

Schlumberger Q1000 (miniDLMS) Driver

File Name	Q1000.dll
Manufacturer	Schlumberger Technologies
Devices	Schlumberger Q1000 power meters
Protocol	miniDLMS (Master)
Version	2.0.8
Last Update	04/30/2026
Platform	Win32
Dependencies	IOKit version 1.8 or later
Superblock Readings	No
Level	31202

Introduction

This is the Schlumberger Q1000 (miniDLMS) Driver for communication between Eclipse Software products and Q1000 power meters by Schlumberger Technologies.

Before using this Driver, it is recommended to check topic **Notes on Quality** and verify if there is any reference to resources needed for an application. This topic lists resources not yet sufficiently tested or used, which may require more attention and previous tests before using them in a production environment.

Driver Settings

Open the properties window of this Driver to define the destination address of a Q1000 meter and the internal timer to perform retries for communication failures. The **Default Slave Address** option defines the physical address attributed to a Q1000 meter. The **Internal Retry Error Timer** option defines a maximum time, in seconds, that this Driver retries due to a communication failure.

To define these options in **Offline** mode using the **IOKit** library, use the **Q1000.DefaultSlaveAddress** Tag to define a value for the **Default Slave Address** option and the **Q1000.RetryErrorTimer** Tag to define a value for the **Internal Retry Error Timer** option. These Tags must be configured only in **Offline** mode using the *N1* parameter equal to -1 (minus one), the *N2* parameter equal to 0 (zero), the *N3* parameter equal to 0 (zero), and the *N4* parameter equal to 3 (three). For more information and examples, please check topic **Documentation of I/O Interfaces**.

[P] Parameters

P1	Address of a Master in the network. This is the address this Driver assumes in the network when requesting data from meters. Allowed values are in the range between 1 (one) and 254, but protocol's documentation recommends using values between 1 (one) and 32
P2	Not used, leave it in 0 (zero)
P3	Not used, leave it in 0 (zero)
P4	Not used, leave it in 0 (zero)

Tag Reference

This section contains information about the configuration of this Driver's **[N/B]** Tags.

Addressing Parameters of PLC Tags

This section contains PLC Tags with general data about a meter.

Serial Number of a Meter

Read-Only

N1	Address of a meter in the network
N2	1 (one)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Text** format for reading a 24-character value corresponding to the serial number of a meter.

MAE's Identification Number

Read-Only

N1	Address of a meter in the network
N2	2 (two)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Text** format for reading a 32-character value corresponding to a **MAE's** identification number.

NOTE

MAE stands for *Mercado Atacadista de Energia (Power Wholesale Market)*, currently **CCEE** (*Câmara de Comercialização de Energia Elétrica, or Power Trading Chamber*).

Integration Constant

Reading and Writing

N1	Address of a meter in the network
N2	3 (three)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag for reading a time interval, in seconds.

Access Password

Reading and Writing

N1	Address of a meter in the network
N2	4 (four)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Text** format for reading a 20-character value corresponding to an access password.

Communication Test

Read-Only

N1	Address of a meter in the network
N2	5 (five)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Text** format for reading a 12-character value ("SI,QM110:nnn"), in which *nnn* is the number of a meter, between "001" and "255".

NOTE

This communication test is only valid when there is only one meter connected to a bus, otherwise all meters respond to this command and this Driver receives garbage.

TC

Read-Only

N1	Address of a meter in the network
N2	6 (six)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Float** format for reading the TC (*Transformador de Corrente* or *Current Transformer*) value.

TP

Read-Only

N1	Address of a meter in the network
N2	7 (seven)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Float** format for reading the TP (*Transformador de Potência* or *Voltage Transformer*) value.

Time of a Meter

Reading and Writing

N1	Address of a meter in the network
N2	8 (eight)
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag in **Date and Time** format for reading the time of a meter or for setting the time of a meter.

Mass Memory's Reading Thread

Reading and Writing

N1	Address of a meter in the network
N2	999
N3	Sequential identifier of a parameter for a device
N4	Not used, leave it in 0 (zero)

Tag for reading that contains the value 0 (zero, the thread has finished) or the number of records read up to the moment.

Enable New Reading of a Log Event in a Meter

Reading and Writing

N1	Address of a meter in the network
N2	998
N3	Not used, leave it in 0 (zero)
N4	Not used, leave it in 0 (zero)

Tag used to enable a new reading of the log of events in a meter using the **Reading Logs of Events** Tag. Possible values are **0**: Event log reading is disabled or **1**: Enables a new event log reading in the meter.

Addressing Parameters of Block Tags

This section contains Block Tags for performing the process of collecting mass memory based on a user-defined date interval, and also for reading snapshot values.

Defining Parameters for Reading Values for Historic

Write-Only

B1	Address of a meter in the network
B2	1 (one)
B3	Sequential identifier of a parameter for a device
B4	Memory to query. For more information, please check table Types of memory

Use 2 (two) Elements to define a query for **Sag**, **Swell**, and **Interruptions** values or add from 3 (three) to 25 Elements to define a query for mass memory values.

Elements of this Block Tag

- **Element 1:** Initial date and time, in **Date and Time** format
- **Element 2:** Final date and time, in **Date and Time** format
- **Element 3 to 25 (only for the B4 parameter equal to 100 or the B4 parameter equal to 101):** Variables to retrieve. For more information, please check table **Mass memory variables**

Types of memory

B4 PARAMETER	TYPE OF MEMORY
100	Mass memory 1
101	Mass memory 2
300	SAG level 1, phase A
301	SAG level 2, phase A
302	SAG level 1, phase B
303	SAG level 2, phase B
304	SAG level 1, phase C
305	SAG level 2, phase C
400	SWELL level 1, phase A
401	SWELL level 2, phase A
402	SWELL level 1, phase B
403	SWELL level 2, phase B
404	SWELL level 1, phase C
405	SWELL level 2, phase C
500	Interruption class 1
501	Interruption class 2
502	Interruption class 3

Types of memory for multiple Sag and Swell readings by summing 16-bit masks (OR)

B4 PARAMETER (MASK)	TYPE OF MEMORY
1000000000000001	SAG level 1, phase A
1000000000000010	SAG level 2, phase A
10000000000000100	SAG level 1, phase B
10000000000001000	SAG level 2, phase B
1000000000010000	SAG level 1, phase C
1000000000100000	SAG level 2, phase C
1000000001000000	SWELL level 1, phase A
1000000010000000	SWELL level 2, phase A
1000000100000000	SWELL level 1, phase B
1000001000000000	SWELL level 2, phase B

B4 PARAMETER (MASK)	TYPE OF MEMORY
1000010000000000	SWELL level 1, phase C
1000100000000000	SWELL level 2, phase C

Mass memory variables

VALUE	DESCRIPTION
1	W delivered aggregate
2	W delivered phase A
3	W delivered phase B
4	W delivered phase C
5	W received aggregate
6	W received phase A
7	W received phase B
8	W received phase C
9	W net aggregate
10	W net phase A
11	W net phase B
12	W net phase C
13	W delivered aggregate SLC
14	W received aggregate SLC
15	W net aggregate SLC
16	VAR delivered aggregate
17	VAR delivered phase A
18	VAR delivered phase B
19	VAR delivered phase C
20	VAR received aggregate
21	VAR received aggregate phase A
22	VAR received aggregate phase B
23	VAR received aggregate phase C
24	VAR quadrant 1 aggregate
25	VAR quadrant 2 aggregate
26	VAR quadrant 3 aggregate
27	VAR quadrant 4 aggregate
28	VAR quadrant 1 phase A
29	VAR quadrant 2 phase A
30	VAR quadrant 3 phase A
31	VAR quadrant 4 phase A

VALUE	DESCRIPTION
32	VAR quadrant 1 phase B
33	VAR quadrant 2 phase B
34	VAR quadrant 3 phase B
35	VAR quadrant 4 phase B
36	VAR quadrant 1 phase C
37	VAR quadrant 2 phase C
38	VAR quadrant 3 phase C
39	VAR quadrant 4 phase C
40	VAR net aggregate
41	VAR net phase A
42	VAR net phase B
43	VAR net phase C
44	VAR delivered aggregate SLC
45	VAR received aggregate SLC
46	VAR quadrant 1 aggregate SLC
47	VAR quadrant 2 aggregate SLC
48	VAR quadrant 3 aggregate SLC
49	VAR quadrant 4 aggregate SLC
50	VAR net aggregate SLC
51	VA arithmetic delivered aggregate
52	VA arithmetic delivered phase A
53	VA arithmetic delivered phase B
54	VA arithmetic delivered phase C
55	VA arithmetic received aggregate
56	VA arithmetic received phase A
57	VA arithmetic received phase B
58	VA arithmetic received phase C
59	VA arithmetic total aggregate
60	VA arithmetic total phase A
61	VA arithmetic total phase B
62	VA arithmetic total phase C
63	VA vectorial delivered aggregate
64	VA vectorial delivered phase A
65	VA vectorial delivered phase B
66	VA vectorial delivered phase C
67	VA vectorial received aggregate
68	VA vectorial received phase A

