

**Contents**

**CONTENTS** ..... 1

**INTRODUCTION** ..... 2

**GENERAL INFORMATION** ..... 4

    DEVICE SPECIFICATIONS ..... 4

    NETWORK SPECIFICATIONS ..... 4

    DRIVER CHARACTERISTICS ..... 4

    CONFORMANCE TESTING ..... 5

**SELECTING THE DRIVER** ..... 6

**CONFIGURING THE DRIVER** ..... 7

    CONFIGURING THE COMMUNICATION SETTINGS ..... 7

    CONFIGURING THE DRIVER WORKSHEETS ..... 8

**DATATYPE DESCRIPTION** ..... 13

    SAMPLE DATAITEM ..... 13

    EVENT DATAITEM ..... 14

    CONDITION DATAITEM ..... 16

**TIMESTAMP INFORMATION** ..... 18

**EXECUTING THE DRIVER** ..... 19

**TROUBLESHOOTING** ..... 20

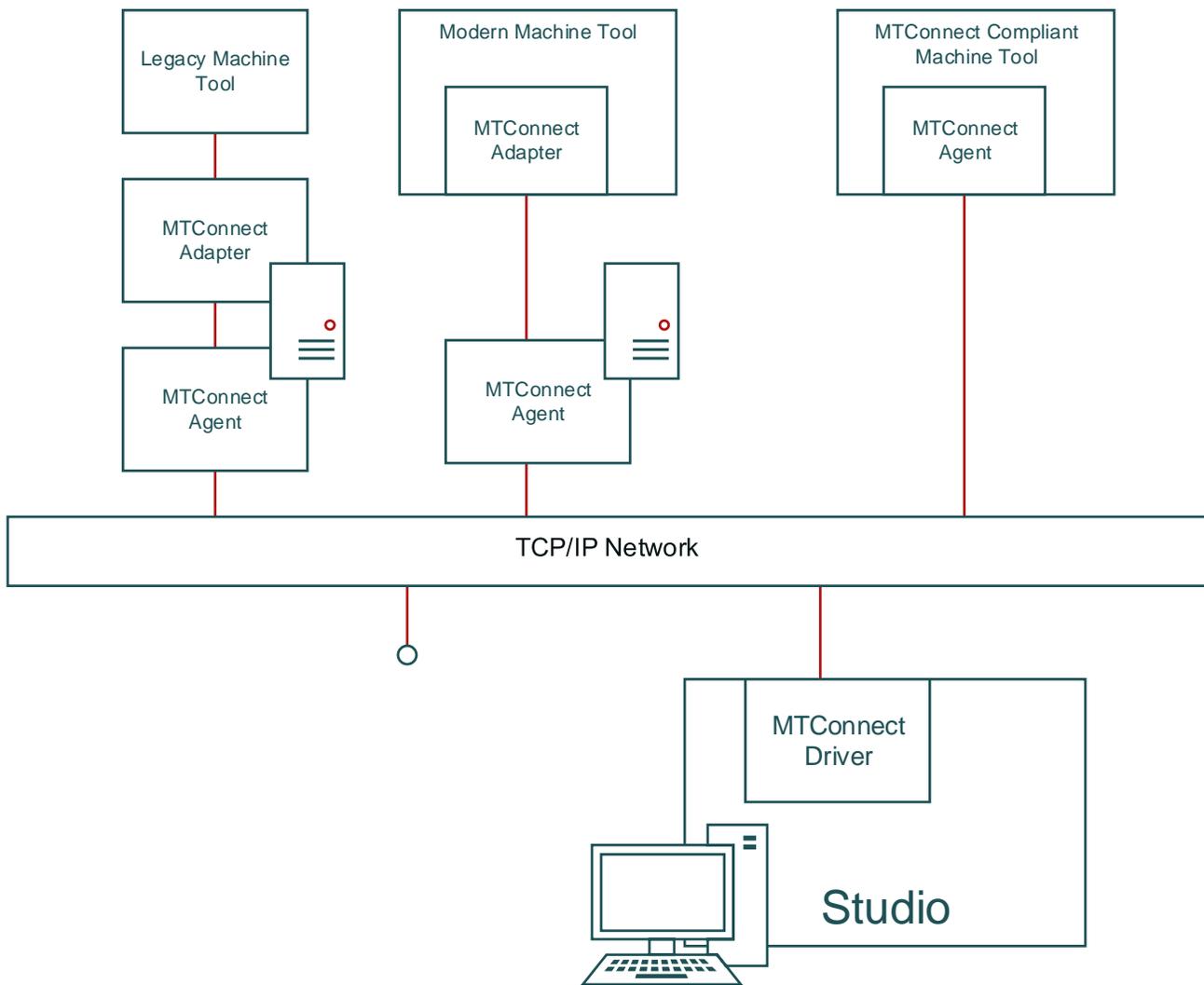
**SAMPLE APPLICATION** ..... 22

**REVISION HISTORY** ..... 23

## Introduction

The MTCOON driver enables communication between the Studio system and remote devices using the MTConnect protocol, according to the specifications discussed in this document.

MTConnect is a lightweight, extensible and open protocol that exchanges data between production equipment, sensor packages and many other hardware. Application using MTConnect data provide more efficient operations, improved production optimization and increased productivity. The basic network architecture of a MTConnect system is:



**MTConnect Architecture**

This document will help you to select, configure and execute the MTCOON driver, and it is organized as follows:

- **Introduction:** This section, which provides an overview of the document.
- **General Information:** Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver:** Explains how to select the MTCOON driver in the Studio system.
- **Configuring the Driver:** Explains how to configure the MTCOON driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver:** Explains how to execute the MTCOON driver during application runtime.
