Card considered as a single drive. Each NC Card has a NC Card No. (established when NC Card is set up), and each drive name corresponds to NC Card No. indicated below.

NC Card No.	Drive name
01	M01
02	M02
03	M03
04	M04
05	M05
06	M06
07	M07
08	M08
:	:
:	:
FF	MFF

Designation of File Name

(2) Designation of Directory/File

The following is the directory configuration of the NC Card.
Note that the directory does not accept additions or deletions.

\PRG\USER\	Machining program storage directory
\PRG\FIX\	Fixed cycle program storage directory
\PRG\MDI\	MDI program storage directory
\PRM\	Parameter file directory
\LAD\	PLC program directory
\REG\	PLC device data directory
\DAT\	Work offset, tool offset, common variable storage directory
\LOG\	History data directory (for maintenance)

Each directory contains the following files:

\PRG\USER\	Machining program storage directory
Program number.PRG	: Machining program file (User program, sub-program, macro program) (Each machining program is considered as a separate file)
\PRG\FIX\	Fixed cycle program storage directory
Program number.PRG	: Fixed cycle program file (Each fixed cycle program is considered as a separate file)
\PRG\MDI\	MDI program storage directory
MDI.PRG	: MDI program file
\PRM\	Parameter file directory
ALL.PRM	: Parameter file
PARAMET.BIN	: Parameter file (for maintenance)
FILESYS.BIN	: File system file
\LAD\	PLC program directory
USERPLC.LAD	: User PLC program file
\REG\	PLC device data directory
RREG.REG	: R register data file
TREG.REG	: T (timer) register data file
CREG.REG	: C (counter) register data file
\DAT\	Work offset, tool offset, tool life data, common variable storage directory
WORK.OFS	: Work offset data file
TOOL.OFS	: Tool offset data file
TOOLLIFE.TLF	: Tool life management data file
COMMON.VAR	: Common variable data file
Group No. (8 digits). TL1	: Tool life management data file for each tool group.
Group No. (8 digits). TL2	: Tool life management data file for each tool group.
\LOG\	History data directory (for maintenance)
ILLEGAL.ERR	

Designation of File Name

For some files, there are some restrictions in copying or compare operation.

[Files that cannot be deleted]

\PRM\ Parameter file directory
 ALL.PRM : Parameter file
 PARAMET.BIN : Parameter file (for maintenance)
 FILESYS.BIN : File system file

[Files whose data is initialized or becomes the default when deleted]

\PRG\MDI\ MDI program storage directory
 MDI.PRG..... Changes into file with file size 0.

\LAD\ PLC program directory
 USERPLC.LAD..... Changes into PLC program with 0 step numbers.

\REG\ PLC device data directory
 RREG.REG..... All R register data turns to 0.
 TREG.REG..... All T(timer) register data turns to 0.
 CREG.REG..... All C(counter) register data turns to 0.

\DAT\ Work offset, tool offset, tool life data, common variable storage directory
 WORK.OFS..... All work offset data turns to 0.
 TOOL.OFS..... All tool offset data turns to 0.
 TOOLLIFE.TLF..... All tool life management data turns to 0.
 COMMON.VAR..... All common variable data turns to 0.

[Files that bring some errors when files are overwritten]

\PRG\USER\ Machining program storage directory
 Program number.PRG : Machining program file
 (User program, sub-program, macro program)
 (Each machining program is considered as a separate file)

\PRG\FIX\ Fixed cycle program storage directory
 Program number.PRG : Fixed cycle program file
 (Each fixed cycle program is considered as a separate file)

\PRG\MDI\ MDI program storage directory
 MDI.PRG : MDI program file

[Files that can be verified]

\PRM\ Parameter file directory
 ALL.PRM : Parameter file

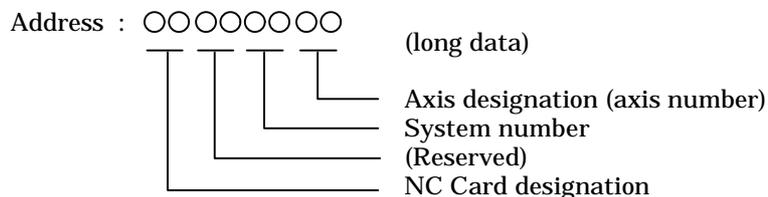
\DAT\ Workpiece offset, tool offset, tool life data, common variable storage
 directory
 WORK.OFS : Workpiece offset data file
 TOOL.OFS : Tool offset data file
 COMMON.VAR : Common variable data file

Files other than those above cannot be verified. An error will result if verification of a file other than one of the above is attempted.

1.2 Address

In each function, the NC Card, system and axis for the operation are designated using the address, one of arguments.

An address consists of 32-bit data with the following data structure:



When in actual use, use the following macro, and command the specified logical sum (|) that is necessary. Note that there are no rules for the order of logical sum calculation.

Address designation macro

Axis designation	ADR_AXIS (n)
System designation	ADR_SYSTEM (n)
NC Card designation	ADR_MACHINE (n)

Example) Designation of 2nd system and 1st axis

ADR_SYSTEM (2) | ADR_AXIS (1)

Example) Designation of NC Card 2nd and 6th axis

ADR_MACHINE (2) | ADR_AXIS (6)

The following is the relationship between NC Card number and NC Card designation.

NC Card No.	NC Card designation
01	01
02	02
03	03
04	04
05	05
06	06
07	07
08	08
⋮	⋮
⋮	⋮
FF	FF

What needs to be designated in each address depends on each function.

1.3 Designation of Data Type

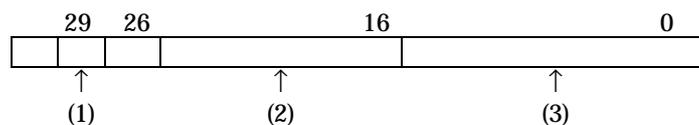
The data type used in each function includes the following:

Data type	Data area
(1) 1-byte integer type (T_CHAR)	char
(2) 2-byte integer type (T_SHORT)	short
(3) 4-byte integer type (T_LONG)	long
(4) 8-byte real number type (T_DOUBLE)	double
(5) Character string type (T_STR)	<pre> struct STRINGTYPE { long IBufferSize; // Buffer size of lpszBuff LPSTR lpszBuff; // Buffer pointer }; </pre>
(6) Decimal integer character string type (T_DECSTR)	STRINGTYPE
(7) Hexadecimal character string type (T_HEXSTR)	STRINGTYPE
(8) Binary character string type (T_BINSTR)	STRINGTYPE
(9) Real number character string type (T_FLOATSTR)	<pre> struct FLOATSTR { short nIntDataNos; // Number of integer digits short nDeciDataNos; // Number of decimal digits long IOption; // Option // FLTSTR_DECI_ZERO_SUPPRESS // Decimal suppress zero // FLTSTR_ADD_PLUS_SIGN + Add // + symbol long IBufferSize; // Buffer size of lpszBuff LPSTR lpszBuff; // Buffer pointer }; </pre>
(10) Special type: For getting current program (T_GETPRGBLOCK).	
Used in the melGetCurrentPrgBlock.	
	<pre> struct GETPRGBLOCK { int iCurrentBlockNum; // Current block // (Block in the acquired data) // 0: Operation not in progress // 1: 1st block // 2: 2nd block long IPrgDataSize; // Buffer size of lpszPrgData LPSTR lpszPrgData; // Pointer for buffer storing the program }; </pre>

When using this type in each function, secure a data area in the application, and specify the pointer for the secured data area and data type in the argument.

