Mitsubishi Motion Controller

Sales and Service

No. 13-05E

Precautions Related to Reading and Writing Project Data Between Different Languages of Motion Controller Engineering Software

Thank you for your continued patronage of Mitsubishi Motion controller and Mitsubishi FA products.
The following describes the precautions to note when reading and writing project data between different languages using Motion controller engineering software.
We ask for your understanding in this matter.

1. Target Models

Motion controller engineering software
- MELSOFT MT Works2: SW1DNC-MTW2-☐
- MT Developer: SW6RNC-GSV☐

2. Phenomenon

When a project that was edited using Japanese-version MELSOFT MT Works2 (hereinafter omitted "MELSOFT") is written to a Motion CPU and then subject to an operation such as reading using English-version MT Works2, the characters of the Motion SFC program name, comments, and the like may become garbled or the characters of control statements of the Motion SFC program may partially disappear.
If this happens, a conversion error occurs with program data that impacts motion control, causing the data not to be transferrable to the Motion CPU. (Transferred data exists in sections that do not impact control.)

3. Cause

First, to prevent characters from becoming garbled when reading or saving MT Works2 project data between Windows of different languages, the character code needs to be changed to Unicode.
As shown in the table below, while Unicode is substantially supported by MT Works2 , the phenomenon described above occurs with data written inside the Motion CPU (a CPU other than Q17nDS/Q170MS).
The reason for this is that Unicode cannot be supported since data compatibility with MT Developer (SW6RNC-GSV) is supported.
For details, see the following page.

		Unicode Support	Remarks
Engineering tool	MT Developer	Unicode not supported	
	MT Works2	Unicode supported	Partially (CPU name setting, labels/ structures) not supported
Motion controller	Q17n(N) / Q17nH / Q17nD / Q170M	Unicode not supported	
	Q17nDS / Q170MS	Unicode supported	

4. How to Avoid the Problem

When you want to read, edit, or write project data using a language version other than the source editing language, we recommend that you use or change the characters to one-byte alphanumeric characters (including one-byte symbols).

Date of issue	April 2013	Title	Precautions Related to Reading and Writing Project Data Between Different Languages of Motion Controller Engineering Software	Mitsubishi Electric Corp., Nagoya Works 5-1-14 Yada-minami, Higashi-ku, Nagoya 461-8670 Tel.: +81 (52) 721-2111 Main line

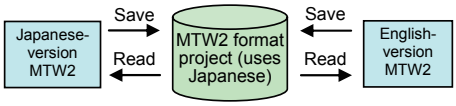
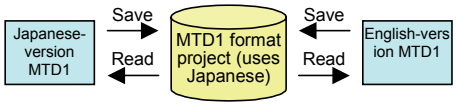
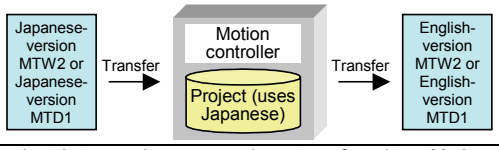
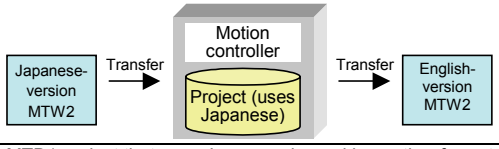
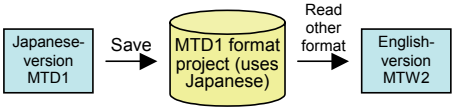
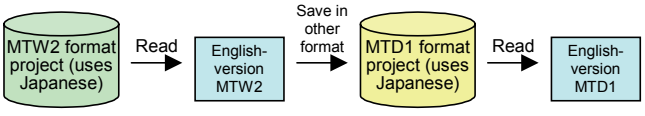
The following shows the operating conditions in the case of project data that uses Japanese (two-byte characters). All operations described below run normally with project data that uses one-byte alphanumeric characters (including one-byte symbols) only.

Note: In the explanation below, MT Works2 is abbreviated as MTW2 and MT Developer is abbreviated as MTD1.

Note that while the following describes an illustrative scenario with Japanese and English, the same holds true for other language combinations such as Chinese and Korean.

Table 1 Operation when Project Created Using Japanese Version Is Opened Using English Version

In the "Open Project" column, a circle indicates that the project can be opened, and a triangle indicates that the project cannot be opened.

No.	Operation	Operation in English Version		
1	When an MTW2 project that uses Japanese is saved or read using Japanese-/English-version MTW2 on a PC 	When a project that uses Japanese is read using English-version MTW2, the result is as follows (Note 1, Note 2):		
			Open Project	Garbled Characters
		The PC has Japanese fonts	○	No
		The PC does not have Japanese fonts	○	Yes (Note 4)
2	When an MTD1 project that uses Japanese is saved or read using Japanese-/English-version MTD1 on a PC 	The characters may become garbled or the project may not be openable due to the two-byte characters used. Example: When the characters "表", "予", "申", "能", "十", "ソ", and the like are used in the project name		
			Open Project	Garbled Characters
		The PC has Japanese fonts	△	Yes
		The PC does not have Japanese fonts	△	Yes
3-1	When a project that uses Japanese and was transferred to a Motion controller is transferred (read) using English-version MTW2 or English-version MTD1 [When the CPU is Q17nD or earlier] 	Same as No. 2		
			Open Project	Garbled Characters
		The PC has Japanese fonts	△	Yes
		The PC does not have Japanese fonts	△	Yes
3-2	When a project that uses Japanese and was transferred to a Motion controller is transferred (read) using English-version MTW2 [When the CPU is Q17nDS/Q170MS] 	Both MTW2 and the Motion controller support Unicode, and thus the project runs normally. (Note 2, Note 3)		
			Open Project	Garbled Characters
		The PC has Japanese fonts	○	No
		The PC does not have Japanese fonts	○	Yes (Note 4)
4	When an MTD1 project that uses Japanese is read in another format using English-version MTW2 on a PC (The project cannot be opened with Version 1.05F or earlier.) 	Same as No. 2		
			Open Project	Garbled Characters
		The PC has Japanese fonts	△	Yes
		The PC does not have Japanese fonts	△	Yes
5	When an MTW2 project that uses Japanese is read using English-version MTW2, saved in another format as a MTD1 project, and further read using English-version MTD1 on a PC 	Same as No. 2		
			Open Project	Garbled Characters
		The PC has Japanese fonts	△	Yes
		The PC does not have Japanese fonts	△	Yes

Note 1. When Japanese characters (including two-byte characters) are used in label and structure definitions and the project is read using English-version MTW2, the characters do not become garbled to the extent that the character code is destroyed, but program conversion is not executable.

Note 2. CPU name settings located in basic settings under system settings do not support Unicode and therefore become garbled when Japanese characters (including two-byte characters) are used. This does not, however, impact program execution processing.

Note 3. When label and structure definition data that uses Japanese characters (including two-byte characters) is transferred to a CPU (memory card of Unit No. 1) and the data is read, the characters become garbled and thus program conversion is not executable.
Note, however, that when only the program is transferred without transferring the label and structure definition data to the CPU and the label and structure definition data used is the data inside the project, conversion is possible.

Note 4. Since the PC does not have Japanese fonts, the characters are simply not correctly displayed (the character code is normal). There is no impact on program execution processing.