- **Troubleshooting:** Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application:** Explains how to use a sample application to test the MTCOON driver configuration
- **Revision History:** Provides a log of all changes made to the driver and this documentation.

 **Notes:**

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows NT/2000/XP/Vista environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

## General Information

This chapter identifies all of the hardware and software components required to implement communication between the MTCOON driver in Studio and remote devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

### Device Specifications

You can use this driver to communicate with MTConnect Agents. (The devices used for conformance testing are listed on the next page.)

### Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port:** Default port TCP 5000
- **Physical Protocol:** HTTP over TCP/IP
- **Logic Protocol:** MTConnect XML
- **Device Runtime Software:** None
- **Specific PC Board:** None
- **Adapters/Converters:** None
- **Cable Wiring Scheme:** None

### Driver Characteristics

The MTCOON driver package consists of the following files, which are automatically installed in the \DRV subdirectory of Studio:

- **MTCOON.INI:** Internal driver file. *You must not modify this file.*
- **MTCOON.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **MTCOON.PDF:** This document, which provides detailed information about the MTCOON driver.
- **MTCOON.DLL:** Compiled driver.

 **Note:**

You must use Adobe Acrobat® Reader™ to view the **MTCOON.PDF** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's Web site.

You can use the MTCOON driver on the following operating systems:

- Windows XP/Vista/7/8/10

For a description of the operating systems used to test driver conformance, see "Conformance Testing" below.

### Conformance Testing

The following hardware/software was used for conformance testing:

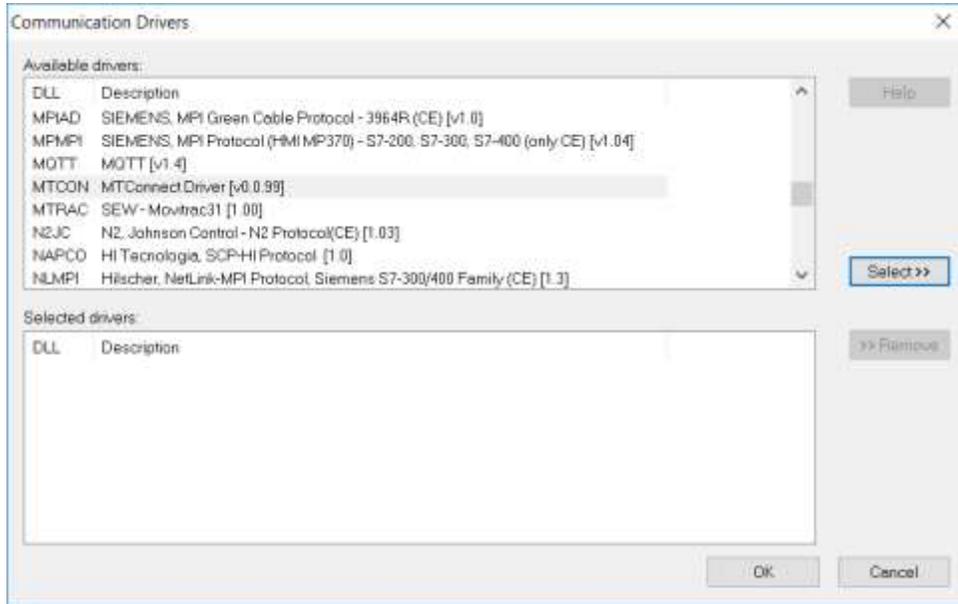
- **Driver Configuration (a):**
  - **PLC Program:** None
  - **Baud Rate:** -
  - **Protocol:** MTConnect (HTTP) Version 1.4
  - **Data Bits:** -
  - **Stop Bits:** -
  - **Parity:** -
  - **COM Port:** -
  - **TCP/IP Port:** 5000
  - **Cable:** Ethernet Cable

Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
1.00	8.1+SP1	Windows 10	Windows 10	– 5 PCs running simulator/3 cloud agents

## Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the `\DRV` subdirectory but they remain dormant until manually selected for specific applications. To select the MTCOON driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **MTCOON** driver from the *Available Drivers* list, and then click the **Select** button.



**Communication Drivers Dialog**

3. When the **MTCOON** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

**Note:**

It is not necessary to install any other software on your computer to enable communication between Studio and the target device. However, the MTCOON driver is used only by Studio; you may need to install additional software to program the device.

**Caution:**

For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

## Configuring the Driver

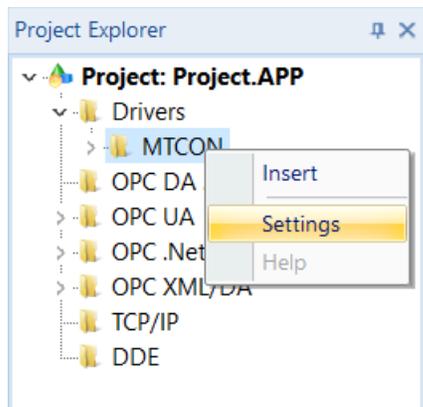
Once you have selected the MTCOIN driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver’s communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

### Configuring the Communication Settings

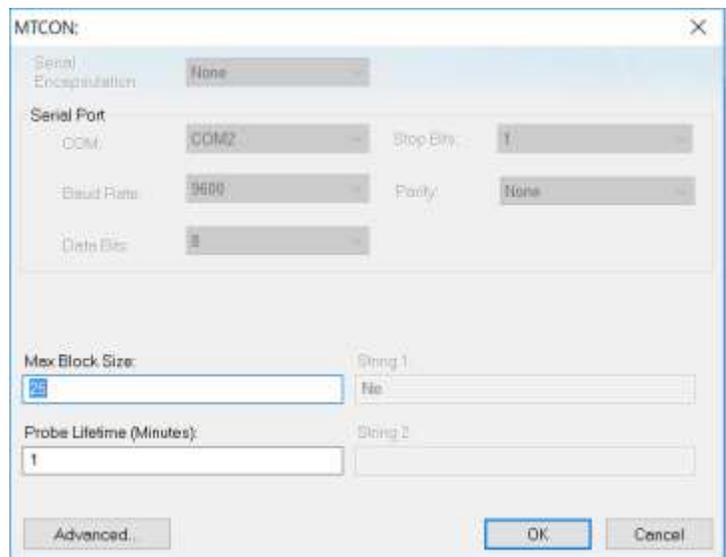
The communication settings are described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only MTCOIN driver-specific settings and procedures will be discussed here. To configure the communication settings for the MTCOIN driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The MTCOIN driver is listed here as a subfolder.
2. Right-click on the *MTCOIN* subfolder and then select the **Settings** option from the pop-up menu. The *MTCOIN: Communication Parameters* dialog is displayed:



Select Settings from the Pop-Up Menu



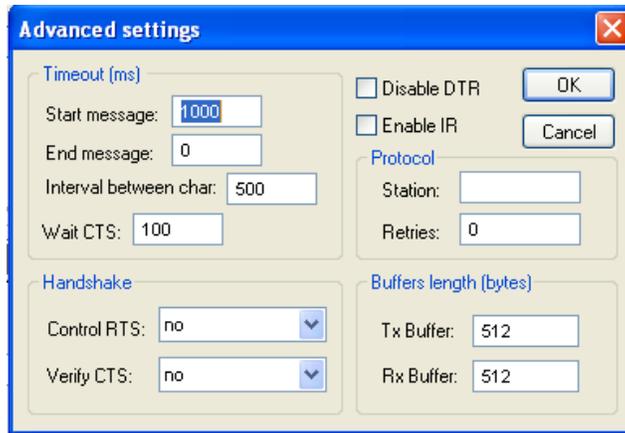
MTCOIN: Communication Parameters Dialog

3. Verify the *Serial Port* settings, and change them if necessary.

- Configure the additional driver-specific settings, as described in the following table:

Setting	Default Value	Valid Values	Description
Max Block Size	25	Integer Value greater than 1	Max number of Dataltems to be requested in a single transaction.
Probe Lifetime (minutes)	1440 (1 day)	110 to 115200	Probe lifetime specified the period of time between probe requests, default value is set to 1440 min (1 day).

- In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:



**Advanced Settings Dialog**

*Start Timeout* is the only parameter that is used by the MTCOON driver, that specifies the maximum timeout for a request.

- Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

## Configuring the Driver Worksheets

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

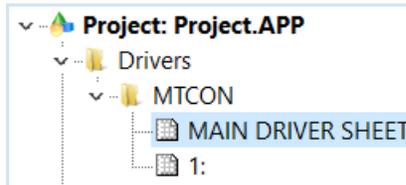
The configuration of these worksheets is described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only MTCOON driver-specific parameters and procedures will be discussed here.

## MAIN DRIVER SHEET

When you select the MTCO driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *MTCO* driver subfolder. To configure the Main Driver Sheet:

1. Select the *Comm* tab in the *Workspace* pane.
2. Open the *Drivers* folder, and then open the *MTCO* subfolder:



*Main Driver Sheet in the MTCO Subfolder*

3. Double-click on the MAIN DRIVER SHEET icon to open the following worksheet:

Tag Name	Station	IO Address	Action	Scan	Dir	Add
1 11	http://127.0.0.1:5000/VMC-3Axis/	11	Read+Write	Always		
2 12	http://127.0.0.1:5000/VMC-3Axis/	12	Read+Write	Always		
3 13	http://127.0.0.1:5000/VMC-3Axis/	13	Read+Write	Always		
4 14	http://127.0.0.1:5000/VMC-3Axis/	14	Read+Write	Always		
5 15	http://127.0.0.1:5000/VMC-3Axis/	15	Read+Write	Always		
6 16	http://127.0.0.1:5000/VMC-3Axis/	16	Read+Write	Always		
7 17	http://127.0.0.1:5000/VMC-3Axis/	17	Read+Write	Always		
8 18	http://127.0.0.1:5000/VMC-3Axis/	18	Read+Write	Always		
9 19	http://127.0.0.1:5000/VMC-3Axis/	19	Read+Write	Always		
10 110	http://127.0.0.1:5000/VMC-3Axis/	110	Read+Write	Always		
11 111	http://127.0.0.1:5000/VMC-3Axis/	111	Read+Write	Always		

*Main Driver Sheet*

Most of the fields on this sheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the MTCO driver.

- **Station** field (*not used for Serial communication*): Specify the IP Address of the device, using the following syntax:

*<IP address or URL>[:Port Number]/[Device Name]*

Example — 192.168.2.15:5000/CNC-12

anyvalidurl.com:5000/CNC-123

Where:

- **<IP Address or URL>** is the device's IP address or the URL of the agent
- **[Port Number]** is the TCP/IP port number to which you send messages. This parameter is optional (if not specified the default port is 80)
- **I/O Address** field: This field must contain the Dataltem values of the items required to read.