Return Value of the Function

The sub-error code consists of 32-bit data with the following structure.



(1) Error flag 0 : no sub-error 1 : sub-error

(2) Sub-error No.

(3) Additional information

(1) When it is not an error

When each function is not an error, error flag is set to 0.

(2) When it is an error

When there is a sub-error, the error flag is set to 1.

The sub-error No. and additional information have the following meaning:

Sub-error No. : Error No.

Additional information : Error attribute information, etc.

(Usage method differs with each API function)

Refer to the definition file (melsberr.h) of the sub-error code for concrete details about the sub-error.

Refer to the command explanation of each API function for information on whether the API function being used supports the sub-error.

1.5 NC Data Access Variable

In the Custom API Library, various data in NC Card are accessed as variables. These variables are called NC Data Access Variables.

The NC Data Access Variable indicates a single data with Section No. and Sub-section No. Depending on the variable, it may contain system and axis data. For such data, the data is specified by Section No., Sub-section No. and address designation.

For details on the NC Data Access Variables, refer to the section "Custom Application Interface Library Guide (Variable Section) BNP-B2143".

2. Commands

2.1 System Control Commands

mellIoctl

This command executes functions such as mode and parameter setting and status information acquisition for a specified 2-port driver.

dwStatus = mellIoctl (*hWnd*, *lAddress*, *lFunction*, *lpData*) ;

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address
			Designate the following items:
			• NC Card designation
	long	<i>lFunction</i>	: Function code
	LPVOID	<i>lpData</i>	: Data

Output	DWORD <i>dwStatus</i> : Error code
	0 : Normal completion
	ME_SYSFUNC_IOCTL_ADDR : Address illegal
	ME_SYSFUNC_IOCTL_DATA : Data range illegal
	ME_SYSFUNC_IOCTL_FUNCTION : Command illegal
	ME_SYSFUNC_IOCTL_NOTOPEN : Device not opened

Details of the function The following is a list of and details of each function.

○ Function list

- | | | |
|----------------------|---------|-------------------------|
| (1) DEV_SET_TIMEOUT | (0x001) | Read timeout setting |
| (2) DEV_SET_WTIMEOUT | (0x002) | Write timeout setting |
| (3) DEV_CANCEL_RWAIT | (0x003) | Read wait state cancel |
| (4) DEV_CANCEL_WWAIT | (0x004) | Write wait state cancel |

o Details of the function

(1) DEV_SET_RTIMEOUT (0x001) Read timeout setting

Sets the timeout value for reading in the message from reception cue.

Timeout does not occur when the timeout value is set to 0.

Instead, it waits until the message is read in from reception cue.

Default is 50 (5 seconds)

lFunction : DEV_SET_RTIMEOUT
lpData : Long type pointer
0 : No timeout
1 or more : Timeout value (unit: 100ms)

(2) DEV_SET_WTIMEOUT (0x002) Write timeout setting

Sets the timeout value when transmission cue is full and the message cannot be written in.

Timeout does not occur when the timeout value is set to 0.

Instead, it waits until there is a space in the transmission cue.

Default is 50 (5 seconds).

lfunction : DEV_SET_WTIMEOUT
lpData : Long type pointer.
0 : No timeout
1 or more : Timeout value (unit: 100ms)

(3) DEV_CANCEL_RWAIT (0x003) Read wait state cancel

Cancels the read wait state. Read returns as an error. This function is insignificant when not in the read wait state. Use it to cancel a read that is in progress.

lfunction : DEV_CANCEL_RWAIT
lpData : Not used (specify NULL)

(4) DEV_CANCEL_WWAIT (0x004) Write wait state cancel

Cancels the write wait state. Write returns as an error. This function is insignificant when not in the write wait state. Use it to cancel a write that is in progress.

lfunction : DEV_CANCEL_WWAIT
lpData : Not used (specify NULL)

melCloseDirectory

2.2 Commands Related to File Access

melCloseDirectory

This command closes a directory opened by melOpenDirectory.

<code>dwStatus = melCloseDirectory (hWnd, dwDirectoryID);</code>
--

Input	HWND <i>hWnd</i> : indow handle for local window DWORD <i>dwDirectoryID</i> : ID of the directory to be closed (ID obtained by melOpenDirectory)
Output	DWORD <i>dwStatus</i> : Error code 0 : Normal completion ME_FILE_DIR_NOTOPEN : Directory not opened (Note) When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_
Details of the function	Closes the directory opened by melOpenDirectory.
Referenceme	melOpenDirectory (Open directory) melReadDirectory (Read file information)

melCopyFile

melCopyFile

This command copies files.

```
dwStatus = melCopyFile ( hWnd, lpszSrcFileName, lpszDstFileName ) ;
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	LPSTR	<i>lpszSrcFileName</i>	: Copy source file name
	LPSTR	<i>lpszDstFileName</i>	: Copy destination file name
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_FILE_COPY_BUSY	: Now copying this file not permitted (program running)
		ME_FILE_COPY_ENTRYOVER	: Number of entry over
		ME_FILE_COPY_FILEEXIST	: Copy destination file already exists
		ME_FILE_COPY_FILESYSTEM	: There is something wrong in file system
		ME_FILE_COPY_ILLEGALNAME	: Format for file name illegal
		ME_FILE_COPY_MEMORYOVER	: Memory capacity over
		ME_FILE_COPY_NODIR	: Directory does not exist
		ME_FILE_COPY_NODRIVE	: Drive does not exist
		ME_FILE_COPY_NOFILE	: File does not exist
		ME_FILE_COPY_PLCRUN	: Now copying this file not permitted (PLC running)
		ME_FILE_COPY_READ	: Copy source file cannot be read out
		ME_FILE_COPY_WRITE	: Copy destination file cannot be written in
		ME_PCFILE_COPY_CREATE	: File cannot be created (PC only)
		ME_PCFILE_COPY_OPEN	: File cannot be opened (PC only)
		(Note)	When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_

Details of the function

Copies the file designated in *lpszSrcFileName* into the file designated in *lpszDstFileName*.

Designate file name with an absolute path as:

drive name + ":" + directory name + file name

The file name can be omitted in *lpszDstFileName*. When the file name is omitted, it is copied with the same name.

lpszDstFileName should not be an existing file name.

Furthermore, the copy destination directory should already exist.

If an error occurs when copying a file, it is possible to retrieve more detailed error information. This detailed error information is called a sub-error code. The sub-error codes are retrieved using GetLastError of Windows API.

```
dwSubErrorCode = GetLastError ( ) ;
```

Using the retrieved sub-error code and the macro below, it is possible to retrieve the sub-error No. (*dwSubErrorNum*) and line of the file where the error occurred (*dwErrorLineNum*).

