Wonderware Operations Integration – Supervisory Mitsubishi Electric MELSEC Server (G-1.2 Series)

Schneider Electric

Contents

Wonderware Operations Integration - Supervisory Mitsubishi Electric MELSEC Server (G-1.2 Series)

This document describes the technical specifications and configuration options for the Wonderware® Operations Integration - Supervisory Mitsubishi Electric MELSEC Server (or MELSEC OI Server, for short).

Last revision 2016-05-27

Legal Information

Schneider Electric Software, LLC
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

http://software.invensys.com/

For comments or suggestions about the product documentation, send an e-mail message to ProductDocumentationComments@schneider-electric.com.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Introduction to the MELSEC OI Server

These are the technical specifications for Wonderware Operations Integration - Supervisory Mitsubishi Electric MELSEC Server.

Requirements

MELSEC OI Server requires Wonderware Operations Integration – Core G-1.2 or later.

Supported hardware and software

This OI Server enables Ethernet and serial communication with Q, QnA, and L Series PLCs from Mitsubishi Electric, using the MELSEC protocol.

To program these PLCs, you need to use the GX Developer or GX Works2 programming software from Mitsubishi Electric.

Device configuration

Keep in mind that in order to communicate with a PLC over UDP/IP , you need to use the programming software to configure the Ethernet settings on the PLC itself. For a Q Series PLC in GX Works2, it would be something like this:

Protocol	Open System	Fixed Buffer	Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
UDP		Receive	Procedure Exist	Disable	No Confirm	1388	255.255.255.255	FFFF

In this configuration, you will be able to communicate over UDP using port 5000 (0x1388) .

> **Note:** It is necessary to configure the Destination IP Address when communicating over UDP/IP. You can configure it as 255.255.255.255, and it will communicate with any PLC that tries to communicate with it.

For an L Series PLC in GX Developer, you need a similar configuration:

Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.
UDP	MELSOFT Connection				
TCP	MELSOFT Connection				
UDP	MC Protocol		1386		
TCP	MC Protocol		1387		

With the configuration above, you will be able to communicate over UDP/IP using port 4998 (0x1386) and over TCP/IP using port 4999 (0x1387).

Conformance

The following hardware and software was used for conformance testing of this OI Server.

Configuration: Q Series PLC (Ethernet)

- Device: Q00JCPU with QJ71E71-100 module
- Cable: Ethernet
- UDP Port: 5001
- TCP Port: 5002

Configuration: Q Series PLC (Serial)

- Device: Q00JCPU with QJ71C24 module

- Cable: Null Modem Cable with Gender Adapter
- Serial Port: CH1 (RS232)
- Comm Settings: 9600, 8, 1, Odd

Configuration: L Series PLC (Ethernet)

- Device: L26CPU-BT with Ethernet port embedded on the CPU
- Cable: Ethernet
- UDP Port: 4998
- TCP Port: 4999

Configuring the MELSEC OI Server

Each server instance has its own hierarchy of objects, and each object has parameters that you need to configure in order to establish communication between the OI Server and individual devices on the network.

You can view a MELSEC server instance's configuration hierarchy under its **Configuration** node.

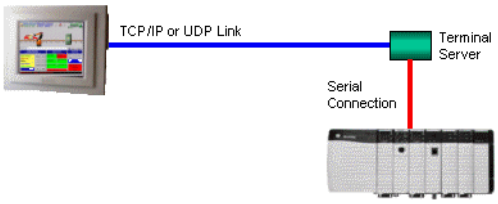
This section only describes how to configure object parameters for a MELSEC server instance. For more general information about adding and configuring objects, see "Configuring Your OI Server" in the *Operations Integration Server Manager Help*.

Direct Serial Communication and Serial Encapsulation

This OI Server supports direct serial communication with the target device, as well as serial encapsulation over a TCP/IP or UDP/IP network link.

Direct serial communication requires that the target device be directly connected to the computer that is running this OI Server. If you plan to use direct serial communication, you need to know the serial communication settings (e.g., baud rate, parity, etc.) that have been configured on the device, because you must configure this OI Server to match those settings.