VALUE	DESCRIPTION
69	VA vectorial received phase B
70	VA vectorial received phase C
71	VA vectorial total aggregate
72	VA vectorial total phase A
73	VA vectorial total phase B
74	VA vectorial total phase C
75	VA vectorial delivered aggregate SLC
76	VA vectorial received aggregate SLC
77	VA vectorial total aggregate SLC
78	Q delivered aggregate
79	Q received aggregate
80	Q net aggregate
81	Q delivered aggregate SLC
82	Q received aggregate SLC
83	Q net aggregate SLC
84	Volt average
85	Volts phase A
86	Volts phase B
87	Volts phase C
88	Amps Average
89	Amps phase A
90	Amps phase B
91	Amps phase C
92	Amps Neutral
93	Analog input 0 net
94	Analog input 1 net
95	Analog input 2 net
96	Analog input 3 net
97	Analog input 4 net
98	Analog input 5 net
99	Analog input 6 net
100	Analog input 7 net
101	Digital pulse input 0 net
102	Digital pulse input 1 net
103	Digital pulse input 2 net
104	Digital pulse input 3 net
105	Digital pulse input 4 net

VALUE	DESCRIPTION
106	Digital pulse input 5 net
107	Digital pulse input 6 net
108	Digital pulse input 7 net
109	Power factor phase A
110	Power factor phase B
111	Power factor phase C
112	Summed register 0 net
113	Summed register 1 net
114	Summed register 2 net
115	Summed register 3 net
116	Summed register 4 net
117	Summed register 5 net
118	Summed register 6 net
119	Summed register 7 net
120	VA arithmetic quadrant 1 aggregate
121	VA arithmetic quadrant 2 aggregate
122	VA arithmetic quadrant 3 aggregate
123	VA arithmetic quadrant 4 aggregate
124	VA arithmetic quadrant 1 phase A
125	VA arithmetic quadrant 2 phase A
126	VA arithmetic quadrant 3 phase A
127	VA arithmetic quadrant 4 phase A
128	VA arithmetic quadrant 1 phase B
129	VA arithmetic quadrant 2 phase B
130	VA arithmetic quadrant 3 phase B
131	VA arithmetic quadrant 4 phase B
132	VA arithmetic quadrant 1 phase C
133	VA arithmetic quadrant 2 phase C
134	VA arithmetic quadrant 3 phase C
135	VA arithmetic quadrant 4 phase C
136	VA vectorial quadrant 1 aggregate
137	VA vectorial quadrant 2 aggregate
138	VA vectorial quadrant 3 aggregate
139	VA vectorial quadrant 4 aggregate
140	VA vectorial quadrant 1 phase A
141	VA vectorial quadrant 2 phase A
142	VA vectorial quadrant 3 phase A

VALUE	DESCRIPTION
143	VA vectorial quadrant 4 phase A
144	VA vectorial quadrant 1 phase B
145	VA vectorial quadrant 2 phase B
146	VA vectorial quadrant 3 phase B
147	VA vectorial quadrant 4 phase B
148	VA vectorial quadrant 1 phase C
149	VA vectorial quadrant 2 phase C
150	VA vectorial quadrant 3 phase C
151	VA vectorial quadrant 4 phase C
152	Power factor aggregate
153	Power factor (arithmetic va) aggregate
154	Power factor (arithmetic va) phase A
155	Power factor (arithmetic va) phase B
156	Power factor (arithmetic va) phase C
157	Power factor (vectorial va) aggregate
158	Power factor (vectorial va) phase A
159	Power factor (vectorial va) phase A
160	Power factor (vectorial va) phase A
161	Watts for power factor aggregate
162	Watts for power factor phase A
163	Watts for power factor phase B
164	Watts for power factor phase C
165	VA for power factor aggregate
166	VA for power factor phase A
167	VA for power factor phase B
168	VA for power factor phase C
169	Percent THD volts phase A US
170	Percent THD amps phase A US
171	Percent THD volts phase A Europe
172	Percent THD amps phase A Europe
173	Percent THD volts phase B US
174	Percent THD amps phase B US
175	Percent THD volts phase B Europe
176	Percent THD amps phase B Europe
177	Percent THD volts phase C US
178	Percent THD amps phase C US
179	Percent THD volts phase C Europe

VALUE	DESCRIPTION
180	Percent THD amps phase C Europe
181	Watts Fundamental Delivered Aggregate
182	Watts Fundamental Delivered Phase A
183	Watts Fundamental Delivered Phase B
184	Watts Fundamental Delivered Phase C
185	Watts Fundamental Received Aggregate
186	Watts Fundamental Received Phase A
187	Watts Fundamental Received Phase B
188	Watts Fundamental Received Phase C
189	Watts Fundamental Net Aggregate
190	Watts Fundamental Net Phase A
191	Watts Fundamental Net Phase B
192	Watts Fundamental Net Phase C
193	VAR Fundamental Delivered Aggregate
194	VAR Fundamental Delivered Phase A
195	VAR Fundamental Delivered Phase B
196	VAR Fundamental Delivered Phase C
197	VAR Fundamental Received Aggregate
198	VAR Fundamental Received Phase A
199	VAR Fundamental Received Phase B
200	VAR Fundamental Received Phase C
201	VAR Fundamental Net Aggregate
202	VAR Fundamental Net Phase A
203	VAR Fundamental Net Phase B
204	VAR Fundamental Net Phase C
205	VAR Fundamental Quadrant 1 Aggregate
206	VAR Fundamental Quadrant 1 Phase A
207	VAR Fundamental Quadrant 1 Phase B
208	VAR Fundamental Quadrant 1 Phase C
209	VAR Fundamental Quadrant 2 Aggregate
210	VAR Fundamental Quadrant 2 Phase A
211	VAR Fundamental Quadrant 2 Phase B
212	VAR Fundamental Quadrant 2 Phase C
213	VAR Fundamental Quadrant 3 Aggregate
214	VAR Fundamental Quadrant 3 Phase A
215	VAR Fundamental Quadrant 3 Phase B
216	VAR Fundamental Quadrant 3 Phase C

VALUE	DESCRIPTION
217	VAR Fundamental Quadrant 4 Aggregate
218	VAR Fundamental Quadrant 4 Phase A
219	VAR Fundamental Quadrant 4 Phase B
220	VAR Fundamental Quadrant 4 Phase C
221	VA arithmetic unbalance total aggregate
222	VA arithmetic distortion total aggregate
223	VA vectorial Quadrant 1 Aggregate SLC
224	VA vectorial Quadrant 2 Aggregate SLC
225	VA vectorial Quadrant 3 Aggregate SLC
226	VA vectorial Quadrant 4 Aggregate SLC
227	Power factor (vectorial va) aggregate SLC
228	Watts Fundamental Delivered Aggregate SLC
229	Watts Fundamental Received Aggregate SLC
230	Watts Fundamental Net Aggregate SLC
231	VAR Fundamental Delivered Aggregate SLC
232	VAR Fundamental Received Aggregate SLC
233	VAR Fundamental Net Aggregate SLC
234	VAR Fundamental Quadrant 1 Aggregate SLC
235	VAR Fundamental Quadrant 2 Aggregate SLC
236	VAR Fundamental Quadrant 3 Aggregate SLC
237	VAR Fundamental Quadrant 4 Aggregate SLC
238	Frequency
239	Watts Transformer Loss Aggregate
240	Watts Transformer Loss Phase A
241	Watts Transformer Loss Phase B
242	Watts Transformer Loss Phase C
243	Vars Transformer Loss Aggregate
244	Vars Transformer Loss Phase A
245	Vars Transformer Loss Phase B
246	Vars Transformer Loss Phase C
247	Watts Line Loss Aggregate
248	Watts Line Loss Phase A
249	Watts Line Loss Phase B
250	Watts Line Loss Phase C
251	Vars Line Loss Aggregate
252	Vars Line Loss Phase A
253	Vars Line Loss Phase B

VALUE	DESCRIPTION
254	Vars Line Loss Phase C
255	Watts Net Phase A SLC
256	Watts Net Phase B SLC
257	Watts Net Phase C SLC
258	VAR Net Phase A SLC
259	VAR Net Phase B SLC
260	VAR Net Phase C SLC
261	Watts Fundamental Net Phase A SLC
262	Watts Fundamental Net Phase B SLC
263	Watts Fundamental Net Phase C SLC
264	VAR Fundamental Net Phase A SLC
265	VAR Fundamental Net Phase B SLC
266	VAR Fundamental Net Phase C SLC
267	Volt squared Phase A SLC
268	Volt squared Phase B SLC
269	Volt squared Phase C SLC
270	Amp squared Phase A SLC
271	Amp squared Phase B SLC
272	Amp squared Phase C SLC
273	Meter Input 1 quantity 1
274	Meter Input 1 quantity 2
275	Meter Input 1 quantity 3
276	Meter Input 1 quantity 4
277	Meter Input 1 quantity 5
278	Meter Input 1 quantity 6
279	Meter Input 2 quantity 1
280	Meter Input 2 quantity 2
281	Meter Input 2 quantity 3
282	Meter Input 2 quantity 4
283	Meter Input 2 quantity 5
284	Meter Input 2 quantity 6
285	Meter Input 3 quantity 1
286	Meter Input 3 quantity 2
287	Meter Input 3 quantity 3
288	Meter Input 3 quantity 4
289	Meter Input 3 quantity 5
290	Meter Input 3 quantity 6

