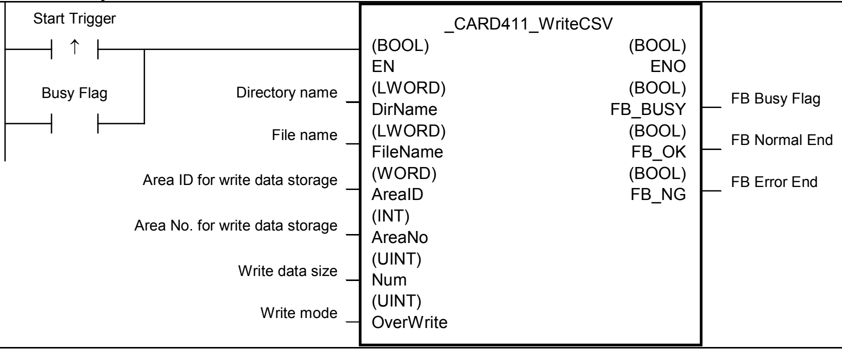
CARD 411	Write CSV File: _CARD411_WriteCSV

Basic function

Saves I/O data values in the Memory Card as a comma-separated variable file (.CSV extension) with new lines every 10 fields.

Symbol



File name	Lib\FBL\omronlib\PLC\Card_CARD411_WriteCSV10.cxf	
Applicable models	CPU Unit	Unit version 3.0 or higher
	CX-Programmer	Version 5.0 or higher

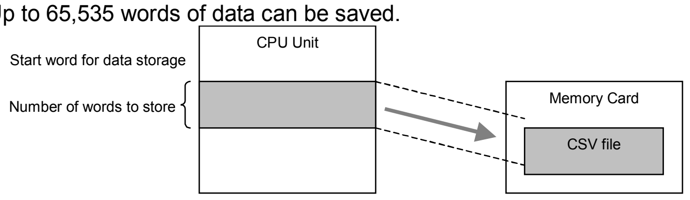
Conditions for usage

Shared Resources
- Memory Card

Memory Card Status
- The Memory Card must be recognized by the CPU Unit.
  The Memory Card Recognized Flag (A343.15) will be ON when CPU Unit has recognized the Memory Card.

Function description

When the Start Trigger turns ON, the function saves the specified I/O memory data as a CSV file (.IOM extension) in the Memory Card's root directory. The data file is a CSV text file with a new line every 10 fields.
Up to 65,535 words of data can be saved.



D0 … D10

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
000A	000B	000C	000D	000E	000F				

D11 … D15

For example, the file at the right will be created when 16 words of data are stored beginning at D0.

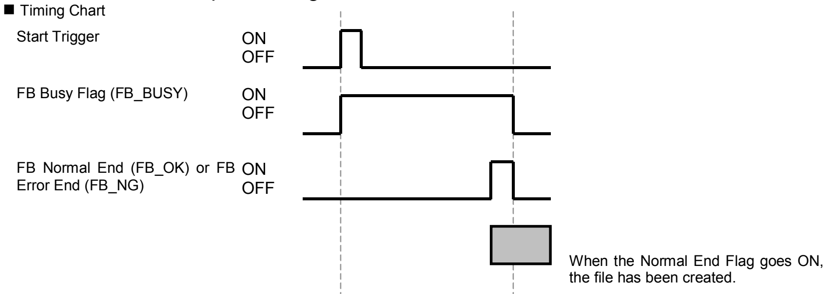
Comma-separated variable text file with new line every 10 fields

```
0000,0001,0002,0003,0004,0005,0006,0007,0008,0009
000A,000B,000C,000D,000E,000F
```

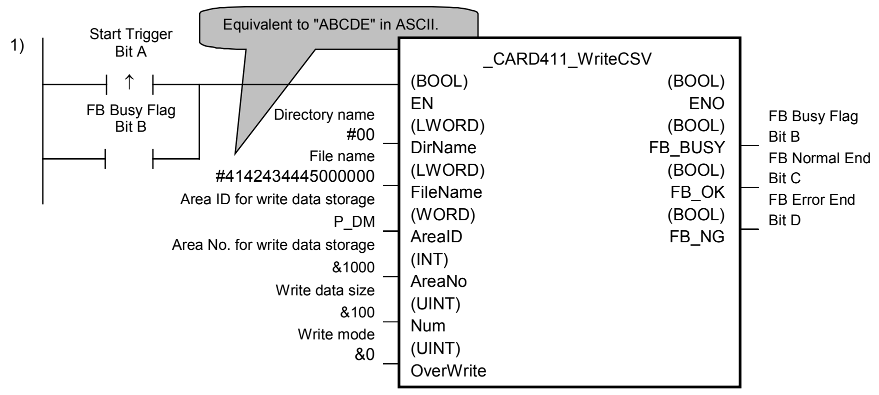
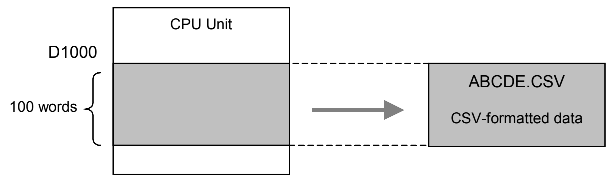
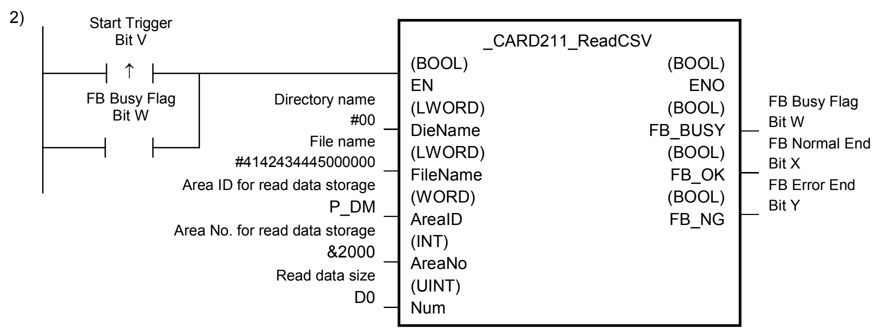
FB precautions