```
dwSubErrorNum = RetCopyFileSubError ( dwSubErrorCode ) ;
```

```
dwErrorLineNum = RetCopyFileErrorLine ( dwSubErrorCode ) ;
```

The sub-error Nos. and meanings are shown below.

ME_FILE_COPY_READ Sub-error No. when there is an error

ME_SUB_FILE_COPY_BUSY : File output was attempted during data input/output

ME_FILE_COPY_WRITE Sub-error No. when there is an error

ME_SUB_FILE_COPY_BUSY : File input was attempted during data input/output

ME_SUB_FILE_COPY_FORMAT : Error in the format

ME_SUB_FILE_COPY_PARAMETERNO : Error in the parameter No.

ME_SUB_FILE_COPY_TAG : Error in the tag

ME_SUB_FILE_COPY_TOOLOFFSETNUM : Error in the designation of the tool compensation No.

ME_SUB_FILE_COPY_TOOLOFFSETDATA : Error in the command of the tool compensation data

ME_SUB_FILE_COPY_OFFSETTYPE : Error in the designation of the offset type

ME_SUB_FILE_COPY_DATARANGE : Error in the data range

This API function does not check if designated directory or file name is proper or not. Check directory or file name for such irregular operation as like; transferring between files that type or purpose differs from another (Ex. overwrite parameter file (PARAMET.BIN) on the user program file (\PRG\USER_.PRG)), or copying a file to the directory in spite that has different purpose, etc.

Reference

melDeleteFile (Delete file)

melRenameFile (Rename file)

melVerifyFile (File Verify)

melDeleteFile

melDeleteFile

This command deletes a designated file.

```
dwStatus = melDeleteFile ( hWnd, lpzFileName ) ;
```

Input	HWND <i>hWnd</i> : Window handle for local window LPSTR <i>lpzFileName</i> : File name
Output	DWPRD <i>dwStatus</i> : Error code 0 : Normal completion ME_FILE_DEL_BUSY : Now deleting this file not permitted (program running) ME_FILE_DEL_FILESYSTEM : There is something wrong in file system ME_FILE_DEL_ILLEGALNAME : Format for file name illegal ME_FILE_DEL_NODIR : Directory does not exist ME_FILE_DEL_NODRIVE : Drive does not exist ME_FILE_DEL_NOFILE : File does not exist ME_FILE_DEL_NOTDELETE : Deleting this file is not permitted
	(Note) When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_
Details of the function	Deletes the file designated in <i>lpzFileName</i> . Designate the file name with an absolute path as: drive name + ":" + directory name + file name
Reference	melCopyFile (Copy file) melRenameFile (Rename file) melVerifyFile (File Verify)

melGetDiskFree

melGetDiskFree

This command gets size of free space of the drive or directory.

```
dwStatus = melGetDiskFree ( hWnd, lpszDirectoryName ) ;
```

Input	HWND <i>hWnd</i> : Window handle for local window LPSTR <i>lpszDirectoryName</i> : Directory name
Output	DWORD <i>dwStatus</i> : Size of free space or error code 0 or more : Size of free space (bytes) ME_FILE_DISKFREE_FILESYSTEM : There is something wrong in file system ME_FILE_DISKFREE_NODIR : Directory does not exist ME_FILE_DISKFREE_NODRIVE : Drive does not exist (Note) When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_
Details of the function	Returns the size of free space of the directory designated in <i>lpszDirectoryName</i> . Byte is the unit for indicating the size of free space. Designate the directory with an absolute path as: drive name + ":" + directory name When the drive of PC is designated as a drive name, the directory designation is ignored and the size of free space of the drive is returned. When an NC Card is designated as a drive name, the size of free space of the designated directory is returned. When a sub-directory exists in the designated directory, the usage amount of the sub-directory is not included in the size of free space.
Reference	melCloseDirectory (Close directory) melGetDriveList (Get drive list) melOpenDirectory (Open directory) melReadDirectory (Read file information)

melGetDriveList

melGetDriveList

This command gets the mounted NC Card as the drive list.

```
dwStatus = melGetDriveList ( hWnd, lpszDriveList, lBuffSize ) ;
```

Input	HWND <i>hWnd</i> : Window handle of local window LPSTR <i>lpszDriveList</i> : Pointer for file information storage area long <i>lBuffSize</i> : Size of file information storage area (byte)
Output	DWORD <i>dwStatus</i> : Size of the drive list acquired or error code 0 : No drive list 1 or more : Byte number ME_FILE_DRVLIST_DATASIZE : Data are too much to be stored in buffer secured by APP ME_FILE_DRVLIST_READ : Drive list read error (Note) When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_ * <i>lpszDriveList</i> : Drive list
Details of the function	Reads out the currently connected NC Card as the drive list. The following is the format of the drive list stored in the area indicated by <i>lpszDriveList</i> . drive name: CRLF drive name: CRLF... drive name:CRLF\0 CR and LF codes are inserted between the drive names. CR,LF,NULL codes are added to the end of the data. If the drive list that is read out is more than the <i>lBuffSize</i> , <i>lBuffSize</i> worth of data is stored in the area indicated by <i>lpszDriveList</i> , and ME_FILE_DRVLIST_DATASIZE is returned. It does not read the drive list of PC.
Reference	melCloseDirectory (Close directory) melGetDiskFree (Get size of free space) melOpenDirectory (Open directory) melReadDirectory (Read file information)

Details of the function	<p>Opens the designated directory.</p> <p>After the directory has been opened, its file and directory information can be acquired using melReadDirectory.</p> <p>The open state is valid until melCloseDirectory is executed.</p> <p>When the directory is successfully opened, the directory ID is obtained.</p> <p>The directory ID is necessary when using melReadDirectory and melCloseDirectory.</p> <p>The directory ID is valid until melCloseDirectory is executed.</p> <p>Only one directory can be opened at a time.</p> <p>Designate the directory with an absolute path as:</p> <p style="padding-left: 40px;">drive name + ":" + directory name + file name</p> <p>When the file name is omitted, all of the files in the directory become the subject of read-out. It is possible to use "?" and "*" (wild cards) in the file name so that it corresponds with some files in the directory. Make sure to add "\" at the end of directory path when omitting the file name.</p> <p>Both the NC Card and PC can be designated as the drive name.</p> <p>However, when designating the directory of PC, designation of "No comment" or "Comment used" in <i>IFileType</i> is invalid.</p> <p>• Example of directory designation</p> <p style="padding-left: 40px;">Example 1) Parameter directory designation of NC Card No.1</p> <p style="padding-left: 80px;">M01 : \PRM\</p> <p style="padding-left: 40px;">In melReadDirectory, all of the files following M01:\PRM\ become the object of readout.</p> <p style="padding-left: 40px;">Example 2) Program directory designation of NC Card No.2</p> <p style="padding-left: 80px;">M02 : \PRG\USER*.PRG</p> <p style="padding-left: 40px;">In melReadDirectory, files following M02: \PRG\USER\ and possessing PRG become the object of readout.</p>
Reference	<p>melCloseDirectory (Close directory)</p> <p>melReadDirectory (Read file information)</p>

melReadDirectory

melReadDirectory

This command reads out the file information of the directory currently opened by melOpenDirectory.