The MTConnect Driver specifies addresses using the Data Item ID. This ID is case sensitive, and must adhere to the W3C Standard ID type. The ID must start with a colon, comma, underscore, or letter (A-Z or a-z). It can then be followed with numbers, letters, a hyphen, or a period.

 **Note:**

**If the programming software of the device does not provide this information, since MTConnect is a HTTP protocol, open the station value on the web browser it will provide an XML document with PROBE information containing all Dataltems available. (Search for Dataltem Tag "<Dataltem>")**

Example of XML from station: <http://localhost:5000/VMC-3Axis/>

```
<MTConnectDevices ...>
<Header .../>
<Devices>
<Device id="dev" iso841Class="6" name="VMC-
3Axis" sampleInterval="10" uuid="000">
<Description manufacturer="SystemInsights"/>
<DataItems>
<DataItem category="EVENT" id="avail" type="AVAILABILITY"/>
...
```

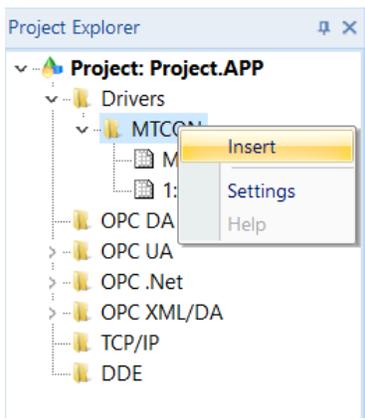
In this specific case "avail" is a Dataltem that can be specified on the address

## STANDARD DRIVER WORKSHEET

When you select the MTCO<sup>N</sup> driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

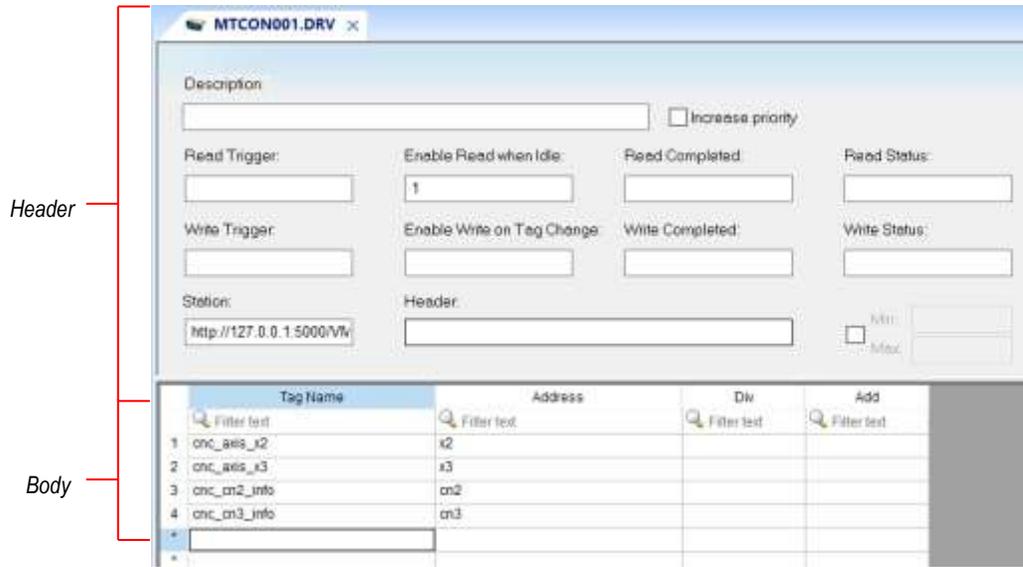
To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *MTCO<sup>N</sup>* subfolder.
2. Right-click on the *MTCO<sup>N</sup>* subfolder, and then select **Insert** from the pop-up menu:



**Inserting a New Worksheet**

A new MTCO<sup>N</sup> driver worksheet is inserted into the *MTCO<sup>N</sup>* subfolder, and the worksheet is opened for configuration:



**MTCO<sup>N</sup> Driver Worksheet**

 **Note:**

Worksheets are numbered in order of creation, so the first worksheet is **MTCOON001 .drv**.

