

Product: MotionWorks IEC Express & Pro v3.x	Doc#: PN.MWIEC.01 rev 3
Title: MotionWorks IEC Tag Export Specification	

## 1 Overview

MotionWorks IEC v3.x can write a CSV file listing user variables allocated when the controller is configured as a Modbus/TCP server. This comma separated value (CSV) file will be written to the directory location specified by the user in the Hardware Configuration. The file contains the variable name, data type, resource name, modbus address and comment.

## 2 File Format

The file contains comma separated value (CSV) data. The first row of the file is a header row of the column headings: “Variable name”, “Address”, “Data Type”, “Data Size (Bytes)”, and “Resource name”. The subsequent rows contain the data for each variable. The details of each column are described below.

The following is an example of the CSV data as viewed with Microsoft Excel.

Variable Name	Address	Data Type	Data Size (Bytes)	Comment	Resource
NewVar1	40001.3	BOOL	1	Simple var 1	MyResource
NewVar2	40002.B	BOOL	1	Simple var 2	MyResource
NewVar3	40003.5	BOOL	1	Simple var 3	MyResource
Counter	40100	INT	2	Number of parts made	MyResource
Health	40004.2	BOOL	1	Indicates the system is up	MyResource
V048	40005.8	BOOL	1		MyResource
V049	40006.F	BOOL	1		MyResource
Data	41000	XYDataStruct	8000	Pair of 500 X and Y positions	MyResource
XTest	40110	LREAL	8	For debug	MyResource
JogIncrement	40118	LREAL	8	Minimum position increment to jog	MyResource
Positions	41000	PosArr	36	Array of INTs used for indexing	MyResource
ErrorIDs	41050	ErrArr	36	Array of the last 18 ErrorIDs as UINT	MyResource
AxesStat	41100	StatArr	9	Axis Status Struct	MyResource
OutToPC	41200	Pcio	120	Struct of info sent to the PC	MyResource
InFromCNC	41300	PCio	120	Struct of info received from the PC	MyResource
SquareOffset	40126	LREAL	8	The amount to offset each new square in the X direction	MyResource

### 2.1 Variable Name

A variable with an address that falls into the Modbus server memory region. The maximum character limit for the variable name is 24 characters.

## 2.2 Address

The modbus address is calculated based on the IEC address of the variable of the project. If the datatype is BOOL, then the bit within the modbus address is identified by a decimal point a character with a range of 0 to F.