```
dwStatus = melReadDirectory ( hWnd, dwDirectoryID, lpszFileInfo, lBuffSize );
```

Input

HWND	<i>hWnd</i>	:	Window handle for local window
DWORD	<i>dwDirectoryID</i>	:	ID of the directory of which information is to be read out (ID obtained by melOpenDirectory)
LPSTR	<i>lpszFileInfo</i>	:	Pointer for file information storage area
long	<i>lBuffSize</i>	:	Size of file information storage area (byte)

Output

DWORD *dwStatus*: Size of the file information that has been read out or error code

0	:	No file information
1 or more	:	Byte number
ME_FILE_DIR_DATASIZE	:	Data are too much to be stored in buffer secured by APP
ME_FILE_DIR_NOTOPEN	:	Not opened
ME_FILE_DIR_READ	:	File information read error

(Note) When an error occurs in the PC, error code ME_FILE_ becomes ME_PCFILE_

* *lpszFileInfo*: File information

Details of the function

Reads out the file information of the directory currently opened by melOpenDirectory. In melReadDirectory, one file information can be read per read out. By repeating melReadDirectory, the file name list of the directory designated in melOpenDirectory can be acquired.

The format of file information stored in the area indicated by *lpszFileInfo* is as follows:

file name \t size \t date \t comment \0

A TAB code is inserted between file name, size, date and comment. The NULL code is added to the end of the data.

Of the information following file name, only that designated in melOpenDirectory is stored.

For example, when "No date" is designated in melOpenDirectory, the format is as follows:

file name \t size \t comment \0

The "Comment used" designation in melOpenDirectory is invalid for files in which comments cannot be added, and the format becomes as follows:

file name \t size \t date \0

When the file information read-out exceeds lBuffSize, ME_FILE_DIR_DATASIZE is returned.

Reference

melCloseDirectory (Close directory)
melOpenDirectory (Open directory)

melRenameFile

melRenameFile

This command changes the file name.

```
dwStatus = melRenameFile ( hWnd, lpszSrcFileName, lpszDstFileName ) ;
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	LPSTR	<i>lpszSrcFileName</i>	: Old file name
	LPSTR	<i>lpszDstFileName</i>	: New file name
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_FILE_REN_BUSY	: Now renaming not permitted (program running)
		ME_FILE_REN_FILEEXIST	: New file name already exists
		ME_FILE_REN_FILESYSTEM	: There is something wrong in file system
		ME_FILE_REN_ILLEGALNAME	: Format for file name illegal
		ME_FILE_REN_NODIR	: Directory does not exist
		ME_FILE_REN_NODRIVE	: Drive does not exist
		ME_FILE_REN_NOFILE	: File does not exist
		ME_FILE_REN_NOTRENAME	: Renaming this file is not permitted
		ME_FILE_REN_SAMENAME	: New name and old one of file are the same

(Note) When an error occurs in the PC, error code
ME_FILE_ becomes ME_PCFILE_

Details of the function Changes the file name designated in *lpszSrcFileName* to file name designated in *lpszDstFileName*.
Designate the old file name in *lpszSrcFileName* with an absolute path as:
drive name + ":" + directory name + file name
Designate only the new file name in *lpszDstFileName* excluding the drive name or directory name. *lpszDstFileName* cannot be an existing file name.

Reference melCopyFile (Copy file)
melDeleteFile (Delete file)
melVerifyFile (File Verify)

melVerifyFile

melVerifyFile

This command verifies the file name.

```
dwStatus = melVerifyFile ( hWnd, lpzSrcFileName, lpzDstFileName );
```

Input

HWND	<i>hWnd</i>	: Window handle for local window
LPSTR	<i>lpzSrcFileName</i>	: Verification source file name (NC side file)
LPSTR	<i>lpzDstFileName</i>	: Verification destination file name (PC side file)

Output

DWORD *dwStatus*: Error code

0	: Normal completion
ME_FILE_VERIFY_DIFFER	: Difference in the verification
ME_FILE_VERIFY_FILESYSTEM	: There is something wrong in the file system
ME_FILE_VERIFY_ILLEGALNAME	: Format for the file name is illegal
ME_FILE_VERIFY_NODIR	: Directory does not exist
ME_FILE_VERIFY_NODRIVE	: Drive does not exist
ME_FILE_VERIFY_NOFILE	: File does not exist
ME_FILE_VERIFY_NOTSUPPORTED	: Verification function is not supported
ME_FILE_VERIFY_NAMELENGTH	: File name is too long
ME_FILE_VERIFY_OPEN	: File cannot be opened

(Note) When an error occurs on the personal computer side, ME_FILE_.... of the error code becomes ME_PCFILE_

Details of the function

Compares the file designated *lpzSrcFileName* and the file designated *lpzDstFileName*.

Designate the file name with an absolute path as:

drive name + ":" + directory name + file name

Designate the NC side file in *lpzSrcFileName*, and the PC side file in *lpzDstFileName*.

If an error occurs when verifying a file, it is possible to retrieve more detailed error information. This detailed error information is called a sub-error code. The sub-error codes are retrieved using GetLastError of Windows API.

```
dwSubErrorCode = GetLastError ( );
```

Using the retrieved sub-error code and the macro below, it is possible to retrieve the sub-error No. (*dwSubErrorNum*) and line of the file where the error occurred (*dwErrorLineNum*).

```
dwSubErrorNum = RetCopyFileSubError ( dwSubErrorCode );
```

```
dwErrorLineNum = RetCopyFileErrorLine ( dwSubErrorCode );
```

The sub-error Nos. and meanings are listed below.

ME_FILE_VERIFY_DIFFER	Sub-error No. when there is an error
ME_SUB_FILE_COPY_NONE	: Difference in the file
ME_SUB_FILE_COPY_FORMAT	: Error in the format
ME_SUB_FILE_COPY_PARAMETERNO	: Error in the parameter No.
ME_SUB_FILE_COPY_TAG	: Error in the tag
ME_SUB_FILE_COPY_TOOLOFFSETNUM	: Error in the designation of the tool compensation No.
ME_SUB_FILE_COPY_TOOLOFFSETDATA	: Error in the command of the tool compensation data

melVerifyFile

ME_SUB_FILE_COPY_OFFSETTYPE : Error in the designation of the
offset type
ME_SUB_FILE_COPY_DATARANGE : Error in the data range

Reference

melCopyFile (Copy file)
melDeleteFile (Delete file)
melRenameFile (Rename file)

melCancelModal

2.3 Commands Related to Data Access

melCancelModal

This command cancels registration as high-speed read-out data.

```
dwStatus = melCancelModal ( hWnd, lAddress, lModalId );
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation
	long	<i>lModalId</i>	: ID of the data of which registration is to be cancelled (ID obtained by melRegisterModal)
Output	DWORD	<i>dwStatus</i>	: Error code
	0		: Normal completion
	ME_DATA_MDLCANCEL_ADDR		: Address illegal
	ME_DATA_MDLCANCEL_NOTREGIST		: Modal ID not registered
Details of the function	Cancels high-speed read-out registration carried out by melRegisterModal. Designate the data ID to be cancelled in <i>lModalId</i> .		
Reference	melReadModal (High-speed read-out of data)		
	melRegisterModal (Register high-speed read-out data)		

melReadData

melReadData

This command reads out data.

```
dwStatus = melReadData ( hWnd, lAddress, lSectionNum, lSubSectionNum,  
                        lpReadData, lReadType ) ;
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address
			Designate the following items:
			• NC Card designation
			• System designation (Necessary for some data)
			• Axis designation (Necessary for some data)
		long	<i>lSectionNum</i>
	long	<i>lSubSectionNum</i>	: Sub-section No.
	LPVOID	<i>lpReadData</i>	: Pointer for the area in which data is to be stored
	long	<i>lReadType</i>	: Data type
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_DATA_READ_ADDR	: Address illegal
		ME_DATA_READ_DATASIZE	: Data are too much to be stored in buffer secured by APP
		ME_DATA_READ_DATATYPE	: Data type illegal
		ME_DATA_READ_READ	: Now reading data not permitted
		ME_DATA_READ_SECT	: Section No. illegal
	ME_DATA_READ_SUBSECT	: Sub-section No. illegal	
	ME_DATA_READ_WRITEONLY	: Write only data	
	* <i>lpReadData</i> : Data that has been read out		
Details of the function	Reads out the data designated in <i>lAddress</i> , <i>lSectionNum</i> , <i>lSubSectionNum</i> .		
	The data that is read out is converted into a type designated in <i>lReadType</i> , and is stored in the area indicated by <i>lpReadData</i> .		
Reference	melReadModal (High-speed read-out of data)		
	melWriteData (Write data)		

melReadModal

melReadModal

This command executes high-speed data read-out.