VALUE	DESCRIPTION
291	Meter Input 4 quantity 1
292	Meter Input 4 quantity 2
293	Meter Input 4 quantity 3
294	Meter Input 4 quantity 4
295	Meter Input 4 quantity 5
296	Meter Input 4 quantity 6
297	Meter Input 5 quantity 1
298	Meter Input 5 quantity 2
299	Meter Input 5 quantity 3
300	Meter Input 5 quantity 4
301	Meter Input 5 quantity 5
302	Meter Input 5 quantity 6
303	Meter Input 6 quantity 1
304	Meter Input 6 quantity 2
305	Meter Input 6 quantity 3
306	Meter Input 6 quantity 4
307	Meter Input 6 quantity 5
308	Meter Input 6 quantity 6
309	Volt Squared Aggregate
310	Volt Squared Phase A
311	Volt Squared Phase B
312	Volt Squared Phase C
313	Amp squared Aggregate
314	Amp Squared Phase A
315	Amp Squared Phase B
316	Amp Squared Phase C
317	Volt line-line A-B
318	Volt line-line B-C
319	Volt line-line C-A
320	Volt line-line Average
321	Volt fundamental Phase A
322	Volt fundamental Phase B
323	Volt fundamental Phase C
324	Volt fundamental Average
325	Volt line-neutral Phase A
326	Volt line-neutral Phase B
327	Volt line-neutral Phase C

VALUE	DESCRIPTION
328	Volt line-neutral Average
329	Volt fundamental line-line A-B
330	Volt fundamental line-line B-C
331	Volt fundamental line-line C-A
332	Volt fundamental Average
333	Volt fundamental line-neutral Phase A
334	Volt fundamental line-neutral Phase B
335	Volt fundamental line-neutral Phase C
336	Volt fundamental line-neutral Average
337	Displacement PF Phase A
338	Displacement PF Phase B
339	Displacement PF Phase C
340	Displacement PF Average

Reading Mass Memory's Variables

Read-Only

B1	Address of a meter in the network
B2	2 (two)
B3	Sequential identifier of a parameter for a device
B4	Not used, leave it in 0 (zero)

This reading occurs when the *B1* and *B3* parameters refer to a **Defining Parameters for Reading Values for Historic** Tag that is configured with the *B4* parameter equal to 100 or 101.

Elements of this Block Tag

- **Timestamp:** Date and time of the registered value
- **Element 1:** Value with a bit mask containing the following items
 - **Bit 0:** Short interval
 - **Bit 1:** Long interval
 - **Bit 2:** Test mode
 - **Bit 3:** Outage
 - **Bit 4:** Not used
 - **Bit 5:** DST active

- **Bit 6:** Clock error
- **Bit 7:** External eoi
- **Bit 8:** Invalid data
- **Bit 9:** Time adjustment
- **Bit 10:** Not used
- **Bit 11:** Not used
- **Bit 12:** Not used
- **Bit 13:** Not used
- **Bit 14:** TOU enable
- **Bit 15:** Checksum error
- **Element 2 to 25:** Value of variables, in the order defined by the **Defining Parameters for Reading Values for Historic Tag**

Reading Interruptions

Read-Only

B1	Address of a meter in the network
B2	2 (two)
B3	Sequential identifier of a parameter for a device
B4	Not used, leave it in 0 (zero)

This reading occurs when the *B1* and *B3* parameter refer to a **Defining Parameters for Reading Values for Historic Tag** that is configured with the *B4* parameter equal to 500, 501, or 502.

Elements of this Block Tag

- **Timestamp:** Date and time of the registered value
- **Element 1:** Value with a bit mask containing a status
 - **Bit 0:** Start in DST
 - **Bit 1:** End in DST
 - **Bit 2:** Test mode
 - **Bit 3:** Clock error
 - **Bit 4:** Not used
 - **Bit 5:** Not used
 - **Bit 6:** Not used

- **Bit 7:** Checksum error
- **Element 2:** Date and time of the beginning of this interruption in **Date and Time** format
- **Element 3:** Date and time of the end of this interruption in **Date and Time** format
- **Element 4:** Event duration, in seconds in **Float** format
- **Element 5:** Value with a bit mask containing the reason for this interruption
 - **Bit 0:** Soft reset pin
 - **Bit 1:** Soft reset instruction
 - **Bit 2:** Loss of clock
 - **Bit 3:** Not used
 - **Bit 4:** Double bus fault
 - **Bit 5:** Software watchdog timeout
 - **Bit 6:** Power up
 - **Bit 7:** External hard reset

Reading SAG or SWELL

Read-Only

B1	Address of a meter in the network
B2	2 (two)
B3	Sequential identifier of a parameter for a device
B4	Not used, leave it in 0 (zero)

This reading occurs when the *B1* and *B3* parameters refer to a **Defining Parameters for Reading Values for Historic** Tag that is configured with the *B4* parameter equal to 300, 301, 302, 303, 304, 305, 400, 401, 402, 403, 404, or 405.

Elements of this Block Tag

- **Element 1:** Value with a bit mask containing one of the following statuses
 - **Bit 0:** Start in DST
 - **Bit 1:** End in DST
 - **Bit 2:** Test mode
 - **Bit 3:** Clock error
 - **Bit 4:** Not used
 - **Bit 5:** Not used

- **Bit 6:** Not used
- **Bit 7:** Checksum error
- **Element 2:** Date and time of the beginning of this event in **Date and Time** format
- **Element 3:** Date and time of the end of this event in **Date and Time** format
- **Element 4:** Minimum (SAG) or maximum (SWELL) value in RMS Volts in **Float** format
- **Element 5:** **averageCurrentRmsDuringEvent** value in **Float** format

Reading SAG and SWELL

Read-Only

B1	Address of a meter in the network
B2	2 (two)
B3	Sequential identifier of a parameter for a device
B4	Not used, leave it in 0 (zero)

This reading occurs when the *B1* and *B3* parameters refer to a **Defining Parameters for Reading Values for Historic** Tag is configured as a bit mask in the *B4* parameter. It returns a value identical to the **Reading SAG or SWELL** Tag, except that it contains an extra field indicating the type of performed reading (SAG or SWELL, Phases A, B, or C, levels 1 or 2).

Elements of this Block Tag

- **Element 1:** Value with a bit mask containing one of the following statuses
 - **Bit 0:** Start in DST
 - **Bit 1:** End in DST
 - **Bit 2:** Test mode
 - **Bit 3:** Clock error
 - **Bit 4:** Not used
 - **Bit 5:** Not used
 - **Bit 6:** Not used
 - **Bit 7:** Checksum error
- **Element 2:** Date and time of the beginning of this event in **Date and Time** format
- **Element 3:** Date and time of the end of this event in **Date and Time** format
- **Element 4:** Minimum (SAG) or maximum (SWELL) value in RMS Volts in **Float** format
- **Element 5:** **averageCurrentRmsDuringEvent** value in **Float** format
- **Element 6:** Identifier of an event, between 300 and 305 or between 400 and 405

Reading Logs of Events

Read-Only

B1	Address of a meter in the network
B2	3 (three)
B3	Not used, leave it in 0 (zero)
B4	Not used, leave it in 0 (zero)

This Tag is used for reading the code and description of an event.

Elements of this Block Tag

- **Timestamp:** Date and time of the record of an event
- **Element 1:** Code of an event
- **Element 2:** Description of an event in **Text** format

NOTE

To force a new reading of events, write any value to the **Enable New Reading of a Log Event in a Meter** Tag. Writing to this Tag causes the next reading of a block with event logs to retrieve data from a meter, to avoid downloading all events whenever this Block Tag is read, thus generating data duplicity.

Reading Snapshot Values in Real Time

Read-Only

B1	Address of a meter in the network
B2	4 (four)
B3	Not used, leave it in 0 (zero)
B4	Not used, leave it in 0 (zero)

This Tag is used for reading snapshot values and returns a block containing front-end values of a meter.

Elements of this Block Tag

- **Timestamp:** Date and time when values were read
- **Element 1:** Va
- **Element 2:** Vb

- **Element 3:** Vc
- **Element 4:** Ia
- **Element 5:** Ib
- **Element 6:** Ic
- **Element 7:** In
- **Element 8:** Wa
- **Element 9:** Wb
- **Element 10:** Wc
- **Element 11:** Waggregate
- **Element 12:** VARa
- **Element 13:** VARb
- **Element 14:** VARc
- **Element 15:** VARaggregate
- **Element 16:** VAa_vec
- **Element 17:** VAa_ari
- **Element 18:** VAb_vec
- **Element 19:** VAb_ari
- **Element 20:** VAc_vec
- **Element 21:** VAc_ari
- **Element 22:** VAaggregate_vec
- **Element 23:** VAaggregate_ari
- **Element 24:** meterForm
- **Element 25:** voltagePhaseStatus
- **Element 26:** averagePF_A
- **Element 27:** averagePF_B
- **Element 28:** averagePF_C
- **Element 29:** averagePF_aggregate
- **Element 30:** pf_a
- **Element 31:** pf_b
- **Element 32:** pf_c
- **Element 33:** pf_agg
- **Element 34:** pf_a_vec

- **Element 35:** pf_b_vec
- **Element 36:** pf_c_vec
- **Element 37:** pf_agg_vec
- **Element 38:** pf_a_ari
- **Element 39:** pf_b_ari
- **Element 40:** pf_c_ari
- **Element 41:** pf_agg_ari
- **Element 42:** frequency
- **Element 43:** batteryVoltage
- **Element 44:** currentTimeInDST

Reading Harmonic's Snapshot Values in Real Time

Read-Only

B1	Address of a meter in the network
B2	5 (five)
B3	Index of a phase
B4	Not used, leave it in 0 (zero)

Tag Block for reading harmonic's snapshot values. This Tag returns a block containing snapshot values of harmonics for a certain phase of a meter, defined in the *B3* parameter.