- If the Memory Card is already being accessed when the FB is started, the operation will be performed after the completion of the access.
- The FB is processed over multiple cycles. The FB_BUSY output variable can be used to check whether the FB is being processed.
- *FB_OK* or *FB_NG* will be turned ON for one cycle only after processing is completed. Use these flags to detect the end of FB processing.

■ Timing Chart



- This FB writes data to the Memory Card over a number of cycles. Consequently, the data will not be simultaneous. To preserve data simultaneity, transfer the desired data to a separate data area and use this FB to create a file from the data in that data area. Refer to the *Application example* below for a specific example.

EN input condition	Connect EN to an OR between an upwardly differentiated condition for the start trigger and the FB_BUSY output from the FB.
Restrictions **Input variables**	• Always use an upwardly differentiated condition for EN. • If the input variables are out of range, the ENO Flag will turn OFF and the FB will not be processed.
Output variables	• This FB requires multiple cycles to process. Always connect an OR including the FB_BUSY output variable to the EN input variable to ensure that the FB is processed to completion (see *Symbol*). • Do not turn the FB_BUSY output variable ON or OFF outside the FB.
Other	• If the Memory Card is missing or cannot be detected, the FB_NG Flag will be turned ON. • Never turn OFF the Power Supply when the CPU Unit's BUSY indicator (Accessing Memory Card indicator) is lit. Refer to the Related Manuals for other Memory Card precautions.
Application example	In this example, the 100 words from D1000 to D1099 are refreshed every cycle by a data link. The following program section saves the 100 words from D1000 to D1099 while preserving simultaneity. 1) When bit A goes ON, the 100 words from D1000 to D1099 are copied to W400 to W499. 2) When bit A goes ON, a file named *ABCDE.CSV* is created in the Memory Card's root directory. The file is a CSV file with a new line every 10 fields. The 100 words of data (read from D1000 to D1099 when bit A went ON) are saved to that file. The simultaneity of this data is preserved.  1) Start Trigger Bit A FB Busy Flag Bit B Equivalent to "ABCDE" in ASCII. Directory name #00 File name #4142434445000000 Area ID for write data storage P_DM Area No. for write data storage &1000 Write data size &100 Write mode &0 _CARD411_WriteCSV (BOOL) EN (LWORD) DirName (LWORD) FileName (WORD) AreaID (INT) AreaNo (UINT) Num (UINT) OverWrite (BOOL) ENO (BOOL) FB_BUSY (BOOL) FB_OK (BOOL) FB_NG FB Busy Flag Bit B FB Normal End Bit C FB Error End Bit D  CPU Unit D1000 100 words ABCDE.CSV CSV-formatted data  2) Start Trigger Bit V FB Busy Flag Bit W Directory name #00 File name #4142434445000000 Area ID for read data storage P_DM Area No. for read data storage &2000 Read data size D0 _CARD211_ReadCSV (BOOL) EN (LWORD) DieName (LWORD) FileName (WORD) AreaID (INT) AreaNo (UINT) Num (BOOL) ENO (BOOL) FB_BUSY (BOOL) FB_OK (BOOL) FB_NG FB Busy Flag Bit W FB Normal End Bit X FB Error End Bit Y

Related manuals	**Precautions when Using a Memory Card** There are several precautions that must be observed when using Memory Cards. This manual provides just an overview of the precautions. For details, refer to *5-1 File Memory* in the *CS/CJ Series Programmable Controllers Programming Manual* (W394-E1). 1) Format The Memory Card is already formatted when it is shipped, so it is not necessary to format a newly purchased Card. 2) Number of Files allowed in Root Directory There is a limit to the number of files that can be stored in the root directory of the Memory Card. The maximum number of files depends on the Memory Card model and format, but it ranges between 128 and 512 files. 3) Maximum Number of Overwrites A limit of 100,000 write operations has been set for warranty purposes. For example, if the Memory Card is written to every 10 minutes, over 100,000 write operations will be performed within 2 years. 4) Turning the Power OFF Never turn OFF the Power Supply when the CPU Unit's BUSY indicator (Accessing Memory Card indicator) is lit. **Reference for File Data Format** For details on file formats, refer to 5-1-3 Files in the *CS/CJ Series Programmable Controllers Programming Manual* (W394-E1).
Related FBs	Use the following functions when setting the present date or time as the directory name or file name. FB Get Date in ASCII (_CPU020_MakeAsciiDate) FB Get Time in ASCII (_CPU021_MakeAsciiTime)