<code>dwStatus = melReadModal (hWnd, lAddress, lModalId, lpReadData, lReadType) ;</code>
--

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation
	long	<i>lModalId</i>	: ID of the data to be read out (obtained by melRegisterModal)
	LPVOID	<i>lpReadData</i>	: Pointer for the area in which data is to be stored
	long	<i>lReadType</i>	: Data type
Output	DWORD	<i>dwStatus</i>	: Error code
	0		: Normal completion
	ME_DATA_MDLREAD_ADDR		: Address illegal
	ME_DATA_MDLREAD_DATASIZE		: Data are too much to be stored in buffer secured by APP
	ME_DATA_MDLREAD_DATATYPE		: Data type illegal
	ME_DATA_MDLREAD_NOTREGIST		: Modal ID not registered
	ME_DATA_MDLREAD_READ		: Now reading data not permitted
ME_DATA_MDLREAD_WRITEONLY		: Write only data	
	* <i>lpReadData</i> : Data that has been read out		
Details of the function	Reads out the high-speed read-out data registered by melRegisterModal to the area indicated by <i>lpReadData</i> . Designate the data ID obtained by melRegisterModal in <i>lModalId</i> .		
Reference	melCancelModal (Cancel registration of high-speed read-out data) melReadData (Read-out data) melRegisterModal (Register high-speed read-out data)		

melRegisterModal

melRegisterModal

This command registers as high-speed (constant) read-out data.

`dwStatus = melRegisterModal (hWnd, lAddress, lSectionNum, lSubSectionNum, lPriority) ;`

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation• System designation (Necessary for some data)• Axis designation (Necessary for some data)
	long	<i>lSectionNum</i>	: Section No.
	long	<i>lSubSectionNum</i>	: Sub-section No.
	long	<i>lPriority</i>	: Priority order 1: Highest 2: High 3: Middle 4: Low
Output	DWORD	<i>dwStatus</i>	: ID of the data registered or error code
	0 or more		: Data ID (used in melCancelModal and melReadModal)
	ME_DATA_MDLREGIST_ADDR		: Address illegal
	ME_DATA_MDLREGIST_PRIORITY		: Priority designation illegal
	ME_DATA_MDLREGIST_REGIST		: Number of data registration over
ME_DATA_MDLREGIST_SECT		: Section No. illegal	
ME_DATA_MDLREGIST_SUBSECT		: Sub-section No. illegal	
Details of the function	Registers the data designated in <i>lAddress</i> , <i>lSectionNum</i> , and <i>lSubSectionNum</i> as high-speed read-out data. When registered properly, melRegisterModal returns the ID for that data. The data can be read out at a high speed by melReadModal using this data ID. An order of priority can be assigned to the registered data through <i>lPriority</i> . When a high priority is given, the data is processed preferentially in the NC Card. Thus, data needing a short updating cycle should be given a high priority. However, since the order of priority order is relative to the data registered, if there are too many data items registered at a high priority, it may lead to a delayed updating cycle. The maximum number of data that can be registered for high-speed read-out is 128. An error occurs when more than 128 data are registered.		
Reference	melCancelModal (Cancel registration of high-speed read-out data) melReadModal (High-speed read-out of data)		

melWriteData

melWriteData

This command writes in the data.

<pre>dwStatus = melWriteData (hWnd, lAddress, lSectionNum, lSubSectionNum, lpWriteData, lWriteType) ;</pre>

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address
			Designate the following items:
			• NC Card designation
			• System designation (Necessary for some data)
			• Axis designation (Necessary for some data)
		long	<i>lSectionNum</i>
	long	<i>lSubSectionNum</i>	: Sub-section No.
	LPVOID	<i>lpWriteData</i>	: Pointer for the data area
	long	<i>lWriteType</i>	: Data type
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_DATA_WRITE_ADDR	: Address illegal
		ME_DATA_WRITE_DATASIZE	: Data are too much to be stored in buffer secured by APP
		ME_DATA_WRITE_DATATYPE	: Data type illegal
		ME_DATA_WRITE_READONLY	: Read only data
		ME_DATA_WRITE_SECT	: Section No. illegal
		ME_DATA_WRITE_SUBSECT	: Sub-section No. illegal
	ME_DATA_WRITE_WRITE	: Now writing data not permitted	
Details of the function			Writes in the data designated in <i>lAddress</i> , <i>lSectionNum</i> , and <i>lSubSectionNum</i> .
			Set the value to be written in at <i>lpWriteData</i> and designate the type of <i>lpWriteData</i> in <i>lWriteType</i> .
Reference			melReadData (Read-out data)

2.4 Commands Related to Operation

melActivatePLC

Controls the startup and end of PLC program.

```
dwStatus = melActivatePLC ( hWnd, lAddress, lActivePLC );
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation
	long	<i>lActivePLC</i>	: PLC operation mode
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_OPE_ACTPLC_ADDR	: Address illegal
		ME_OPE_ACTPLC_MODE	: Mode illegal
Details of the function			Controls the startup and end of the PLC program in NC Card designated in <i>lAddress</i> . The modes of <i>lActivePLC</i> are as follows:
		M_OPE_ACTPLC_TRUE	: PLC startup
		M_OPE_ACTPLC_FALSE	: PLC end

melGetCurrentAlarmMsg

melGetCurrentAlarmMsg

This command gets the currently occurring alarm message.

```
dwStatus = melGetCurrentAlarmMsg ( hWnd, lAddress, lMsgNos, lAlarmType,
                                   lpAlarmMsg, lReadType );
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation• System Designation (all systems when 0 is set)
	long	<i>lMsgNos</i>	: Number of messages to be gotten
	long	<i>lAlarmType</i>	: Type of alarm to be gotten
	LPVOID	<i>lpAlarmMsg</i>	: Pointer for area in which message is to be stored struct STRINGTYPE { long <i>lBuffSize</i> ; // Buffer size of <i>lpzBuff</i> LPSTR <i>lpzBuff</i> ; // Pointer for buffer in which // message is to be stored };
	long	<i>lReadType</i>	: Data type of <i>lpAlarmMsg</i> (T_STR)

Output	DWORD	<i>dwStatus</i>	: Error code
	0		: Normal completion
	ME_OPE_CURRALM_ADDR		: Address illegal
	ME_OPE_CURRALM_ALMTYPE		: Alarm type illegal
	ME_OPE_CURRALM_DATAERR		: There is an error in communication data between NC and PC
	ME_OPE_CURRALM_DATASIZE		: Data are too much to be stored in buffer prepared by APP
	ME_OPE_CURRALM_DATATYPE		: Data type illegal
	ME_OPE_CURRALM_NOS		: Number of getting messages illegal
	* <i>lpzBuff</i> : Alarm message		

Details of the function Gets the alarm message of the currently occurring alarm in the NC Card designated in *lAddress*. Note that the alarm message is in the language designated in parameter (#1043 lang) of NC.
Designate the number of messages to be gotten within numbers 1 to 10.
Message is gotten in the order of highest priority to lowest priority.