Serial encapsulation enables serial communication with a target device that is connected to a terminal server on your TCP/IP or UDP/IP network. The terminal server is like a virtual serial port: it converts TCP or UDP messages to serial data and then relays them to the connected device. If you plan to use serial encapsulation, the target device should already be connected to the terminal server, and you need to know the IP address of the terminal server and the port number assigned to the device.



Serial encapsulation over a TCP/IP or UDP/IP link

All of the serial communication settings are included in the OI Server's channel parameters.

To configure the serial communication settings for a channel:

1. In the Operations Integration Server Manager, navigate to the ChannelSelector object that you want to configure:
   a) Expand the **Operations Integration Server Manager**, expand the node group, expand **Local** (or the remote computer name), and then expand **Operations Integration Supervisory Servers**.
   b) Locate and expand **Mitsubishi Electric - MELSEC**, and then expand its **Configuration** node.
   c) Select the ChannelSelector object that you want to configure.

   The channel parameters are displayed in the details pane on the right.
2. In the **Serial Encapsulation** box, select the encapsulation mode:

Option	Description
None	Direct serial communication with the device.
TCP/IP	Serial encapsulation over a TCP/IP Ethernet connection.
UDP/IP	Serial encapsulation over a UPD/IP Ethernet connection.

3. If you selected **None** above, configure the direct serial communication settings:
   a) In the **COM** box, select the COM port to which the target device is connected.

b) In the **Baud Rate**, **Data Bits**, **Stop Bits**, and **Parity** boxes, configure the serial communication settings to match the settings that have already been configured on the target device.

4. If you selected either **TCP/IP** or **UDP/IP** above, configure the serial encapsulation settings:
   a) In the **IP Address** box, type the IP address of the terminal server.
   b) In the **Port Number** box, type the port number on the terminal server that has been assigned to the target device.
   c) If you want to make the target device responsible for establishing communication with this OI Server, select **Server Mode**.

> **Note:** This option is not available for UDP/IP.

5. Proceed with configuring the rest of the channel parameters, as described in this documentation.

Configuring a Channel's Communication Settings

Configure the communication settings for a selected channel to ensure uninterrupted communication with the device network.

Communication Type

The specific transport protocol to be used. Select one of the following options from the list:

Option	Description
UDP	UDP/IP Ethernet communication.
TCP	TCP/IP Ethernet communication.
Serial	Direct serial communication.

Advanced

Click this button to open the *Advanced Settings* dialog box, which provides access to additional communication settings such as timeouts, retries, and buffer sizes. You might need to change these settings if the DAServer behaves unexpectedly during run time, but the default settings should work for most network configurations. For more information about these settings, see "Advanced Settings" in *Operations Integration Server Manager Help*.

Setting a Device's Station ID

Set the station ID for a selected device so that the OI Server can identify and communicate with it on the network.

Syntax

For UDP/IP or TCP/IP Ethernet communication, the station ID for a target device must use the following syntax:

```
<IP address>:<port number>
```

For direct serial communication, the station ID for a target device must use the following syntax:

```
<station number>
```

The following syntax diagram shows all of the possible options:
{ *IP address*:*port number* | *station number* }

IP address

The device's IP address on the Ethernet network.

port number

The port number for the MELSEC protocol.

station number

The station number of the PLC.

Examples

Examples of valid station IDs:

```
10.168.23.73:5001
```

```
0
```

MELSEC OI Server Reference

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the OI Server itself.

This section only describes the item reference syntax and options for the MELSEC server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the *Operations Integration Server Manager Help*.

Item Reference Syntax

Item references in this OI Server use the following syntax.

