

Johnson Controls International Metasys Driver

File Name	Metasys.dll
Manufacturer	Johnson Controls International
Devices	Metasys Server v10 or later
Protocol	Metasys API v3/v4
Version	1.0.14
Last Update	02/02/2026
Platform	Win32
Dependencies	IOKIT 2.0
Superblock Readings	Not using superblocks
Level	31314

Introduction

This Driver implements the communication with Metasys Server v10 or later by Johnson Controls International through Metasys API v3/v4.

Driver Configuration

All configurations are performed on Driver's configuration dialog box.

Additionally, shall be defined manually the files **MetasysState0.txt** and **MetasysState1.txt** containing the state names corresponding to the values 0 e 1 respectively, so the driver can translate these states for numerical values 0 and 1.

The files shall contain a list of words, one at each line. A word placed at the state 0 file shall not be placed at the state 1 file, and vice-versa.

All words are converted to lowercase internally, so the words are not case-sensitive.

Rows starting with the symbol # will be considered comments.

Example:

MetasysState0.txt

Stopped

Local

Close

Off

Inactive

FALSE

Deactivate

Normal

MetasysState1.txt

Started

Start

Reset

On

Remote

Open

Active

TRUE

Activate

Alarm

It is also possible to create a file called **MetasysReadAhead.txt** which informs a part of a command name which is associated with a status tag, which can be read immediately after.

The file will be read at driver startup, with the following content :

```
#comments starts with #
#put here the keywords to associate the command to a state
#keyword_command;keyword_state
#objects shall belong to the same controller
QDL-S;QDL-C
SF-COM;SF-ST5
SYSENA;SF-ST5
SF-CVF;SF-SVF
```

The procedure starts after a successful command. If the command tag contains the first keyword, the keyword is replaced by the second keyword and the driver checks if a status tag exists with that name. If positive, this tag is inserted into a special queue waiting for the wait time (Read Ahead After Cmd – Wait Time) to be read.

Configuring Properties

The parameters to define the communication with Metasys system are defined at the driver configuration window.

PROPERTY	DESCRIPTION
Metasys Server	DNS Name or IP Address for the Metasys Server or NAE Concentrator
User/Password	Username and Password for Login
User2/Password2	Username and Password for alternative Login in case of failure of first user account (after 2 retries)
Cache File Path	Folder name where cache files will be read and saved. There are 3 types of cache files: - GUID File (MetasysGUID.txt): Contains the internal codes of each Metasys object, which are stored at this file to speed up the startup process. - Command and State Cache Files (DeviceName_MetasysStateCache.txt): Contains the description of commands and states of the objects marked with N1>0, stored in a cache file per NAE controller, to speed up the startup and translation process for the numeric values 0 or 1. - State words (MetasysState0.txt and MetasysState1.txt): contains the state names corresponding to the values 0 and 1 respectively. If the cache file path is not informed, the files will be saved at the same folder where the driver dll is located.
Force New Login After (s)	Force new login independently of the token expire time. 0 = disable
Protocol Version	V3: Uses only polling for reading V4: Allows polling and events (subscriptions)
Read Method	Polling: performs tag readings by polling. Can be used both V3 and V4; Events: performs tag readings by subscriptions and can be used only with V4; ByTag(N2): Allows choosing the read method by tag at the N2 parameter (0 = polling, 1 = events). It can be used only with V4.
Subscription Renew (s)	When a tag is read by subscription/events, if its value doesn't change during this interval (in seconds). a new subscription is requested, which forces receiving a new update. If the

PROPERTY	DESCRIPTION
	received value is the same as the current value, the tag timestamp will not be updated, except if the EnableDeadBand property is FALSE.
Polling Options	Options related to polling reading
Items per Read Request	Tells how many items can be added to the same read request. Minimum value is 1. This property is also used to define how many items can be subscribed in a single subscription request.
Inter-Request Delay (ms)	Informs a default delay in milliseconds between read requests for all NAE's. This delay will be used if a NAE doesn't have its own delay informed at the tag InterReqDelayMS.
Force New Login After Consec Errors	Force new login after a number of consecutive errors. 0 = disable
Force New Login with Read Error Rate > %	Force new login if error rate (in %) is greater than what is informed. 0 = disable
Read Ahead After Cmd - Wait Time (s)	Tempo de espera para realizar a leitura de um tag de status associado a um comando. Esta associação é feita através do arquivo MetasysReadAhead.txt
Make Parallel Requests per Device	Informs if the the driver shall send parallel requests for each NAE declared.
Full Logs	Informs if the driver logs (when enabled) will be generated with detailed information.

Beyond the Properties window, these configurations can also be defined in runtime at **Elipse E3** or **Elipse Power** applications. For this, start the driver in **Offline** mode, i.e. execute the application with the option **Start driver OFFLINE** enabled, configurable at the **Setup** tab at Property window. The options are described below.