Elements of this Block Tag

- **Element 1:** Size of a register for this phase (*fileRecordSize*). This value does not change after reconfiguration and it is calculated based on the configuration parameters of a meter, the parameters from the **HARMONICS CONFIGURATION** variable
- **Element 2:** Last register written (*lastRecordWritten*). Represents the number of registers on RAM memory, and it is incremented at each interval configured in the *intervalLengthInSecs* parameter of the **HARMONICS CONFIGURATION** variable of a meter
- **Element 3:** Total harmonic distortion, in percentage, or *thdVolt_US*. The next figure shows its definition for the US market

$$THD(Voltage) = \frac{\sqrt{\sum_{n=2}^{20} V_n^2}}{V_1} \times 100$$

Total harmonic distortion (US)

- **Element 4:** Same as Element 3 (three) with current, or *thdAmps_US*
- **Element 5:** Total harmonic distortion, in percentage, or *thdVolt_EUROPE*. The next figure shows its definition for the European market

$$THD(Voltage) = \frac{\sum_{n=2}^{20} Vn^2}{\sum_{n=1}^{20} Vn^2} \times 100$$

Total harmonic distortion (EU)

- **Element 6:** Same as Element 5 (five), but for current, or *thdAmps_EUROPE*
- **Element 7:** This Element returns the value for active power on this phase, by considering only fundamental voltage and current, or *fundamentalPower*
- **Element 8:** *displacementPowerFactor*

NOTE

Elements 9 (nine), 10, 11, and 12 next are Elements of an array of values, and can repeat according to the maximum number of harmonics plus 1 (one), in which the first 4 (four) Elements correspond to DC values, that is, Elements 9 (nine), 10, 11, and 12 of this Block Tag correspond to DC values. Elements 13, 14, 15, and 16 correspond to the first harmonic, the fundamental. Elements 17, 18, 19, and 20 correspond to the second harmonic, and so on.

- **Element 9: Harmonic 0** is DC as a percentage of the fundamental, or *voltAmplitude*. **Harmonic 1** is the fundamental in Volts RMS. For all other harmonics, values are a percentage of that fundamental
- **Element 10:** Angles in degrees, or *voltAngle*
- **Element 11:** Analogous to Element 9 (nine), or *currentAmplitude*
- **Element 12:** Analogous to Element 10, or *currentAngle*
- **Element 13:** Harmonic 1 (Volts RMS), or *voltAmplitude*
- **Element 14:** Harmonic 1, or *voltAngle*

Notes

- Mass memory and interruption readings must be performed according to these steps:
 1. Define all parameters by writing to a **Defining Parameters for Reading Values for Historic** Tag.
 2. Start a reading thread by writing to a **Mass Memory's Reading Thread** Tag. Only one thread can be started each time.
 3. Check the status of this reading thread until it is finished by reading a **Mass Memory's Reading Thread** Tag.
 4. Perform a data reading by using a Block Tag with the *B2* parameter equal to 2 (two, data is only available until the next thread starts, when they are overwritten).
- To communicate via modem, it must be configured on Windows Control Panel with a baud rate of 9600 bps, and all compression and error correction options must be disabled.

- A communication test, that is, the **Communication Test** Tag, can also be performed manually, via HyperTerminal or other tool. To do so, once physically connected to a meter, via serial or modem, send the value "I" and a response must be received.
- A Q1000 meter provides the concept of access levels, and according to this concept, certain types of functionality are only available by providing a password linked to that level. Up to 4 (four) different passwords can be defined in a meter, corresponding to 4 (four) possible levels and, if this occurs, this password must be informed using an **Access Password** Tag.

Notes on Quality

Version 2.0

- **Reading Snapshot Values**
 - This reading **was not sufficiently tested** nor used in a production environment. Only bench tests were performed
- **Reading Harmonic's Snapshot Values**
 - This reading **has not yet been tested** as for the publication of this Driver

NOTE

For more information about using these readings, please contact *Eclipse Software's Support*.

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to **Q1000 (miniDLMS)** Driver.

Configuration of a Driver

I/O Interface configuration is performed on a Driver's configuration dialog box. To access the configuration of this dialog box in **Eclipse E3** in version 1.0, follow these steps:

1. Right-click a Driver object (IODriver).
2. Select the **Properties** item on the contextual menu.
3. Select the **Driver** tab.
4. Click **Other parameters**.

In **Eclipse E3** version 2.0 or later, click **Configure driver**  on a Driver's toolbar. In **Eclipse SCADA**, follow these steps:

1. Open the Organizer.
2. Select a Driver on Organizer's tree.
3. Click **Extras** on the **Driver** tab.

Currently, an I/O Interface allows opening only one connection for each Driver. This means that, if users want to access two serial ports, they must add two Drivers to an application and then configure each one of these Drivers for each serial port.

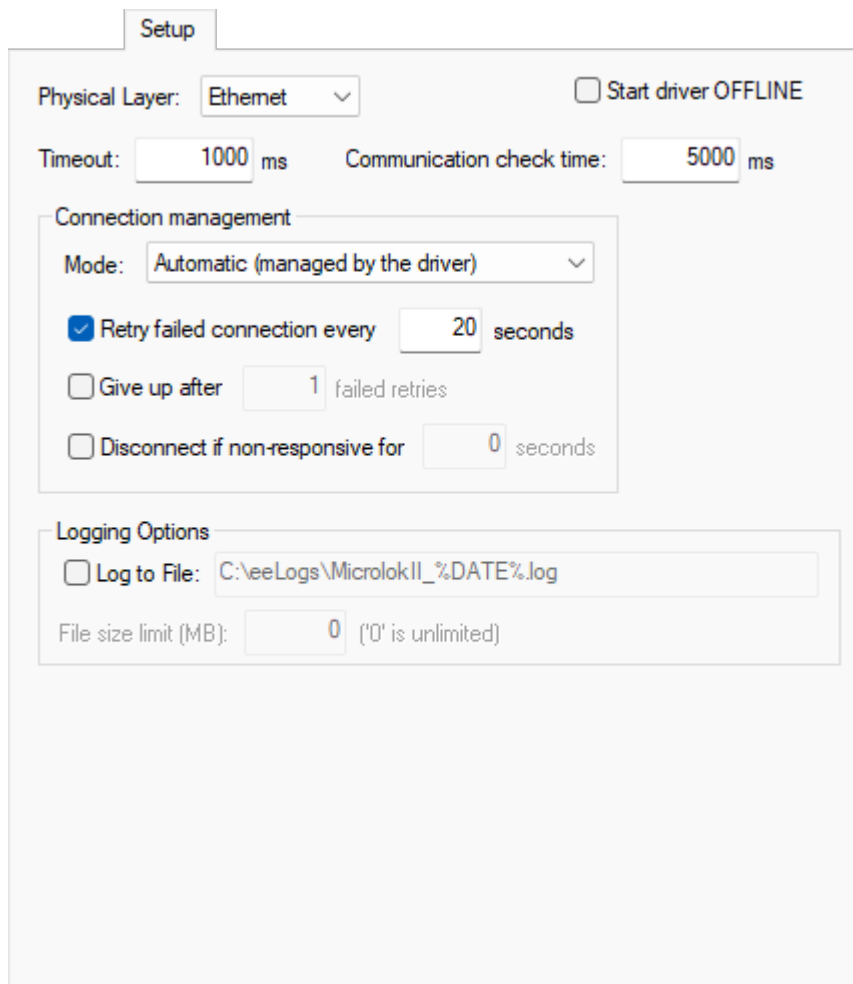
Configuration Dialog Box

The dialog box of I/O Interfaces allows configuring the I/O connection used by a Driver. This dialog box contains the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, described on the next topics. If a Driver does not implement a specific I/O connection, its corresponding tab is not available for configuration. Some Drivers may contain additional tabs, specific for that Driver, on the configuration dialog box.