For Discrete (1-bit) memory registers — i.e., register types X, DX, Y, DY, B, SB, M, SM, L, S, V, F, TS, TC, SS, SC, CS, CC — use the following syntax:

```
<register type><address>
```

For Word (2-byte) memory registers — i.e., register types TN, SN, CN, D, SD, W, SW, Z, R, ZR — use the following syntax:

```
<register type><address>@[data type]
```

For bit-accessible memory registers and strings — i.e., register types DSH, WSH, RSH — use the following syntax:

```
<register type><address>.<bit number or string length>
```

The following syntax diagram shows all of the possible options:
{ *register type* | X | DX | Y | DY | B | SB | M | SM | L | S | V | F | TS | TC | TN | SS | SC | SN | CS | CC | CN | D | DSH | SD | W | WSH | SW | Z | R | RSH }*address* { | @ { *data type* | SHORT | WORD | BCD | LONG | DWORD | LBCD | FLOAT } | . { *bit number* | *string length* } }

Where…

register type

The type of memory register on the connected device, such as Input Relay (X), Output Relay (Y), File Register (R), and so on.

address

The address of the memory register on the connected device. This value can be either decimal or hexadecimal, depending on the register type.

data type

The formatted data type of the value read from the memory register. This is optional; if the data type is not specified, the value will be formatted as Word (i.e., unsigned,16-bit decimal) by default.

bit number or string length

The number of the bit to read/write. Bit 00 is the low bit. Bit 15 is the high bit.

For string registers like DSH, WSH and RSH, the length of the string is specified here.

Address Descriptions

The address descriptions consist of the register type, its item name and the allowable range of values, the default data type, allowable suffixes (if any), and allowable access methods.

Register Type	Item Reference		Length	Data Types	Access
	Item Name	Range			
Input	X	0000 to 3FFF (hex)	1 bit	-	Read/Write

Register Type	Item Reference		Length	Data Types	Access
	Item Name	Range			
Direct Input	DX	0000 to 3FFF (hex)	1 bit	-	Read/Write
Output	Y	0000 to 3FFF (hex)	1 bit	-	Read/Write
Direct Output	DY	0000 to 3FFF (hex)	1 bit	-	Read/Write
Link Relay	B	0000 to 3FFF (hex)	1 bit	-	Read/Write
Special Link Relay	SB	0000 to 07FF (hex)	1 bit	-	Read/Write
Internal Relay	M	0 to 16383	1 bit	-	Read/Write
Special Internal Relay	SM	0 to 2047	1 bit	-	Read/Write
Latch Relay	L	0 to 16383	1 bit	-	Read/Write
Step Relay	S	0 to 16383	1 bit	-	Read/Write
Edge Relay	V	0 to 2047	1 bit	-	Read/Write
Annunciator Relay	F	0 to 2047	1 bit	-	Read/Write
Timer Contacts	TS	0 to 2047	1 bit	-	Read/Write
Timer Coil	TC	0 to 2047	1 bit	-	Read/Write
Timer Value	TN	0 to 2047	2 bytes	Word, BCD	Read/Write
Integrating Timer Contact	SS	0 to 2047	1 bit	-	Read/Write
Integrating Timer Coil	SC	0 to 2047	1 bit	-	Read/Write
Integrating Timer Value	SN	0 to 2047	2 bytes	Word	Read/Write
Counter Contact	CS	0 to 1023	1 bit	-	Read/Write
Counter Coil	CC	0 to 1023	1 bit	-	Read/Write
Counter Value	CN	0 to 1023	2 bytes	Word, BCD	Read/Write
Data Register	D	0 to 12287	2 bytes	Word, BCD	Read/Write
		0 to 12286		DWord, LBCD, Float	Read/Write
Data Register, bit access	D	0.0 to 12287.15	2 bytes	-	Read/Write
Data Register, string access, HiLo byte ordering	DSH	0.2 to 12286.2	String	String	Read/Write
		0.128 to 12223.128			Read/Write
Special Data Register	SD	0 to 2047	2 bytes	Word, BCD	Read/Write
		0 to 2046		DWord, LBCD, Float	Read/Write
Link Register	W	0000 to 3FFF (hex)	2 bytes	Word, BCD	Read/Write
		0000 to 3FFE (hex)		DWord, LBCD, Float	Read/Write
Link Register, bit access	W	0000.00 to 3FFF.15 (hex)	2 bytes	-	Read/Write
Link Register, string access, HiLo byte ordering	WSH	0000.002 to 3FFE.002 (hex)	String	String	Read/Write
		0000.128 to 3FBF.128 (hex)			Read/Write
Special Link Register	SW	0000 to 07FF (hex)	2 bytes	Word, BCD	Read/Write
		0000 to 07FE (hex)		DWord, LBCD, Float	Read/Write
Index Register	Z	0000 to 15	2 bytes	Word, BCD	Read/Write