Config Options for Johnson Controls International Metasys Driver

PARAMETER	OFFLINE STRING PROPERTY	DATATYPE	DESCRIPTION
Metasys Server	Metasys.UseChecksum	Text	Metasys Server Name
User	Metasys.User	Text	User for Login
Password	Metasys.Password	Text	Password for Login
Cache File Path	Metasys.GUIDPath	Text	Cache Folder
Items per Read Request	Metasys.MaxItemsRead	Text	Max Items per request
Protocol Version	Metasys.APIVersion	DWord	3 or 4
Read Method	Metasys.ReadConfig	DWord	0 = polling, 1 =Events, 2 = by Tag
Inter Request Delay	Metasys.InterReqDelay	Dword	Delay between readings
Force New Login after (s)	Metasys.ForceLoginTime	Dword	Force new login after x seconds
Force New Login after consec errors	Metasys.ForceLoginConsecErrors	Dword	Force new login after x consecutive errors
Force New Login with read error rate	Metasys.ForceLoginReadErrorRate	Dword	Force new login if error rate is greater than informed
Read Ahead After Cmd - Wait Time	Metasys.ReadAheadWaitTime	Dword	Wait time to read a status tag after command

PARAMETER	OFFLINE STRING PROPERTY	DATATYPE	DESCRIPTION
Full Logs	Metasys.FullLogs	Int	Enable full logs

All offline properties must be configured via PLC Tags by using parameters *N1* equal to -1 (minus one), *N2* equal to 0 (zero), *N3* equal to 0 (zero), and *N4* equal to 3 (three).

Tag Reference

IOTags (individual tags) shall be used for communication.

Tags are addressed through the parameters **Device**, **Item**, **N1**, **N2** and **N3** at the following format:

Property	Meaning
Device	Metasys_Server_Name:NAE_Device
Item	MetasysObject;property{Extra} See more details below in Addressing :
N1	Object Type: 0 = Standard 1 = Enum (will be converted to a number) 2 = Analog with alarm limits
N2	Read type, if the Read Method property = By Tag (N2) 0 = Polling 1 = Events
N3	Retries for alarm limits retries (N1 = 2)

Addressing

At item parameter, **MetasysObject** is the name of a metasys system object, and **property** is an optional parameter which informs the property to be read. If not informed, the property will be **presentValue**.

If a special property name **PVALM** is informed, the properties **presentValue** and **alarmState** will be read, with the first one mapped to the tag property Value and the second to the property Quality. If the **alarmState** Value is HighAlarm or Alarm, it will be mapped as OPC_LIMIT_HIGH (bit 1 - Quality) and if the values is LowAlarm it will be mapped as OPC_LIMIT_LOW (bit 0 - Quality), resulting respectively at the values 194 (OPC_GOOD + OPC_LIMIT_HIGH) and 193 (OPC_GOOD + OPC_LIMIT_LOW).

Extra is a JSON formatted text where you can inform a list of discrete states (States) or an Alarm Name (Alarm).

States is a JSON where we can declare state names and corresponding values for each tag. The driver will try first to use the state names declared at the tag, if not found, it will use the state names declared at the state files 0 and 1.

The expected States format is:

```
{"states" : [ {"Name1":Value1}, {"Name2":Value2} , {"NameN":ValueN}]}
```

Alarm is a JSON where you can declare, only for analog tags, a children alarm name inside Metasys system, at the format: **{ "alarmName": "Name" }**

This alarm object will have its limits (Baixo e Alto, Warning e Alarm) read and saved into a file through the special tag "GetAlarmLimits".

Some tag examples:

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1 => will be read the presentValue

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1;presentValue => will be read the presentValue

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1;PVALM => will be read the presentValue and alarmState as explained above

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1{"alarmName": "Alarm1"} => will be read the presentValue with Alarm object as "Alarm1"

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: FieldBus1.4oPav.FX34.BinaryInputs;status => the property status will be read

Device: WIN-CGVCS5PQ7TV:NAE4561 Item: Programming.AV1{"states": [{"Local": 0}, {"Remote": 1}, {"Automatic": 2}]} => will be read the presentValue, interpreting its contents as the states Local, Remote and Automatic.

Note: If the returned value of a property is an Enum and N1=1 the driver will try to translate the text for a corresponding integer value, based on the content of the files MetasysState0.txt and MetasysState1.txt.

For write operations, the driver obtains the command name as the property field, and sends whatever is received as the tag value, performing the translation to the corresponding Enum state if necessary.

Special Tags

Name	Device	Item	Read/Write	Description
ClearStateCache	Metasys_Server_Name:NAE_Device	ClearStateCache	R/W	Erases the command and state cache for a specific NAE controller, without stopping the system execution. The possible values are: 0: Idle, ready to start 1: Processing 2: Finished, cache file saved To start the process, write to the tag the value 1. It will only be accepted if the current value is 0 (Idle). The tag value will be 2 automatically when the process is finished, and stays at this value until receiving a write with value 0, being ready to start again if necessary.
FlushGUID	Metasys_Server_Name:NAE_Device	FlushGUID	W	Saves the driver GUID file (with GUID's of all NAE's of this driver)
GetGUID	Metasys_Server_Name:NAE_Device	GetGUID:ObjectName	R	Returns the ObjectName GUID if it has already been obtained. Returns error if ObjectName doesn't exist or if it is still under processing.