### 2.2.1 Mapping IEC Address to Modbus Registers

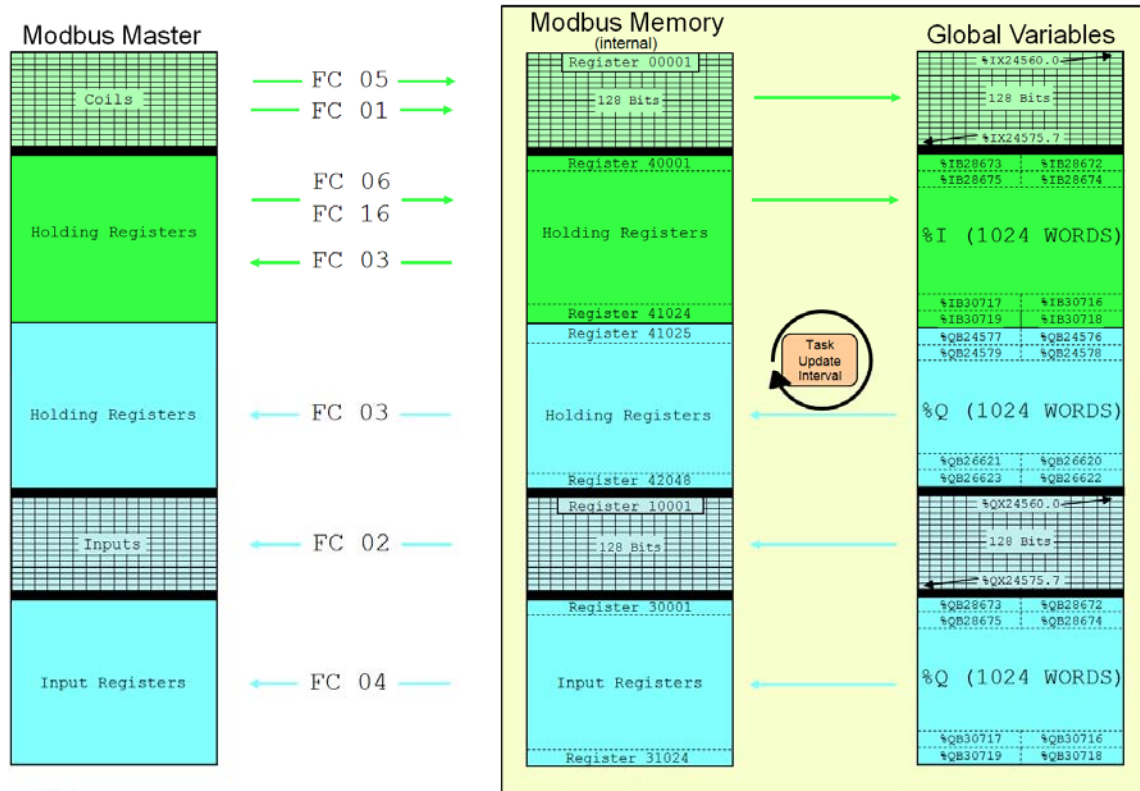
The following memory map describes how IEC addresses relate to Modbus registers for MP2000iec controllers:

[-] <b>Modbus FC#05 Qty: 128 Coils, Address Range: %IB24560 - %IB24575</b>
[-] <b>Modbus FC#02 Qty: 128 Inputs, Address Range: %QB24560 - %QB24575</b>
[-] <b>Modbus FC#04 Qty: 1024 Input Registers, Address Range: %QB28672 - %QB30719</b>
[-] <b>Modbus FC#06,16 Qty: 1024 Registers, Address Range: %IB28672 - %IB30719</b>
[-] <b>Modbus FC#03 Qty: 1024 Registers, Address Range: %QB24576 - %QB26623</b>

For example, if a Tag is declared at IEC address %IB28674, then it is located at Modbus Holding register 40003. (The exact register depends on whether the Master uses 40000 or 40001 as the starting address.)

The figure below illustrates how a Modbus Master reads and writes global variables on the Modbus Slave.

### MP2000iec as a Modbus Server / Slave



## 2.3 DataType

The table below list the standard data types, their range, their size and the character identifier used in the IEC Address.

Character (Represents Size)	Data type	Description	Size (bits)	Range
X	BOOL	Boolean	1	0...1
B	SINT	Short integer	8	-128...127
W	INT	Integer	16	-32,768...32,767
D	DINT	Double integer	32	-2,147,483,648 up to 2,147,483,647
W	USINT	Unsigned short integer	8	0 up to 255
W	UINT	Unsigned integer	16	0 up to 65,535
D	UDINT	Unsigned double integer	32	0 up to 4,294,967,295
D	REAL	Real numbers	32	-3.402823466 E+38 up to -1.175494351 E-38 and +1.175494351 E-38 up to +3.402823466 E+38
L	LREAL	Long real numbers	64	-1.7976931348623158 E+308 up to -2.2250738585072014 E-308 and +2.2250738585072014 E-308 up to +1.7976931348623158
D	TIME	Duration	32	0... 4,294,967,295 ms
B	BYTE	Bit string of length 8	8	0...255 (16#00...16#FF)
W	WORD	Bit string of length 16	16	0...65,535 (16#00...16#FFFF)
D	DWORD	Bit string of length 32	32	0...4,294,967,295 (16#00....16#FFFFFFFF)

In addition, the user may define custom data types. These may be difficult to represent in the HMI if the tag is comprised of a structure, although it is possible to determine the DataSize of the User Defined Datatype by using the “SizeOf()” function.