Register Type	Item Reference		Length	Data Types	Access
	Item Name	Range			
		0000 to 14		DWord, LBCD, Float	Read/Write
Index Register, bit access	Z	00.00 to 15.15	2 bytes	-	Read/Write
		00.00 to 14.31	4 bytes	-	Read/Write
File Register	R	0000 to 32767	2 bytes	Word, BCD	Read/Write
		0000 to 32766		DWord, LBCD, Float	Read/Write
File Register, bit access	R	0000.00 to 32767.15	2 bytes		Read/Write
File Register, strings access, HiLo byte ordering	RSH	0000.002 to 32766.002	String	String	Read/Write
		0000.128 to 32703.128			Read/Write

Supported Data Types

The data type is specified as a suffix in the item syntax. This OI Server supports the following data types.

Data Type / Suffix	Description	Range of Values
SHORT	Signed, 16-bit decimal value.	-32768 to 32767
WORD	Unsigned, 16-bit decimal value.	0 to 65535
BCD	16-bit binary coded decimal (BCD) with byte swap.	0 to 9999
LONG	Signed, 32-bit decimal value.	-2147483648 to 2147483647
DWORD	Unsigned, 32-bit decimal value.	0 to 4294967295
LBCD	32-bit binary coded decimal (BCD) with byte swap.	0 to 99999999
FLOAT	32-bit floating point value.	N/A

Examples of Item References

These are examples of valid item references for this OI Server. For more information about the referenced addresses, see the manufacturer's documentation for your device.

Register Type	Address on the Device	Item Reference
Data Register	D0000 - Decimal	D0
Link Register	W0007 - Hexadecimal	W7
File Register	R0010 - Decimal	R10
Timer Value	TN0017 - Decimal	TN17
Timer Contact	TS0000 - Decimal	TS0
Timer Coil	TC0007 - Decimal	TC7
Counter Current Value	CN1000 - Decimal	CN1000
Counter Contact	CS0007 - Decimal	CS7
Counter Coil	CC1400 - Decimal	CC1400
Input Relay	X0007 - Hexadecimal	X7
Output Relay	Y7772 - Hexadecimal	Y7772
Internal Relay	M0007 - Decimal	M7
Link Relay	B0000 - Hexadecimal	B0

Register Type	Address on the Device	Item Reference
Latch Relay	L0010 - Decimal	L10
Annunciator	F0007 - Decimal	F7
Special Link Relay	SB0000 - Hexadecimal	SB0
Edge Relay	V0009 - Decimal	V9
Step Relay	S0000 - Decimal	S0
Special Link Register	SW0007 - Hexadecimal	SW7
Retentive Timer Contact	SS0000 - Decimal	SS0
Retentive Timer Coil	SC0010 - Decimal	SC10
Retentive Timer Current Value	SN0001 - Decimal	SN1
Direct Input	DX0000 - Hexadecimal	DX0
Direct Output	DY0005 - Hexadecimal	DY5
Special Relay	SM0002 - Decimal	SM2
Special Register	SD0001 - Decimal	SD1
Index Register	Z0001 - Decimal	Z1
File Register	ZR0001 - Hexadecimal	ZR1

MELSEC OI Server Error Codes

The following tables describe the additional error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
1	Error Connect	IP address or port number is invalid.	Check the IP Address and port number.
2	PLC Error or Invalid Memory Address	You may have tried to Read a memory from the PLC (Device) that is not present on that CPU.	Check if the address (Device) configured does exist in the PLC.
3	Checksum Error	The check sum byte received is not equal to the expected byte.	Contact your technical support representative.
4	Protocol Error	The driver received an unexpected message from the device.	Contact your technical support representative.

Code	Description	Possible Causes	Solution
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).