The following is a list of alarm types.

M_ALM_NC_ALARM	: NC alarm
M_ALM_STOP_CODE	: Stop code
M_ALM_PLC_ALARM	: PLC alarm message
M_ALM_OPE_MSG	: Operator message
M_ALM_ALL_ALARM	: No alarm type classification

For the messages in *lpzBuff*, codes CR and LF are inserted between messages.
NULL is inserted at the end of the buffer.

melGetCurrentPrgBlock

melGetCurrentPrgBlock

This command gets the currently running program.

```
dwStatus = melGetCurrentPrgBlock ( hWnd, lAddress, lPrgBlockNos, lpPrgBlock,  
                                   lReadType );
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address Designate the following items: <ul style="list-style-type: none">• NC Card designation• System designation
	long	<i>lPrgBlockNos</i>	: Number of blocks to be gotten
	LPVOID	<i>lpPrgBlock</i>	: Pointer for the area in which the gotten program is to be stored struct GETPRGBLOCK { int <i>iCurrentBlockNum</i> ; // Currently running block // (block in the gotten data) // 0: Not in operation // 1: First block // 2: Second block long <i>lPrgDataSize</i> ; // Buffer size of <i>lpPrgData</i> LPSTR <i>lpPrgData</i> ; // Pointer for the buffer in which the program is to be stored };
	long	<i>lReadType</i>	: Data type of <i>lpPrgBlock</i> (T_GETPRGBLOCK)
Output	DWORD	<i>dwStatus</i>	: Error code
		0	: Normal completion
		ME_OPE_GETPRGBLK_ADDR	: Address illegal
		ME_OPE_GETPRGBLK_DATAERR	: There is an error in communication data between NC and PC
		ME_OPE_GETPRGBLK_DATASIZE	: Data are too much to be stored in buffer secured by APP
		ME_OPE_GETPRGBLK_DATATYPE	: Data type illegal
	ME_OPE_GETPRGBLK_NOS	: Number of blocks designation illegal	
	* <i>lpPrgData</i> : Program data		

Details of the function Reads out the currently running program block of the system designated in *lAddress*. Designate the number of blocks to be gotten within numbers 1 to 10. Codes CR and LF are inserted between the blocks of program stored in the buffer indicated by *lpPrgData*. NULL is inserted at the end of the buffer. The following are set when operation search is not carried out:
iCurrentBlockNum = 0
Buffer indicated by *lpPrgData* = NULL

melSelectExecPrg

melSelectExecPrg

This command carries out operation search.