Most of the fields on this worksheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the MTCOON driver.

3. Configure the **Station** and **Header** fields as follows:

- **Station** field (*not used for Serial communication*): Specify the IP Address of the device, using the following syntax:

*<IP address or URL>[:Port Number]/[Device Name]*

Example — 192.168.2.15:5000/CNC-12

`http://anyvalidurl.com:5000/CNC-123`

Where:

- *<IP Address or URL>* is the device’s IP address or the URL of the agent
- *[Port Number]* is the TCP/IP port number to which you send messages. This parameter is optional (if not specified the default port is 80)
- **Header** field: leave this field blank.
- **Address** field: This field must contain the DataItem values of the items required to read.

The MTCOON Driver specifies addresses using the Data Item ID. This ID is case sensitive, and must adhere to the W3C Standard ID type. The ID must start with a colon, comma, underscore, or letter (A-Z or a-z). It can then be followed with numbers, letters, a hyphen, or a period.

## Data Type Description

Studio will convert the received data to the Studio type based on the MTConnect datatypes. The following table shows the relation between MTConnect type and Studio type.

### Sample Dataltem

MTConnect Datatype	Studio Type
ACCELERATION	Float
ACCUMULATED_TIME	Float
ANGULAR_ACCELERATION	Float
ANGULAR_VELOCITY	Float
AMPERAGE	Float
ANGLE	Float
AXIS_FEEDRATE	Float
CLOCK_TIME	Float
CLOCK_TIME	Float
CONCENTRATION	Float
CONDUCTIVITY	Float
DISPLACEMENT	Float
ELECTRICAL_ENERGY	Float
EQUIPMENT_TIMER	Float
FILL_LEVEL	Float
FLOW	Float
FREQUENCY	Float
LENGTH	Float
LINEAR_FORCE	Float
LOAD	Float
MASS	Float
PATH_FEEDRATE	Float
PATH_POSITION	String
PH	String

POSITION	Float
POWER_FACTOR	Float
PRESSURE	Float
PROCESS_TIMER	Float
RESISTANCE	Float
ROTARY_VELOCITY	Float
SOUND_LEVEL	Float
SPINDLE_SPEED	Float
STRAIN	Float
TEMPERATURE	Float
TENSION	Float
TILT	Float
TORQUE	Float
VOLT_AMPERE	Float
VOLT_AMPERE_REACTIVE	Float
VELOCITY	Float
VISCOSITY	Float
VOLTAGE	Float
WATTAGE	Float

**Event Dataltem**

MTCConnect Datatype	Studio Type
ACTUATOR_STATE	String
ALARM	String
ACTIVE_AXES	String
AVAILABILITY	String
AXIS_COUPLING	String
AXIS_FEEDRATE_OVERRIDE	Float

AXIS_INTERLOCK	String
BLOCK	String
BLOCK_COUNT	Integer
CHUCK_INTERLOCK	String
CHUCK_STATE	String
CODE	String
COMPOSITION_STATE	String
CONTROLLER_MODE	String
CONTROLLER_MODE_OVERRIDE	String
COUPLED_AXES	String
DIRECTION	String
DOOR_STATE	String
END_OF_BAR	Bool
EMERGENCY_STOP	String
EQUIPMENT_MODE	String
EXECUTION	String
FUNCTIONAL_MODE	String
HARDNESS	String
INTERFACE_STATE	String
LINE	String
LINE_LABEL	String
LINE_NUMBER	Integer
MATERIAL	String
MESSAGE	String
OPERATOR_ID	String
PALLET_ID	String
PART_COUNT	Integer
PART_ID	String
PART_NUMBER	String
PATH_FEEDRATE_OVERRIDE	String
PATH_MODE	String