Setup Tab

The **Setup** tab contains general configurations of a Driver. This tab is divided into the following groups:

- **General configurations:** Configurations of a Driver's physical layer, time-out, and initialization mode
- **Connection management:** Configurations on how the I/O Interface keeps a connection and which recovery policy is used on failure
- **Logging options:** Controls the generation of log files



The screenshot shows the 'Setup' tab of a configuration dialog box. It is divided into three main sections:

- Physical Layer:** A dropdown menu is set to 'Ethernet'. To the right is an unchecked checkbox labeled 'Start driver OFFLINE'.
- Timeouts:** Two input fields: 'Timeout:' with '1000' and 'ms', and 'Communication check time:' with '5000' and 'ms'.
- Connection management:** A dropdown menu is set to 'Automatic (managed by the driver)'. Below it are three options:
 - 'Retry failed connection every' with '20' and 'seconds'.
 - 'Give up after' with '1' and 'failed retries'.
 - 'Disconnect if non-responsive for' with '0' and 'seconds'.
- Logging Options:**
 - 'Log to File:' with the path 'C:\eeLogs\MicrolokII_%DATE%.log'.
 - 'File size limit (MB):' with '0' and '(0 is unlimited)'.

Setup tab

General options on the Setup tab

OPTION	DESCRIPTION
Physical Layer	Select the physical layer on a list. Available options are Serial , Ethernet , Modem , and RAS . The selected interface must be configured on its specific tab
Timeout	Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer

OPTION	DESCRIPTION
Communication check time	Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the IO.CommunicationStatus Tag
Start driver OFFLINE	Select this option so that a Driver starts in Offline mode or stopped. This means that the I/O interface is not created until this Driver is configured to Online mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time

Options on the Connection management group

OPTION	DESCRIPTION
Mode	Selects a management mode of a connection. Selecting the Automatic option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the Manual option allows an application to fully manage a connection
Retry failed connection every ... seconds	Select this option to enable a Driver's connection retry in a certain interval, in seconds. If the Give up after failed retries option is not selected, this Driver keeps retrying until a connection is performed, or until the application is stopped
Give up after ... failed retries	Enable this option to define a maximum number of connection retries. When the specified number of consecutive connection retries is reached, a Driver goes to the Offline mode, assuming that a hardware problem was detected. If a Driver establishes a successful connection, the number of unsuccessful retries is cleared. If this new connection is lost, then the retry counter starts at zero
Disconnect if non-responsive for ... seconds	Enable this option to force a Driver to disconnect if no byte was received by the I/O interface during the specified time-out, in seconds. This time-out must be greater than the time-out configured in the Timeout option

Options on the Logging Options group

OPTION	DESCRIPTION
<p>Log to File</p>	<p>Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes. If the %PROCESS% macro is used in the log file name, it is replaced by the identifier of the current process. This option is particularly useful when using several instances of the same Driver in Elipse E3, thus allowing each instance to generate a separate log file. For example, when configuring this option with value "c:\e3logs\drivers\sim_%PROCESS%.log", it generates a file named c:\e3logs\drivers\sim_00000FDA.log for process OFDAh. Users can also use the %DATE% macro in the file name. In this case a log file is generated every day, in the format aaaa_mm_dd. For example, when configuring this option with value "c:\e3logs\drivers\sim_%DATE%.log", it generates a file named c:\e3logs\drivers\sim_2005_12_31.log in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log in 01/01/2006. Similarly, the %DATE_HOUR% macro generates one log file per hour, in the format aaaa_mm_dd_hh</p>
<p>File size limit (MB)</p>	<p>Configure the log file size limit, in megabytes. A value equal to 0 (zero) means that there is no size limit for the log file</p>

Serial Tab

Use this tab to configure parameters for a **Serial** Interface.

Serial

Port:

Baud rate:

Data bits:

Parity:

Stop bits:

Enable 'ECHO' suppression

Handshaking

DTR control:

RTS control:

Wait for CTS before send

CTS timeout: ms

Delay before send: ms

Delay after send: ms

Inter-byte delay (microseconds): μ s

Inter-frame delay (milliseconds): ms

Serial tab

General options on the Serial tab

OPTION	DESCRIPTION
Port	Select a serial port on the list, from COM1 to COM4 , or type the name of a serial port in the format COMn , such as "COM15". When typing the name of a serial port manually, the dialog box only accepts names of serial ports starting with the expression "COM"
Baud rate	Select a baud rate on the list (1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200) or type a baud rate, such as 600
Data bits	Select 7 (seven) or 8 (eight) data bits on the list
Parity	Select a parity on the list. The available options are None, Even, Odd, Mark, or List
Stop bits	Select the number of stop bits on the list. The available options are 1, 1.5, or 2 stop bits
Enable 'ECHO' suppression	Enable this option to remove the echo received after the I/O Interface sends data via serial port. If this echo is not equal to the bytes just sent, then the I/O Interface aborts communication
Inter-byte delay (microseconds)	Defines a delay between each byte transmitted by the I/O Interface, in millionths of a second, that is, 1000000 is equal to a second. This option must be used with small delays of less than a millisecond
Inter-frame delay (milliseconds)	Defines a delay between packets sent or received by the I/O Interface, in thousandths of a second, that is, 1000 is equal to a second. This delay is applied if the I/O Interface

OPTION	DESCRIPTION
	sends two consecutive packets, or between a received packet and the next sending

The **Handshaking** group configures the usage of **RTS**, **CTS**, and **DTR** signals in the handshaking process, that is, it controls when data can be sent or received via serial line. Most of the time, configuring the **DTR control** option to **ON** and the **RTS control** option to **Toggle** works with **RS232**-type serial lines as well as with **RS485**-type serial lines.

Available options on the Handshaking group

OPTION	DESCRIPTION
DTR control	Select the value ON to keep the DTR signal always on while the serial port is open. Select the value OFF to turn the DTR signal off while the serial port is open. Some devices require the DTR signal always on to allow communication
RTS control	Select the value ON to keep the RTS signal always on while the serial port is open. Select the value OFF to turn the RTS signal off while the serial port is open. Select the value Toggle to turn the RTS signal on while sending bytes via serial port and turn it off when not sending bytes, therefore enabling the reception
Wait for CTS before send	Available only when the RTS control option is configured with the value Toggle . Use this option to force a Driver to check the CTS signal before sending bytes via serial port, after turning the RTS signal on. In this mode, the CTS signal is handled as a permission flag for sending
CTS timeout	Determines a maximum time, in milliseconds, that a Driver waits for the CTS signal after turning the RTS signal on. If the CTS signal is not turned on within this time-out, that Driver then fails the current communication and returns an error
Delay before send	Some serial port devices have a delay when enabling a data sending circuit after the RTS signal is turned on. Configure this option to wait a certain number of milliseconds after turning the RTS signal on and before sending the first byte. IMPORTANT : This delay must be used carefully, because it uses 100% of CPU resources while waiting. System's general performance degrades as this value increases
Delay after send	This is the same effect of the Delay before send option, but in this case the delay is performed after sending the last byte, before turning the RTS signal off

Ethernet Tab

Use this tab to configure parameters of an **Ethernet** Interface. These parameters, except port configurations, must also be configured for use in the **RAS** Interface.

Ethernet

Transport: TCP/IP ▼

PING before connecting
 Timeout: 4000 ms
 Retries: 1

Listen for connections on port: 0

Share listen port with other processes

Interface: (All Interfaces) ▼

Use IPv6 Use SSL SSL Settings

Enable 'ECHO' suppression

IP Filter:

Connect to

<input type="checkbox"/> Main IP:		Port:	502	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 1:		Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 2:		Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 3:		Port:	0	<input type="checkbox"/> Local port:	0

Ethernet tab

Available options on the Ethernet tab

OPTION	DESCRIPTION
Transport	Select the value TCP/IP for a TCP socket (<i>stream</i>) or select the value UDP/IP to use a UDP socket (<i>connectionless datagram</i>)
Listen for connections on port	Use this option to wait for new connections in a specific IP port, common in Slave Drivers. If this option remains unselected, a Driver connects to the address and port specified in the Connect to option
Share listen port with other processes	Select this option to share the listening port with other Drivers and processes
Interface	Select the local network interface, identified by its IP address, that a Driver uses to establish and receive connections, or select the value (All Interfaces) to allow connection in any network interface
Use IPv6	Select this option to force a Driver to use addresses in IPv6 format on all Ethernet connections. Leave this option deselected to use the IPv4 format
Enable 'ECHO' suppression	Enable this option to remove the echo from received data. An echo is a copy of sent data, which can be returned before a reply message
IP Filter	List of restricted or allowed IP addresses from where a Driver accepts connections (<i>Firewall</i>). Please check the IO.Ethernet.IPFilter property for more information
PING before connecting	Enable this option to execute a ping command, that is, to check whether a device can be reached on a network, for a device before trying a socket connection. This is a quick way

OPTION	DESCRIPTION
	<p>of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are:</p> <ul style="list-style-type: none"> • Timeout: Specify the number of milliseconds to wait for a reply from a ping command. Users must use a ping command to check the normal reply time, configuring this option for a value above that average. Usually this value can be configured between 1000 and 4000 milliseconds, that is, between 1 (one) and 4 (four) seconds • Retries: Number of retries of a ping command, not counting the first attempt. If all attempts fail, then the socket connection is aborted

Available options on the Connect to group

OPTION	DESCRIPTION
Main IP	Type the IP address of a remote device. Users can use an IP address separated by dots, as well as a URL. In case of a URL, a Driver uses the available DNS service to map that URL to an IP address, such as "192.168.0.13" or "Server1"
Port	Type the IP port of a remote device, between 0 (zero) and 65535
Local port	Select this option to use a fixed local IP port when connecting to a remote device
Backup IP 1, 2, and 3	Indicate the IP address, the IP port, and the fixed local IP port of up to 3 (three) backup addresses of a remote device

Modem Tab

Use this tab to configure parameters of a **Modem** Interface. Some options on the **Serial** tab affect the configuration of a modem, therefore users must also configure the **Serial** Interface.