Variable Tables

Input Variables

Name	Variable name	Data type	Default	Range	Description
EN	EN	BOOL			1 (ON): FB started 0 (OFF): FB not started.
Directory name	DirName	LWORD		At right.	Specifying the root directory: Set the directory to #00. Specifying a subdirectory: Specify the directory name (always 8 characters) in ASCII with the character codes at the beginning. If fewer than 8 characters are required, pad the extra characters with zeroes (#00). For example, to set the name ”ABCD,” input #4142434400000000. When indirectly specifying ASCII data in data area words, input the data as shown below. n+3 #3132 n+2 #3334 n+1 #3536 n #3738 In this case, the directory name is "12345678".
File name	FileName	LWORD		At right.	Specify the file name (always 8 characters) in ASCII with the character codes at the beginning. If fewer than 8 characters are required, pad the extra characters with zeroes (#00). For example, to set the name ”123.CSV,” input #3132330000000000. When indirectly specifying ASCII data in data area words, input the data as shown below. n+3 #3132 n+2 #3334 n+1 #3536 n #3738 In this case, the file name is "12345678.CSV".
Area ID for write data storage	AreaID	WORD	#0082	At right.	P_CIO (#00B0): CIO Area P_WR (#00B1): Work Area P_HR (#00B2): Holding Area P_DM (#0082): DM Area P_EM0 (#0050) to P_EMC (#005C): EM Area bank 0 to C
Area No. for write data	AreaNo	INT	&0		
Write data size	Num	UINT	&0		When adding data, the number of words data must be a multiple of 10 words.
Write mode	OverWrite	UINT	&0	&0 to &1	When creating a new file, specify *&0.* Specify the overwrite mode if the file already exists. &0: Add data &1: Overwrite When adding data (&0), the number of words to store must be a multiple of 10 words. (In other words, the last row must be a complete row of 10 fields.)

Output Variables

Name	Variable name	Data type	Range	Description
ENO (May be omitted.)	ENO	BOOL		1 (ON): FB processed normally. 0 (OFF): FB not processed or ended in an error.
FB Busy Flag	FB_BUSY	BOOL		Automatically turns OFF when processing is completed.
FB Normal end	FB_OK	BOOL		Turns ON for one cycle when processing ends normally.
FB Error end	FB_NG	BOOL		Turns ON for one cycle when processing ends in an error.

Reference

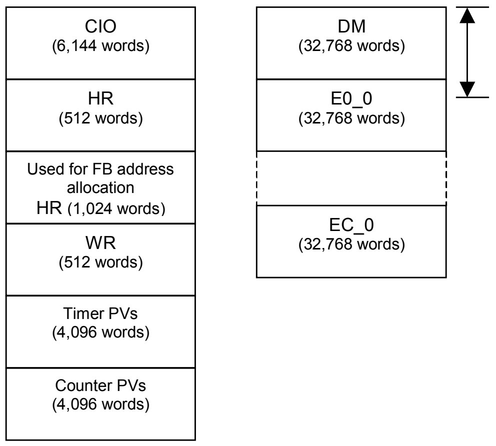
ASCII Table

Text	ASCII	Text	ASCII	Text	ASCII	Text	ASCII	Text	ASCII	Text	ASCII
0	#30	**8**	#38			**H**	#48	**P**	#50	**X**	#58
1	#31	**9**	#39	**A**	#41	**I**	#49	**Q**	#51	**Y**	#59
2	#32			**B**	#42	**J**	#4A	**R**	#52	**Z**	#5A
3	#33			**C**	#43	**K**	#4B	**S**	#53		
4	#34			**D**	#44	**L**	#4C	**T**	#54		
5	#35			**E**	#45	**M**	#4D	**U**	#55		
6	#36			**F**	#46	**N**	#4E	**V**	#56		
7	#37			**G**	#47	**O**	#4F	**W**	#57		

Examples:
Character 0: ASCII #30
Character A: ASCII #41
Character X: ASCII #58

Exceeding Data Area Boundaries

The following diagram shows the arrangement of the CPU Unit's I/O memory.
If the specified number of read words exceeds the specified data area's capacity, another data area will also be overwritten.



For example, if 40,000 words are specified with a start address of D00000, the function will save the data in words D00000 to D32767 and E0_0000 to E0_7231.

Version History

Version	Date	Contents
1.00	2005.2.	Original production

Note

This manual is a reference that explains the function block functions.
It does not explain the operational limitations of Units, components, or combinations of Units and components. Always read and understand the Operation Manuals for the system’s Units and other components before using them.