POWER_STATE	Bool
POWER_STATUS	String
PROGRAM	String
PROGRAM_EDIT	String
PROGRAM_EDIT_NAME	String
PROGRAM_COMMENT	String
PROGRAM_HEADER	String
ROTARY_MODE	String
ROTARY_VELOCITY_OVERRIDE	Float
SERIAL_NUMBER	String
SPINDLE INTERLOCK	String
TOOL_ID	String
TOOL_ASSET_ID	String
TOOL_NUMBER	String
TOOL_OFFSET	String
USER	String
WIRE	String
WORKHOLDING_ID	String
WORK_OFFSET	String

**Condition Datatem**

MTCConnect Datatype	Studio Type
ACTUATOR	String
CHUCK_INTERLOCK	String
COMMUNICATIONS	String
DATA_RANGE	String
DIRECTION	String
END_OF_BAR	String
HARDWARE	String

INTERFACE_STATE	String
LOGIC_PROGRAM	String
MOTION_PROGRAM	String
SYSTEM	String

**Note:**  
 Quality of the tags varies according to the communication state, MTConnect can use a specific string value “UNAVAILABLE” to indicate that the current value of the Dataltem is unavailable in this case the driver’s tag is considered UNCERTAIN.

Database Spy			
Taq/Expression	Value	Quality	Continuous
test4	0.000000	UNCERTAIN	<input checked="" type="checkbox"/>
test5	0.000000	UNCERTAIN	<input checked="" type="checkbox"/>
test6	UNAVAILABLE	UNCERTAIN	<input checked="" type="checkbox"/>
test7	0.000000	UNCERTAIN	<input checked="" type="checkbox"/>
test8	UNAVAILABLE	UNCERTAIN	<input checked="" type="checkbox"/>
test9	UNAVAILABLE	UNCERTAIN	<input checked="" type="checkbox"/>
test10	UNAVAILABLE	UNCERTAIN	<input checked="" type="checkbox"/>

## Timestamp Information

MTConnectg Agents provide current information about Dataltems. This information contains the value and the timestamp of the device/agent. If the Studio find an appropriate timestamp of the Dataltems it will use it and the tag associated with the Dataltem will have the timestamp received, if not then Studio will use the timestamp of the local machine.

Database Spy			
Tag/Expression	Value	Quality	Continuous
avail_motor1->timestamp	08/31/2018 06:55:22.832	GOOD	<input checked="" type="checkbox"/>
avail_motor2->timestamp	08/31/2018 06:55:25.518	GOOD	<input checked="" type="checkbox"/>
avail_motor3->timestamp	08/31/2018 06:55:29.048	GOOD	<input checked="" type="checkbox"/>
avail_motor4->timestamp	08/31/2018 06:55:20.986	GOOD	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