Modem

Select the modem to use:

▼ Modem settings...

Dial Number:

Accept incoming calls

Modem tab

The **Modem** Interface uses the TAPI modems installed on the computer.

Available options on the Modem tab

OPTION	DESCRIPTION
Select the modem to use	Select a modem on the list of available modems on the computer. If the value Default modem is selected, then the first available modem is used. Selecting this option is recommended specially when an application is used on another computer
Modem settings	Click to open the configuration window of the selected modem
Dial Number	Type a default number for dialing. This value can be changed at run time. Users can use the w character to represent a pause or a waiting time for a dial tone. For example, "0w33313456" dials the number 0 (zero), waits, and then dials the number "33313456"
Accept incoming calls	Enable this option so that a Driver answers the phone when receiving an external call. To use this option, users must configure the Connection management option on the Setup tab to the value Manual

RAS Tab

Use this tab configure parameters of a **RAS** Interface. Users must also configure the **Ethernet** tab.

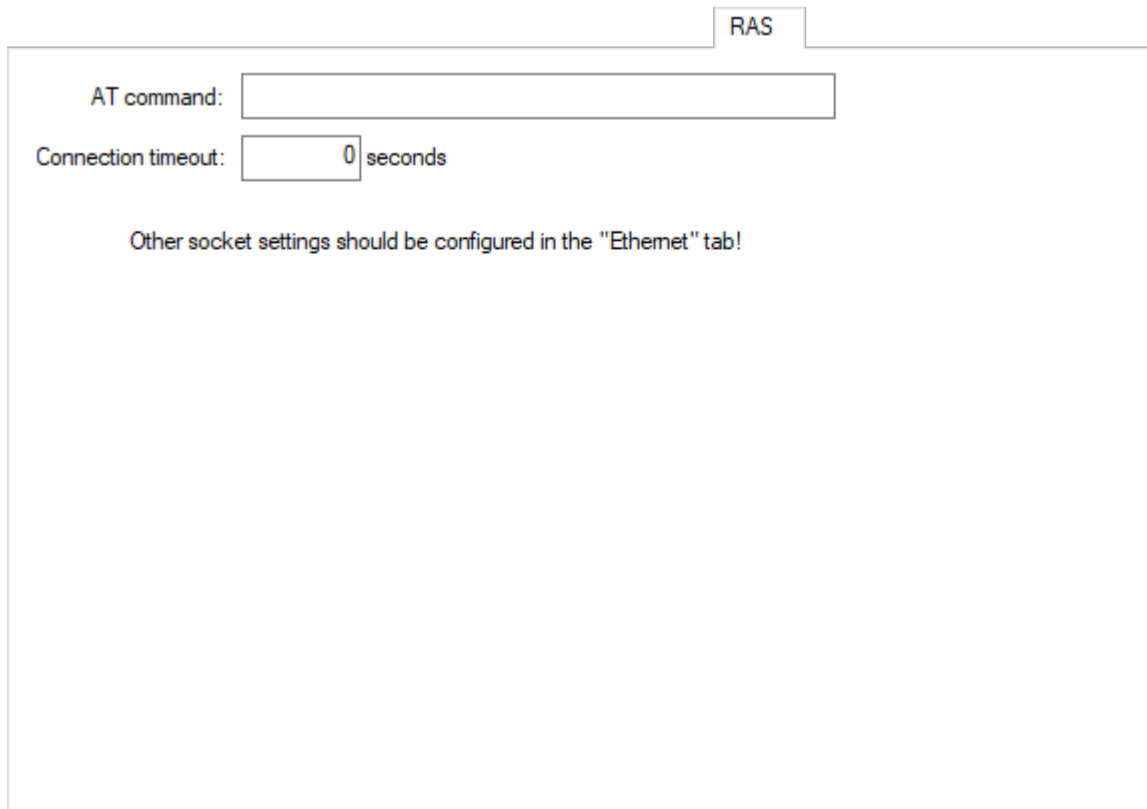
A **RAS** Interface opens a socket connection with a RAS device. A RAS device is a server of modems available through TCP/IP, waiting for socket connections on an IP port. For each connection accepted on this port, users have access to one modem.

When connecting to a RAS device, first the I/O Interface **IOKit** connects to the socket on the IP address and port configured on the **Ethernet** tab. After opening the socket, the following initialization or connection steps are performed:

1. Clears the socket, that is, removes any **TELNET** greeting message received from a RAS device.
2. Sends an **AT** dial message, in **ASCII** format, in the socket.
3. Waits for a **CONNECT** reply.
4. If the time-out expires, the connection is aborted.
5. If the **CONNECT** reply is received within the time-out, the socket is available for communication with a device, that is, the connection was established.

If step 5 (five) is successful, then the socket behaves as a normal socket, with the RAS device working as a router between a Driver and the device. Bytes sent by a Driver are received by the RAS device and sent to the destination device using a modem. Bytes received by the modem's RAS device are sent back to a Driver using the same socket.

After establishing a connection, the **RAS** interface monitors data received by a Driver. If a "NO CARRIER" **String** is found, the socket is closed. If the RAS device does not send a **NO CARRIER** signal, the **RAS** Interface cannot detect when the modem connection between the RAS device and the final I/O device fails. To recover from this failure, users are strongly advised to enable the **Disconnect if non-responsive** option on the **Setup** tab.



RAS tab

Available options on RAS tab

OPTION	DESCRIPTION
AT command	A String with the full AT command used to dial to a destination device. For example, "ATDT33313456" dials by tone to number "33313456"
Connection timeout	Number of seconds to wait for a modem's CONNECT reply, after sending an AT command

General Configurations

This section contains information about the configuration of general **I/O Tags** and **Properties** of I/O Interfaces.

I/O Tags

General I/O Interfaces Tags (N2/B2 = 0)

The Tags described next are provided for all supported I/O Interfaces.

IO.CommunicationStatus

Type of Tag	I/O Tag
Type of Access	Reading
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	6 (six)
String Configuration	IO.CommunicationStatus

This Tag informs the communication status of a Driver. It indicates how communication works relative to receiving valid data within a time period arbitrated in the configuration. For more information, please check topic **Setup Tab**. Possible values are **0 - Inactive communication**: The Driver did not receive valid data or stopped receiving data after n milliseconds, as configured in the properties window, or **1 - Active communication**: The Driver is receiving valid data.

IO.IOKitEvent

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	1 (one)
Size Property	4 (four)
ParamItem Property	IO.IOKitEvent

This Block returns Driver events generated by several sources in I/O Interfaces. The **TimeStamp** property of this Block represents the moment this event occurred. The Block Elements are the following:

- **Element 0**: Type of event. Possible values are **0**: Information, **1**: Warning, or **2**: Error
- **Element 1**: Source of an event. Possible values are **0**: Driver (specific of a Driver), **-1**: IOKit (generic events of I/O Interfaces), **-2**: **Serial** Interface, **-3**: **Modem** Interface, **-4**: **Ethernet** Interface, or **-5**: **RAS** Interface
- **Element 2**: Error number, specific for each source of event

- **Element 3:** Message of an event, a **String** specific for each event

NOTE

A Driver keeps a maximum number of 100 events internally. If additional events are reported, older events are discarded.

IO.PhysicalLayerStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	2 (two)
String Configuration	IO.PhysicalLayerStatus

This Tag indicates the status of a physical layer. Possible values are the following:

- **0:** Physical layer stopped, that is, a Driver is in **Offline** mode, the physical layer failed when initializing, or exceeded the maximum number of reconnection attempts
- **1:** Physical layer started but not connected, that is, a Driver is in **Online** mode but the physical layer is not connected. If the **Connection management** option is configured with the value **Automatic**, the physical layer can be connecting, disconnecting, or waiting for a reconnection attempt. If the **Connection management** option is configured with the value **Manual**, then the physical layer remains in this status until forced to connect
- **2:** Physical layer connected, that is, the physical layer is ready for use. This **DOES NOT** mean a device is connected, only that the access layer is working

IO.SetConfigurationParameters

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	3 (three)
Size Property	2 (two)
ParamItem Property	IO.SetConfigurationParameters

Use this Tag to change any property of a Driver's configuration dialog box at run time.

This Tag works only while a Driver is in **Offline** mode. To start a Driver in **Offline** mode, select the **Start driver OFFLINE** option on that Driver's configuration dialog box. Users can write to a PLC Tag or to a Block Tag containing the parameters to change. Writing individual Block Elements is not supported, the whole Block must be written at once.