```
dwStatus = melSelectExecPrg ( hWnd, lAddress, lpSelectPrg, lDataType,  
                             lSequenceNum, lBlockNum ) ;
```

Input	HWND	<i>hWnd</i>	: Window handle for local window
	long	<i>lAddress</i>	: Address
			Designate the following items:
			• NC Card designation
			• System designation
	LPVOID	<i>lpSelectPrg</i>	: STRINGTYPE structure address Store the head address of the file name character string that operation searches in the STRINGTYPE structure.
	long	<i>lDataType</i>	: Data type of <i>lSelectPrg</i> (specify T_STR)
	long	<i>lSequenceNum</i>	: Number of sequence to be searched
	long	<i>lBlockNum</i>	: Number of block to be searched
Output	DWORD	dwStatus	: Error code
		0	: Normal completion
		ME_OPE_SELECTPRG_ADDR	: Address illegal
		ME_OPE_SELECTPRG_DATATYPE	: Data type illegal
		ME_OPE_SELECTPRG_FILESYSTEM	: There is something wrong in file system
		ME_OPE_SELECTPRG_NOPRG	: Program file not found
		ME_OPE_SELECTPRG_PRGFORMAT	: Format for program file name illegal
		ME_OPE_SELECTPRG_RUNNING	: Program running
		ME_OPE_SELECTPRG_LONGPATH	: Path name is too long
		ME_OPE_SELECTPRG_NCPCCOM	: Could not communicate with the PC side file server
	ME_OPE_SELECTPRG_TIMEOUT	: Time out	

Details of the function

Select the machining program to be executed by the NC Card designated in *lAddress*.

Designate in *lpSelectPrg*, the name of program file to be searched.

Designate the address of the STRINGTYPE structure in *lpSelectPrg*. Designate the head address of the file name character string in the *lpzBuff* of the STRINGTYPE structure.

Only designate the file name when designating a program in the NC Card.

Example) Designate "2000.PRG" when designating the machining program of program No. 2000 in the NC Card.

When selecting the machining program in the NC Card, only the file in the \PRG\USER\ directory can be designated.

When designating a medium program other than the NC Card, the program name is designated by the absolute path that includes the drive name, path name, and program file name. (The personal computer direct operation function is necessary when designating a medium program other than the NC Card.)

Example)

Drive name	"C:"
Path name	"\PARTPRG\MAIN\"
Program file name	"1000.PRG"

In this case, designate "C:\PARTPRG\MAIN\1000.PRG"

Using *lSequenceNum* and *lBlockNum*, an operation search of the operation that designated the sequence No. and block No. is possible.

When *lSequenceNum* and *lBlockNum* are both 0, the program head is operation searched..

3. Appendix

3.1 Command Name Table (alphabetical order)

Command name	Function	Command classification	Page
melActivatePLC	Start/stop PLC program	Operation	31
melCancelModal	Cancel registration of high-speed read-out data	Data Access	26
melCloseDirectory	Close directory	File Access	14
melCopyFile	Copy file	File Access	15
melDeleteFile	Delete file	File Access	17
melGetCurrentAlarmMsg	Get occurring alarm message	Operation	32
melGetCurrentPrgBlock	Get running program	Operation	33
melGetDiskFree	Get size of free space of drive/directory	File Access	18
melGetDriveList	Get drive list	File Access	19
melIoctl	Execution of function	System Control	12
melOpenDirectory	Open directory	File Access	20
melReadData	Read-out data	Data Access	27
melReadDirectory	Read file information	File Access	22
melReadModal	High-speed read-out of data	Data Access	28
melRegisterModal	Register high-speed read-out data	Data Access	29
melRenameFile	Rename file	File Access	23
melSelectExecPrg	Operation search	Operation	34
melVerifyFile	File verification	File Access	24
melWriteData	Write data	Data Access	30

3.2 Error Codes

The following is a list of error codes.

Error codes

Command related errors

ME_CMD_DIFFER	0x80010004	Transmit command and reply command do not match
ME_CMD_ERRNUMGETFAIL	0x80010005	Failed to retrieve an error No.
ME_CMD_NOCOMMAND	0x80010002	No command
ME_CMD_NOOPTION	0x80010001	No option
ME_CMD_SYNTAX	0x80010003	Syntax error

System related errors

ME_SYSFUNC_IOCTL_ADDR	0x80020190	Address illegal
ME_SYSFUNC_IOCTL_DATA	0x80020133	Data range illegal
ME_SYSFUNC_IOCTL_FUNCTION	0x80020132	Command illegal
ME_SYSFUNC_IOCTL_NOTOPEN	0x80020102	Device not opened

File access related errors

ME_FILE_COPY_BUSY	0x80030447	Now copying this file not permitted (program running)
ME_FILE_COPY_ENTRYOVER	0x80030403	Number of entry over
ME_FILE_COPY_FILEEXIST	0x80030401	Copy destination file already exists
ME_FILE_COPY_FILESYSTEM	0x80030443	There is something wrong in file system
ME_FILE_COPY_ILLEGALNAME	0x80030498	Format for file name illegal
ME_FILE_COPY_MEMORYOVER	0x80030404	Memory capacity over
ME_FILE_COPY_NAMELENGTH	0x80030448	File name is too long
ME_FILE_COPY_NODIR	0x80030491	Directory does not exist
ME_FILE_COPY_NODRIVE	0x8003049b	Drive does not exist
ME_FILE_COPY_NOFILE	0x80030442	File does not exist
ME_FILE_COPY_PLCRUN	0x80030446	Now copying this file not permitted (PLC running)
ME_FILE_COPY_READ	0x80030494	Copy source file cannot be read out
ME_FILE_COPY_WRITE	0x80030495	Copy destination file cannot be written in
ME_PCFILE_COPY_CREATE	0x800304a4	File cannot be created (PC)
ME_PCFILE_COPY_FILEEXIST	0x80030402	Copy destination file already exists (PC)
ME_PCFILE_COPY_ILLEGALNAME	0x800304a7	Format for file name illegal (PC)
ME_PCFILE_COPY_NODIR	0x800304a2	Directory does not exist (PC)
ME_PCFILE_COPY_NODRIVE	0x800304a8	Drive does not exist (PC)
ME_PCFILE_COPY_NOFILE	0x800304a1	File does not exist (PC)
ME_PCFILE_COPY_OPEN	0x800304a3	File cannot be opened (PC)
ME_PCFILE_COPY_READ	0x800304a5	Copy source file cannot be read out (PC)
ME_PCFILE_COPY_WRITE	0x800304a6	Copy destination file cannot be written in (PC)
ME_FILE_DEL_BUSY	0x80030247	Now deleting this file not permitted (program running)
ME_FILE_DEL_FILESYSTEM	0x80030243	There is something wrong in file system
ME_FILE_DEL_ILLEGALNAME	0x80030298	Format for file name illegal
ME_FILE_DEL_NAMELENGTH	0x80030248	File name is too long
ME_FILE_DEL_NODIR	0x80030291	Directory does not exist
ME_FILE_DEL_NODRIVE	0x8003029b	Drive does not exist
ME_FILE_DEL_NOFILE	0x80030242	File does not exist
ME_FILE_DEL_NOTDELETE	0x80030201	Deleting this file is not permitted
ME_PCFILE_DEL_ILLEGALNAME	0x800302a7	Format for file name illegal (PC)
ME_PCFILE_DEL_NODIR	0x800302a2	Directory does not exist (PC)
ME_PCFILE_DEL_NODRIVE	0x800302a8	Drive does not exist (PC)
ME_PCFILE_DEL_NOFILE	0x800302a1	File does not exist (PC)
ME_PCFILE_DEL_NOTDELETE	0x80030202	Deleting this file is not permitted (PC)

Error Codes

ME_FILE_DIR_ALREADYOPENED	0x80030101	Another directory is already opened
ME_FILE_DIR_DATASIZE	0x80030103	Data are too much to be stored in buffer secured the application
ME_FILE_DIR_FILESYSTEM	0x80030143	There is something wrong in file system
ME_FILE_DIR_NAMELENGTH	0x80030148	File name is too long
ME_FILE_DIR_NODIR	0x80030191	Directory does not exist
ME_FILE_DIR_NODRIVE	0x8003019b	Drive does not exist
ME_FILE_DIR_NOTOPEN	0x80030190	Directory not opened
ME_FILE_DIR_READ	0x80030194	File information read error
ME_PCFILE_DIR_NODIR	0x800301a2	Directory does not exist (PC)
ME_PCFILE_DIR_NODRIVE	0x800301a8	Drive does not exist (PC)
ME_PCFILE_DIR_NOFILE	0x80030142	File does not exist
ME_PCFILE_DIR_NOTOPEN	0x800301a0	Directory not opened (PC)
ME_PCFILE_DIR_READ	0x800301a5	File information read error (PC)
ME_FILE_DISKFREE_FILESYSTEM	0x80030643	There is something wrong in file system
ME_FILE_DISKFREE_NAMELENGTH	0x80030648	File name is too long
ME_FILE_DISKFREE_NODIR	0x80030691	Directory does not exist
ME_FILE_DISKFREE_NODRIVE	0x8003069b	Drive does not exist
ME_PCFILE_DISKFREE_NODIR	0x800306a2	Directory does not exist (PC)