GetDeviceEnumCount	Metasys_Server_Name:NAE_Device	GetDeviceEnumCount	R	It shall be a block with 2 elements. The first element returns the total object count which contains at least one tag of Enum type (N1 = 1) declared. The second returns from the object count of the previous element, how many have its commands and states already been translated from the Metasys Server.
GetAlarmLimits	Metasys_Server_Name:NAE_Device	GetAlarmLimits	R/W	Request or Get the status of alarm limit read operation, for the tags marked with N1=2. This tag value is: 0 = Idle (Ready to start collection) 1 = Pending (Request Pending) 2 = Executing (Under treatment) 3 = Finished OK: All objects were read, ready to store file 4 = Finished Error: At least one object didn't return its limits When tag value is 3 or 4, it is necessary to write the value 0 to the tag, so the driver will write to disk a file named NAEDevice_MetasysAlarmInfo.txt with a line for each tag, at the following format: Path;GUID;Message;LowAlarm;HighAlarm;LowWarning;HighWarning;Delay;DelayActive
InterReqDelayMS	Metasys_Server_Name:NAE_Device	InterReqDelayMS	W	Informs a Delay in milliseconds between read requests per NAE.
LoginStatus	Metasys_Server_Name:NAE_Device	LoginStatus	R	Informs if the driver is logged at the Metasys Server. The NAE device informed at the Device field can be any, as the driver makes only one login, independently of the number of NAE devices.
MaxItemsRead	Metasys_Server_Name:NAE_Device	MaxItemsRead	W	Informs the maximum number of items to read on each read request for this particular NAE device. If not informed, it uses the default value informed at the driver dialog.
CallStats	If empty: Get stats from all devices. For a specific device, specify as "Metasys_Server_Name:NAE_Device"	CallStats5 (5 minutes) CallStats60 (1 hour)	(5)R	Shall be a block with 12 elements: 0: Total : Total API Calls 1: Timeout: Total calls with timeout 2: Error: Calls finished with error (code != 200) 3: ErrorCodes: List with error codes and its quantities, separated by semicolon. Ex: "401:1234;403:57" 4: OK : Total Success calls 5: TotalReads: Total of read polling calls

				<p>6: TimeoutReads: Read calls with timeout</p> <p>7: ErrorReads: Read calls finished with error (code != 200)</p> <p>8: OKReads: Success Read Calls</p> <p>9: TimeOKReadp50 : Read Response time, percentile 50</p> <p>10: TimeOKReadp95 : Read Response time, percentile 95</p> <p>11: TimeOKReadp99 : Read Response time, percentile 99</p>
SubscriptionStats	If empty: Get stats from all devices. For a specific device, specify as "Metasys_Server_Name:NAE_Device"	SubscriptionStats5 (5 minutes) SubscriptionStats60 (1 hour)	R	<p>Shall be a block with 9 elements:</p> <p>0 : Total Subscribed (Total subscribed properties, since driver startup)</p> <p>1: Requested (Total calls requests with subscriptions)</p> <p>2: Accepted (Total calls with response ok, code 200)</p> <p>3: Failed (Total calls with call failure or timeout)</p> <p>4: Error (Total calls in error, code different from 200)</p> <p>5: Update Messages (Subscription notifications containing data)</p> <p>6: Error Messages (Subscription Error Messages)</p> <p>7: Updated Objects (Number of objects updated)</p> <p>8: Updated Properties (Number of properties updated)</p>

Driver Revision History

VERSION	DATE	AUTHOR	COMMENTS
1.0.14	02/02/2026	M. Salvador	<ul style="list-style-type: none"> [BUG] Fixed excessive logs in case of communication failure. [36510] { fixed on Metasys v 1.00 Build 12 } [BUG] Implemented blocking of tags with syntax error. [36946] { fixed on Metasys v 1.00 Build 13 } [BUG] - Now new devices can be added in runtime. [38214] { fixed on Metasys v 1.00 Build 13 }
1.0.13	08/16/2025	M. Ludwig	Driver updated to IOKit library version 3.0 and Visual Studio 2022 (Case 37987).

VERSION	DATE	AUTHOR	COMMENTS
1.0.9	03/21/2024	M. Salvador	Added support to API V4.
1.0.8	09/20/2022	M. Salvador / M.Ludwig	Initial driver version.

Headquarters

**Rua Mostardeiro, 322/Cj. 902, 1001 e
1002**

90510-002 — Porto Alegre — RS

Phone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: elipse-rs@elipse.com.br

Branch in Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Phone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

Check our website for information about a representative in your country.

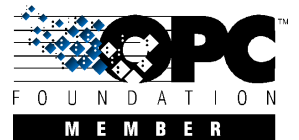
www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner
Gold Independent Software Vendor (ISV)