In **Eclipse SCADA**, users must use a Block Tag. Every parameter to configure uses two Block Elements. For example, if users want to configure 3 (three) parameters, then the size of the Block must be 6 (six, 3×2). The first Element is the property's name, as a **String**, and the second Element is the property's value, according to the next example.

```
// 'Block' must be a Block Tag with automatic reading,
// scan reading, and automatic writings disabled.
// Configure all parameters
Block.element001 = "IO.Type" // Parameter 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parameter 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parameter 3
Block.element006 = 19200
// Writes the whole Block
Block.Write()
```

When using **Eclipse E3**, the ability to create arrays at run time allows using an I/O Tag as well as a Block Tag. Users can use the **Write** method of a Driver to send the parameters directly to that Driver, without creating a Tag, according to the next example.

```
Dim arr(6)
' Configure all array elements
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' There are two methods to send parameters
' Method 1: Using an I/O Tag
tag.WriteEx arr
' Method 2: Without using a Tag
Driver.Write -1, 0, 0, 3, arr
```

A variation of the previous example uses a bidimensional array.

```
Dim arr(10)
' Configure all array elements. Notice the array was resized
' to 10 elements. Empty array elements are ignored by a Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

A Driver does not validate parameter names or passed values, therefore be careful when writing parameters and values. The **Write** method fails if the configuration array is incorrectly created. Users can check the log of a Driver or use the *writeStatus* parameter of the **WriteEx** method to find out the exact cause of an error.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , , strError Then
    MsgBox "Failed configuring Driver parameters: " + strError
End If
```

IO.WorkOnline

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	4 (four)
String Configuration	IO.WorkOnline

This Tag informs the current status of a Driver and allows starting or stopping the physical layer. Possible values are the following:

- **0 - Driver Offline:** Physical layer is closed or stopped. This mode allows a dynamic configuration of a Driver's parameters using the **IO.SetConfigurationParameters** Tag
- **1 - Driver Online:** Physical layer is open or executing. While in **Online** mode, the physical layer can be connected or disconnected and its current status can be checked using the **IO.PhysicalLayerStatus** Tag

In the next example, using **Eclipse E3**, a Driver is configured to **Offline** mode, its COM port is changed, and then configured to **Online** mode again.

```
'Configure to Offline mode
Driver.Write -1, 0, 0, 4, 0
'Change port to COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configure to Online mode
Driver.Write -1, 0, 0, 4, 1
```

The **Write** method may fail when configuring a Driver to **Online** mode, that is, writing the value 1 (one). In this case, this Driver remains in **Offline** mode. The cause of failure can be:

- Type of physical layer incorrectly configured, probably an invalid value was configured in the **IO.Type** property
- This Driver may have run out of memory
- Physical layer probably did not create its working thread. Search the log file for a message "Failed to create physical layer thread!"
- Physical layer could not start. The cause of this failure depends on the type of physical layer. It can be an invalid serial port number, a failure when starting Windows Sockets, or a failure when starting TAPI (modem), among others. This cause is recorded on the log file

IMPORTANT

Even if the configuration of a Driver to **Online** mode is successful, this does not necessarily mean the physical layer is ready to use, that is, ready to execute input and output operations with an external device. The **IO.PhysicalLayerStatus** Tag must be checked to ensure the physical layer is connected and ready for communication.

Properties

These are general properties of all supported I/O Interfaces.

IO.ConnectionMode

9 Controls the management mode of a Connection. Possible values are **0**: Automatic mode, in which a Driver manages the connection or **1**: Manual mode, in which an application manages the connection.

IO.GiveUpEnable

☑ When configured to True, defines a maximum number of reconnection attempts. If all reconnection attempts fail, a Driver enters the **Offline** mode. When configured to False, a Driver tries until a reconnection is successful.

IO.GiveUpTries

9 Number of reconnection attempts before this one is aborted. For example, if the value of this property is equal to 1 (one), a Driver tries only one reconnection when the connection is lost. If this one fails, this Driver enters the **Offline** mode.

IO.InactivityEnable

☑ Configure to True to enable and to False to disable inactivity detection. The physical layer is disconnected if inactive for a certain period of time. The physical layer is considered inactive only if it is capable of sending data but not capable of receiving it back.

IO.InactivityPeriodSec

9 Number of seconds to check for inactivity. If the physical layer is inactive for this period of time, it is then disconnected.

IO.RecoverEnable

☑ Configure to True to enable a Driver to recover lost connections and to False to leave a Driver in **Offline** mode when a connection is lost.

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

The first reconnection is executed immediately after a connection is lost.

IO.StartOffline

☑ Configure to True to start a Driver in **Offline** mode and to False to start a Driver in **Online** mode.


NOTE

It is pointless to change this property at run time, as it can only be changed when a Driver is already in **Offline** mode. To configure a Driver in **Online** mode at run time, write the value 1 (one) to the **IO.WorkOnline** Tag.

IO.TimeoutMs

9 Defines a time-out for the physical layer, in milliseconds. One second is equal to 1000 milliseconds.

IO.Type

 Defines the type of physical interface used by a Driver. Possible values are the following:

- **N or None:** Does not use a physical interface, that is, a Driver must provide a customized interface
- **S or Serial:** Uses a local serial port (COM n)
- **M or Modem:** Uses a local modem, internal or external, accessed via TAPI (*Telephony Application Programming Interface*)
- **E or Ethernet:** Uses a TCP/IP or UDP/IP socket
- **R or RAS:** Uses a **RAS** (*Remote Access Server*) Interface. A Driver connects to a RAS device using the **Ethernet** Interface and then sends an **AT** (*dial*) command

Statistical Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of I/O Interfaces statistics.

I/O Tags

Tags of I/O Interface Statistics (N2/B2 = 0)

The Tags described next display statistics for all I/O Interfaces.

IO.Stats.Partial.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1101
Configuration by String	IO.Stats.Partial.BytesRecv

This Tag returns the number of bytes received in the current connection.

IO.Stats.Partial.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1100
Configuration by String	IO.Stats.Partial.BytesSent

This Tag returns the number of bytes sent through the current connection.

IO.Stats.Partial.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1102
Configuration by String	IO.Stats.Partial.TimeConnectedSeconds

This Tag returns the number of seconds a Driver is connected in the current connection or 0 (zero) if a Driver is disconnected.

IO.Stats.Partial.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1103
Configuration by String	IO.Stats.Partial.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver is disconnected since the last connection ended or 0 (zero) if a Driver is connected.

IO.Stats.Total.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1001
Configuration by String	IO.Stats.Total.BytesRecv

This Tag returns the number of bytes received since a Driver was loaded.

IO.Stats.Total.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1000
Configuration by String	IO.Stats.Total.BytesSent

This Tag returns the number of bytes sent since a Driver was loaded.

IO.Stats.Total.ConnectionCount

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1004
Configuration by String	IO.Stats.Total.ConnectionCount

This Tag returns the number of connections a Driver already established, successfully, since it was loaded.

IO.Stats.Total.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1002
Configuration by String	IO.Stats.Total.TimeConnectedSeconds

This Tag returns the number of seconds a Driver remained connected since it was loaded.

IO.Stats.Total.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1003
Configuration by String	IO.Stats.Total.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver remained disconnected since it was loaded.

Properties

Currently, there are no properties defined specifically to display I/O Interface statistics at run time.

Ethernet Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of an **Ethernet** Interface.

I/O Tags

Tags of an Ethernet Interface (N2/B2 = 4)

The Tags described next allow controlling and identifying an **Ethernet** Interface at run time and they are also valid when the **RAS** Interface is selected.

IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

IO.Ethernet.IPSelect

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	0 (zero)
String Configuration	IO.Ethernet.IPSelect

Indicates the active IP address. Possible values are **0**: The main IP address is selected, **1**: The first alternative or backup IP address is selected, **2**: The second alternative or backup IP address is selected, or **3**: The third alternative or backup IP address is selected.

If the **Ethernet** or **RAS** Interface is connected, this Tag indicates which one of the four configured IP addresses is in use. If the Interface is disconnected, this Tag indicates which IP address is used first on the next attempt to connect.

During the connection process, if the active IP address is not available, the I/O Interface tries to connect using the other IP address. If the connection with the alternative IP address works, it is configured as the active IP address (automatic switchover).

To force a manual switchover, write values from 0 (zero) to 3 (three) to this Tag. This forces a reconnection with the specified IP address (**0**: Main address or **1, 2, 3**: Alternative address) if a Driver is currently connected. If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.IPSwitch

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	1 (one)
String Configuration	IO.Ethernet.IPSwitch

Any value written to this Tag forces a manual switchover. If the main IP address is active, then the first alternative or backup IP address is activated, and so on for all alternative IP addresses and returning to the main address until a connection is established.

If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.SocketState

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	2 (two)
String Configuration	IO.Ethernet.SocketState

The Value property of this Tag corresponds to socket states as a map of bits:

- **Bit 0**: 0 (zero, not listening) or 1 (one, listening)
- **Bit 1**: 0 (zero, disconnected) or 1 (one, connected)

Properties

These properties control the configuration of an **Ethernet** Interface.

NOTE

The **Ethernet** Interface is also used by the **RAS** Interface.

IO.Ethernet.AcceptConnection

☑ Configure to False if a Driver must not accept external connections, that is, if a Driver behaves as a master, or configure to True to enable the reception of connections, that is, if a Driver behaves as a slave.

IO.Ethernet.BackupEnable[2,3]

☑ Configure to True to enable an alternative or backup IP address. If the reconnection attempt with the main IP address fails, a Driver tries to use an alternative IP address. Configure to False to disable its usage.