ME_PCFILE_DISKFREE_NODRIVE	0x800306a8	Drive does not exist (PC)
ME_FILE_DRVLIST_DATASIZE	0x80030701	Data are too much to be stored in buffer secured the application
ME_FILE_DRVLIST_READ	0x80030794	Drive list read error
ME_FILE_REN_BUSY	0x80030347	Now renaming not permitted (program running)
ME_FILE_REN_FILEEXIST	0x80030301	New file name already exists
ME_FILE_REN_FILESYSTEM	0x80030343	There is something wrong in file system
ME_FILE_REN_ILLEGALNAME	0x80030398	Format for file name illegal
ME_FILE_REN_NAMELENGTH	0x80030348	File name is too long
ME_FILE_REN_NODIR	0x80030391	Directory does not exist
ME_FILE_REN_NODRIVE	0x8003039b	Drive does not exist
ME_FILE_REN_NOFILE	0x80030342	File does not exist
ME_FILE_REN_NOTRENAME	0x80030303	Renaming this file is not permitted
ME_FILE_REN_SAMENAME	0x80030305	New name and old one of file are the same
ME_PCFILE_REN_FILEEXIST	0x80030302	New file name already exists (PC)
ME_PCFILE_REN_ILLEGALNAME	0x800303a7	Format for file name illegal (PC)
ME_PCFILE_REN_NODIR	0x800303a2	Directory does not exist (PC)
ME_PCFILE_REN_NODRIVE	0x800303a8	Drive does not exist (PC)
ME_PCFILE_REN_NOFILE	0x800303a1	File does not exist (PC)
ME_PCFILE_REN_NOTRENAME	0x80030304	Renaming this file is not permitted (PC)
ME_PCFILE_REN_SAMENAME	0x80030306	New name and old one of file are the same (PC)
ME_FILE_VERIFY_DIFFER	0x80030805	Verification error
ME_FILE_VERIFY_FILESYSTEM	0x80030843	There is something wrong in the file system
ME_FILE_VERIFY_ILLEGALNAME	0x80030898	Format for file name illegal
ME_FILE_VERIFY_NAMELENGTH	0x80030848	File name is too long
ME_FILE_VERIFY_NODIR	0x80030891	Directory does not exist
ME_FILE_VERIFY_NODRIVE	0x8003089b	Drive does not exist
ME_FILE_VERIFY_NOFILE	0x80030842	File does not exist
ME_FILE_VERIFY_NOTSUPPORTED	0x80030849	Verification function is not supported
ME_FILE_VERIFY_READ	0x80030894	Verification file cannot be read
ME_FILE_VERIFY_NODIR	0x800308a2	Directory does not exist (PC)
ME_FILE_VERIFY_NODRIVE	0x800308a8	Drive does not exist (PC)
ME_FILE_VERIFY_NOFILE	0x800308a1	File does not exist (PC)
ME_FILE_VERIFY_OPEN	0x800308a3	File cannot be opened (PC)

Error Codes

Data access related errors

ME_DATA_MDLCANCEL_ADDR	0x80040590	Address illegal
ME_DATA_MDLCANCEL_NOTREGIST	0x80040501	Modal ID not registered
ME_DATA_MDLREAD_ADDR	0x80040690	Address illegal
ME_DATA_MDLREAD_DATASIZE	0x80040696	Data are too much to be stored in buffer secured the application
ME_DATA_MDLREAD_DATATYPE	0x80040697	Data type illegal
ME_DATA_MDLREAD_NOTREGIST	0x80040601	Modal ID not registered
ME_DATA_MDLREAD_READ	0x8004069d	Now reading data not permitted
ME_DATA_MDLREAD_WRITEONLY	0x8004069f	Write only data
ME_DATA_MDLREGIST_ADDR	0x80040490	Address illegal
ME_DATA_MDLREGIST_PRIORITY	0x80040402	Priority designation illegal
ME_DATA_MDLREGIST_REGIST	0x80040401	Number of data registration over
ME_DATA_MDLREGIST_SECT	0x80040491	Section No. illegal
ME_DATA_MDLREGIST_SUBSECT	0x80040492	Sub-section No. illegal
ME_DATA_READ_ADDR	0x80040190	Address illegal
ME_DATA_READ_DATASIZE	0x80040196	Data are too much to be stored in buffer secured the application
ME_DATA_READ_DATATYPE	0x80040197	Data type illegal
ME_DATA_READ_READ	0x8004019d	Now reading data not permitted
ME_DATA_READ_SECT	0x80040191	Section No. illegal
ME_DATA_READ_SUBSECT	0x80040192	Sub-section No. illegal
ME_DATA_READ_WRITEONLY	0x8004019f	Write only data
ME_DATA_WRITE_ADDR	0x80040290	Address illegal
ME_DATA_WRITE_DATASIZE	0x80040296	Data are too much to be stored in buffer secured the application
ME_DATA_WRITE_DATATYPE	0x80040297	Data type illegal
ME_DATA_WRITE_READONLY	0x8004029b	Read only data
ME_DATA_WRITE_SECT	0x80040291	Section No. illegal
ME_DATA_WRITE_SUBSECT	0x80040292	Sub-section No. illegal
ME_DATA_WRITE_WRITE	0x8004029e	Now writing data not permitted

Operation related errors

ME_OPE_ACTPLC_ADDR	0x80050990	Address illegal
ME_OPE_ACTPLC_MODE	0x80050991	Mode illegal
ME_OPE_CURRALM_ADDR	0x80050d90	Address illegal
ME_OPE_CURRALM_ALMTYPE	0x80050d02	Alarm type illegal
ME_OPE_CURRALM_DATAERR	0x80050d03	There is an error in communication data between NC and PC
ME_OPE_CURRALM_DATASIZE	0x80050d93	Data are too much to be stored in buffer prepared the application
ME_OPE_CURRALM_DATATYPE	0x80050d94	Data type illegal
ME_OPE_CURRALM_NOS	0x80050d01	Number of getting messages illegal
ME_OPE_GETPRGBLK_ADDR	0x80050c90	Address illegal
ME_OPE_GETPRGBLK_DATAERR	0x80050c03	There is an error in communication data between NC and PC
ME_OPE_GETPRGBLK_DATASIZE	0x80050c93	Data are too much to be stored in buffer secured the application
ME_OPE_GETPRGBLK_DATATYPE	0x80050c94	Data type illegal
ME_OPE_GETPRGBLK_NOS	0x80050c01	Number of blocks designation illegal
ME_OPE_GETPRGBLK_NOSEARCH	0x80050c02	Operation search not completed

Error Codes

ME_OPE_SELECTPRG_ADDR	0x80051090	Address illegal
ME_OPE_SELECTPRG_DATATYPE	0x80051094	Data type illegal
ME_OPE_SELECTPRG_FILEREAD	0x80051094	File read error
ME_OPE_SELECTPRG_FILESYSTEM	0x80051043	There is something wrong in file system
ME_OPE_SELECTPRG_FILEWRITE	0x80051095	File write error
ME_OPE_SELECTPRG_LONGPATH	0x80051005	Path name is too long
ME_OPE_SELECTPRG_NCPCCOM	0x80051006	Cannot communicate with the PC side file server
ME_OPE_SELECTPRG_NOPRG	0x80051002	Program file not found
ME_OPE_SELECTPRG_PRGFORMAT	0x80051001	Format for program file name illegal
ME_OPE_SELECTPRG_RESET	0x80051004	Cannot operation search in this state (IN reset)
ME_OPE_SELECTPRG_RUNNING	0x80051003	Cannot operation search in this state (IN operation)
ME_OPE_SELECTPRG_TIMEOUT	0x80051007	Time out

NC Card system related errors

ME_SYS_MALLOC	0x81008001	Work area cannot be secured
ME_SYS_SYSTEMDOWN	0x81000102	NC Card is now system down

PC system related errors

ME_PCSYS_EXECCOMTASK	0x81010101	Communication Task cannot be started
ME_PCSYS_MALLOC	0x81018001	Work area cannot be secured
ME_PCSYS_MLOCK	0x81018002	Work area illegal
ME_PCSYS_SEMAPHRE	0x81018003	Exclusive control not possible
ME_PCSYS_SEMAPHRETIMEOUT	0x81010114	Exclusive control time out

2-ports memory communication related errors

ME_SMEM_ALREADYOPEN	0x82000001	Already opened
ME_SMEM_BADCHANNEL	0x82000006	Channel No. illegal
ME_SMEM_BADFD	0x82000007	File descriptor illegal
ME_SMEM_CANCELED	0x82000016	Quitted because of cancel required
ME_SMEM_CANNOTOPEN	0x82000008	Cannot open
ME_SMEM_CARDNOTEXIST	0x82000004	Card not exist
ME_SMEM_ILLEGALSIZE	0x82000017	Packet size illegal
ME_SMEM_NOTOPEN	0x82000002	Not opened
ME_SMEM_TASKQUIT	0x82000018	Quitted because of task quit
ME_SMEM_TIMEOUT	0x82000014	Time out
ME_SMEM_UNKNOWNFUNC	0x82000032	Command illegal

Others

List of sub-error Nos.

File access related errors

ME_SUB_FILE_COPY_NONE	0	No sub-error: Download, etc., while operating
ME_SUB_FILE_COPY_BUSY	1	File input/output attempted while data inputting/outputting
ME_SUB_FILE_COPY_FORMAT	10	Error in the format
ME_SUB_FILE_COPY_PARAMETERNO	11	Error in the parameter No.
ME_SUB_FILE_COPY_TAG	12	Error in the tag
ME_SUB_FILE_COPY_TOOLOFFSETNUM	13	Error in the designation of the tool compensation No.
ME_SUB_FILE_COPY_TOOLOFFSETDATA	14	Error in the command of the tool compensation data
ME_SUB_FILE_COPY_OFFSETTYPE	15	Error in the designation of the offset type
ME_SUB_FILE_COPY_DATARANGE	20	Error in the data range
ME_SUB_FILE_COPY_IDDIFFER	50	File with incorrect data ID was input

Revision History

Sub-No.	Date of revision	Revision details
*	July, 1997	First edition created.

© 1995-1997 MITSUBISHI ELECTRIC CORPORATION
ALL RIGHTS RESERVED



MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: MITSUBISHI DENKI BLD. MARUNOUCHI, TOKYO 100 TEL.03-218-3426