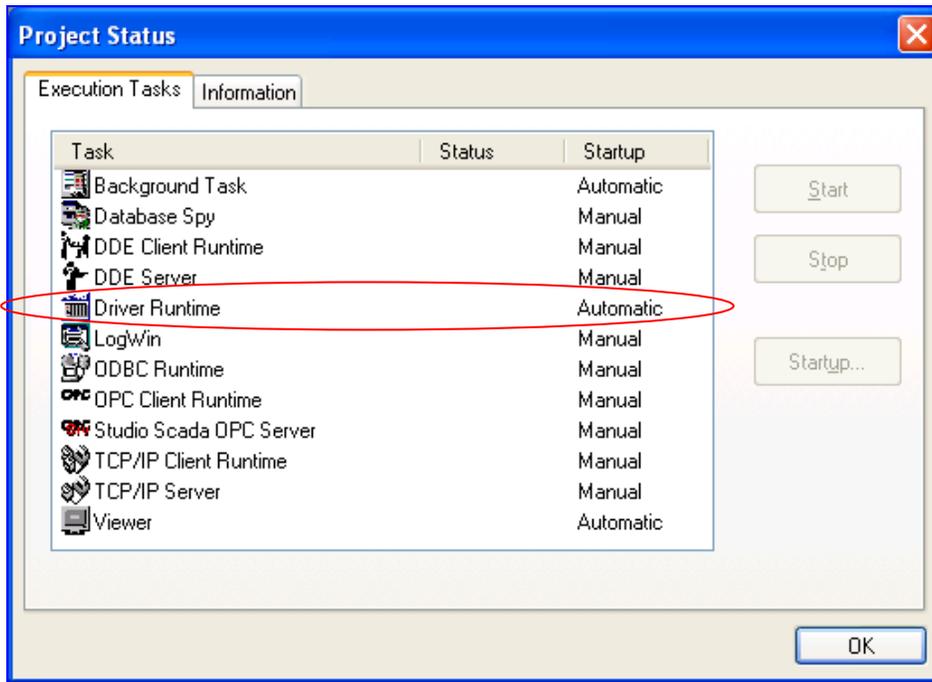
**Database Spy**

## Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application’s runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



*Project Status Dialog*

2. Verify that the *Driver Runtime* task is set to **Automatic**.
  - If the setting is correct, then proceed to step 3 below.
  - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task’s *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

## Troubleshooting

If the MTCOON driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Standard Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communicating without problems	Not required
99	Invalid Station	Station specified is invalid	Check manual for the right station format to use
100	Error initializing API	Unable to load or find required API	-
101	Error retrieving data	Problem on the Agent/Network or invalid station	Check Agent status, confirm that it is running by testing using the browser. Check the station information.
102	Error parsing mtconnecterror response	Invalid XML format received	The Agent sent an error message but the error message was corrupted or unable to be parsed by the driver. Please check the addresses and station used.
103	Device returned error	Device returned an error, check the output window for more details	<p>Check the output window for more details about the error. Some possible error conditions:</p> <p><b>INTERNAL_ERROR:</b> The MTConnect Agent experienced an error while attempting to published the requested information. Check Agent status.</p> <p><b>INVALID_REQUEST:</b> The Request contains information that was not recognized by the MTConnect Agent. Check addresses information.</p> <p><b>INVALID_URI:</b> The station provided is invalid</p> <p><b>NO_DEVICE:</b> Check the “Device Name” parameter value on the Station field.</p> <p><b>QUERY_ERROR:</b> Check the addresses specified.</p> <p><b>TOO_MANY:</b> Check the “Device Name” parameter value on the Station field.</p> <p><b>ASSET_NOT_FOUND:</b> Check the “Device Name” parameter value on the Station field.</p> <p><b>OUT_OF_RANGE:</b> Check the “Device Name” parameter value on the Station field.</p> <p><b>UNSUPPORTED:</b> The Agent does not support the current request, try to access the agent using the browser.</p> <p><b>UNAUTHORIZED:</b> Driver does not have permissions to access the requested data</p> <p><b>INVALID_XPATH:</b> Unable to find any of the requested Dataltems, this means that all addresses provided on this group are wrong. check the addresses provided. A list of all addresses of the current group will be provided on the output window.</p>

			Example: [ID: 1838] MTCOON Read group: http://127.0.0.1:5000/VMC-3Axis/ > Device returned error (Group number: 1) Error expected MTConnectStreams element Device returned an error INVALID_XPATH Check if the following Dataltems exists on the Agent side: velgi2 axis39 cn29 cn39 motor3 auto4msg
104	Device returned less than requested	Agent returned less items than requested	Check the address specified
105	Invalid Operation	The operation is not supported	Write operations are not supported on MTConnect Driver
106	Invalid LifeTime value	The value specified for LifeTime is invalid	Specify a valid LifeTime value. Check the Driver Settings topic for more information.
107	Timeout value must be positive	The value specified for Timeout is invalid	Specify a valid Timeout value. Check the Driver Advanced Settings topic for more information.
108	Timeout	Agent did not respond before the specified timeout	Check the Timeout value specified under Driver Advanced Settings
109	Invalid Blocksize value	The value specified for blocksize is invalid	Specify a valid Blocksize value. Check the Driver Advanced Settings topic for more information.

⇒ **Tip:**  
 You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Protocol Analyzer**, right-click in the *Output* window and select the desired options from the pop-up menu.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Project Information**: To find this information, select **Project** → **Status**.
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

## **Sample Application**

There was not an official sample application available for this driver by the time that this document was written.

## Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Paulo Balbino	29 Aug 2018	Initial version