## 2.4 DataSize

This indicates the number of bytes required for the variable.

## 2.5 Comment

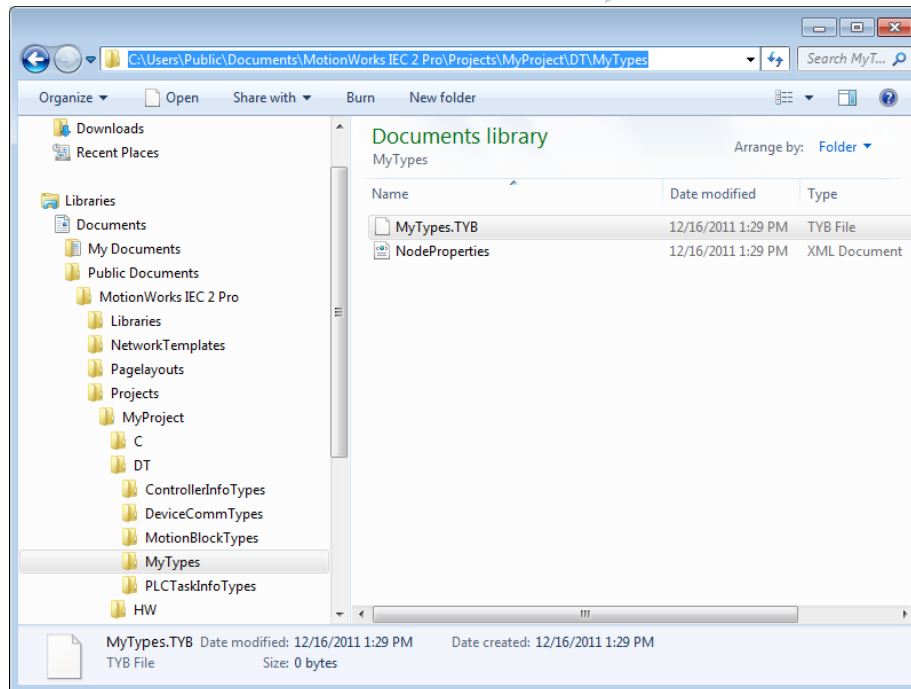
The text comment for the variable and has no practical character limit.

## 2.6 Resource

This will be the same as the name in the project tree, for example, “MP2300Siec”. This is applicable if the MotionWorks IEC project contains more than one MPiec controller. The resource is associated with the IP address of the controller.

## 2.7 User Defined Data types

Information describing User Defined Data Types can be found in the DT folder under the Project Name folder. The DataType information is available in simple text format. Refer to the \*.TYB file. All potentially referenced datatypes are listed here, even if the user has referred to datatypes from other user libraries.



Example DataType file:

```

237 0  Toolbox_DataTypes\Toolbo  SWError_Struct  1090  8  USER  STRUCT
238 0  TrackNumber  INT  3  0
239 0  FirstOnPosition  INT  3  0
240 0  LastOnPosition  INT  3  0
241 0  AxisDirection  INT  3  0
242 0  CamSwitchMode  INT  3  0
243 0  Duration  INT  3  0
244 0  ImproperOnPosition  INT  3  0
245 0  onOFFPositionError  INT  3  0
248 0  Toolbox_DataTypes\Toolbo  SWBoolArray  1091  1  USER  ARRAY  BOOL  1
248 0  0  255
250 0  Toolbox_DataTypes\Toolbo  SWLREALArray  1092  1  USER  ARRAY  LREAL  11
250 0  0  255
252 0  Toolbox_DataTypes\Toolbo  SWDINTArray  1093  1  USER  ARRAY  DINT  4
252 0  0  255
254 0  Toolbox_DataTypes\Toolbo  TrackBoolArray  1094  1  USER  ARRAY  BOOL  1
254 0  0  31
  
```