IO.Ethernet.BackupIP[2,3]

📌 Alternative or backup IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.BackupLocalPort[2,3]

📌 Local port number to be used when connecting to an alternative IP address of a remote device. Used only if **IO.Ethernet.BackupLocalPortEnable** is equal to True.

IO.Ethernet.BackupLocalPortEnable[2,3]

☑ Configure to True to force the use of a specific local port when connecting to an alternative or backup IP address or configure to False to use any available local port.

IO.Ethernet.BackupPort[2,3]

📌 Port number of an alternative or backup IP address of a remote device, used with the **IO.Ethernet.BackupIP** property.

IO.Ethernet.IPFilter

📌 List with a comma-separated IPv4 or IPv6 addresses, which defines from which addresses a Driver accepts or blocks connections. Users can use asterisks, such as "192.168.*.*", or intervals, such as "192.168.0.41-50", in any part of IP addresses. To block an IP address or a range of IP addresses, use the tilde ("~") character at the beginning of the address, according to the next examples:

- **192.168.0.24**: Accepts only connections from IPv4 address 192.168.0.24
- **192.168.0.41-50**: Accepts connections from IPv4 addresses in the interval between 192.168.0.41 and 192.168.0.50
- **192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255
- **fe80:3bf:877:::* (expands to fe80:03bf:0877:0000:0000:0000:0000:0000:*)**: Accepts connections from IPv6 addresses in the interval between fe80:03bf:0877:0000:0000:0000:0000:0000 and fe80:03bf:0877:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Accepts connections from IPv4 addresses 192.168.0.10, 192.168.0.15, and 192.168.0.20
- **~192.168.0.95, 192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255, except the IPv4 address 192.168.0.95

When a Driver receives a connection attempt, the list of filters is scanned sequentially from left to right, searching for a specific authorization or block for the IP address where the connection comes from. If no element on the list corresponds to the IP address, the authorization or block are dictated by the last element of that list:

- If the last element on the list is an authorization, such as "192.168.0.24", then all IP addresses not found on the list are blocked
- If the last element on the list is a block, such as "~192.168.0.24", then all IP addresses not found on the list are authorized

If an IP address appears on more than one filter on the list, the leftmost filter has precedence. For example, in case of "~192.168.0.95, 192.168.0.*", the IP address 192.168.0.95 fits both rules, but the rule that wins is the leftmost one, "~192.168.0.95", and therefore this IP address is blocked.

When **IOKit** blocks a connection, it logs a message "Blocked incoming socket connection from {IP}!".

In case of UDP connections in broadcast listening mode, in which a Driver can receive packets from different IP addresses, blocks or permissions are performed at each packet received. If a packet is received from a blocked IP address, it logs a message "Blocked incoming packet from {IP} (discarding {N} bytes)!".

IO.Ethernet.ListenIP

A IP address of the local network interface that a Driver uses to establish and accept connections. Leave this property empty to establish and accepts connections using any local network interface.

IO.Ethernet.ListenPort

9 Number of the IP port used by a Driver to listen to connections.

IO.Ethernet.MainIP

A IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.MainLocalPort

9 Local port number to use when connecting to the main IP address of a remote device. This value is only used if the **IO.Ethernet.MainLocalPortEnable** property is equal to True.

IO.Ethernet.MainLocalPortEnable

Configure to True to force the use of a specific local port when connecting to the main IP address of a remote device or configure to False to use any available local port.

IO.Ethernet.MainPort

9 Number of the IP port of a remote device, used with the **IO.Ethernet.MainIP** property.

IO.Ethernet.PingEnable

Configure to True to enable sending a **ping** command to the IP address of a remote device, before trying to connect to the socket. This socket's connection time-out cannot be controlled, therefore sending a **ping** command before connecting is a fast way to detect if the connection is going to fail. Configure to False to disable a **ping** command.

IO.Ethernet.PingTimeoutMs

9 Delay time to wait for a response from a **ping** command, in milliseconds.

IO.Ethernet.PingTries

9 Maximum number of attempts of a **ping** command. Minimum value is 1 (one), including the first **ping** command.

IO.Ethernet.ShareListenPort

☑ Configure to True to share a listening port with other Drivers and processes or False to open a listening port in exclusive mode. To successfully share a listening port, all Drivers and processes that use that port must open it in shared mode. When a listening port is shared, each incoming connection is distributed to one of the processes listening. This way, if a Slave Driver only supports one connection at a time, users can use several instances of this Driver listening on the same port, therefore simulating a Driver with support for multiple connections.

IO.Ethernet.SupressEcho

☑ Configure to True to eliminate echoes in communication. An echo is the unwanted reception of an exact copy of all data packets a Driver sent to a device.

IO.Ethernet.Transport

A Defines a transport protocol. Possible values are **T or TCP**: Uses the TCP/IP protocol or **U or UDP**: Uses the UDP/IP protocol.

IO.Ethernet.UseIPv6

☑ Configure to True to use IPv6 addresses on all Ethernet connections or configure to False to use IPv4 addresses (default).

Modem Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Modem** (TAPI) Interface.

I/O Tags

Tags of a Modem Interface (N2/B2 = 3)

The Tags described next allow controlling and diagnosing a **Modem** (TAPI) Interface at run time.

IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

IO.TAPI.ConnectionBaudRate

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	5 (five)
String Configuration	IO.TAPI.ConnectionBaudRate

Indicates a baud rate value for the current connection. If a modem is not connected, returns the value 0 (zero).

IO.TAPI.Dial

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	1 (one)
String Configuration	IO.TAPI.Dial

Write any value to this Tag to force a **Modem** Interface to start a call. This is an asynchronous command, only starting the call process. Users can monitor the **IO.TAPI.IsModemConnected** Tag to detect when a call is established.

IO.TAPI.HangUp

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	4 (four)
String Configuration	IO.TAPI.HangUp

Any value written to this Tag hangs the current call up.

NOTE

Use this command only when managing the physical layer manually or when explicitly trying to force a Driver to restart the communication. If the physical layer is configured for automatic reconnection, a Driver immediately tries to reestablish the connection.

IO.TAPI.IsModemConnected

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	3 (three)
String Configuration	IO.TAPI.IsModemConnected

This Tag indicates the status of a modem connection. Possible values are **0**: The modem is not connected, but it may be performing or receiving an external call or **1**: The modem is connected and a Driver completed or received an external call successfully. While it is in this status, the physical layer can send or receive data.

IO.TAPI.IsModemConnecting

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	6 (six)
String Configuration	IO.TAPI.IsModemConnecting

This Tag indicates the status of a modem connection, with more details than the **IO.TAPI.IsModemConnected** Tag. Possible values are **0**: Modem is not connected, **1**: Modem is connecting, that is, performing or receiving an external call, **2**: Modem is connected. While in this status, the physical layer can send or receive data, or **3**: Modem is disconnecting the current call.

IO.TAPI.ModemStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	2 (two)
String Configuration	IO.TAPI.ModemStatus

Returns a **String** with the current status of a modem. Possible values are the following:

- **"No status!"**: The **Modem** Interface was not open yet or was already closed
- **"Modem initialized OK!"**: The **Modem** Interface was initialized successfully
- **"Modem error at initialization!"**: A Driver could not initialize modem's line. Check that Driver's log file for more details
- **"Modem error at dial!"**: A Driver could not start or accept a call
- **"Connecting..."**: A Driver started a call successfully, and is currently processing that call
- **"Ringing..."**: Indicates that the modem is receiving an external call, but it did not accepted it yet
- **"Connected!"**: A Driver connected successfully, that is, completed or accepted an external call
- **"Disconnecting..."**: A Driver is turning the current call off
- **"Disconnected OK!"**: A Driver turned the current call off
- **"Error: no dial tone!"**: A Driver aborted a call because the available line signal was not detected
- **"Error: busy!"**: A Driver aborted a call because the line was busy
- **"Error: no answer!"**: A Driver aborted a call because no answer was received from the other modem
- **"Error: unknown!"**: Current call was aborted because of an unknown error

IO.TAPI.PhoneNumber

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	0 (zero)
String Configuration	IO.TAPI.PhoneNumber

This Tag is a **String** that reads or changes the telephone number used by the **IO.TAPI.Dial** Tag. When changing this Tag, the new value is used only on the next **Dial** command.

Properties

These properties control the configuration of a **Modem** (TAPI) Interface.

IO.TAPI.AcceptIncoming

9 Configure to False if a modem cannot accept external calls, that is, if a Driver behaves as a master, and configure to True to enable receiving calls, that is, if a Driver behaves as a slave.

IO.TAPI.ModemID

9 This is the identification number of a modem. This ID is created by Windows and used internally to identify a modem on a list of devices installed on a computer. This ID may not remain valid if a modem is reinstalled or an application is executed on another computer.

NOTE

It is advisable to configure this property as 0 (zero), indicating that a Driver must use the first available modem.

IO.TAPI.PhoneNumber

A A telephone number used by **Dial** commands, such as "0w01234566", in which the "w" character forces a modem to wait for a call sign.

RAS Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **RAS** Interface.

I/O Tags

Tags of a RAS Interface (N2/B2 = 5)

Currently, there are no Tags defined specifically to manage a **RAS** Interface at run time.

Properties

These properties control the configuration of a **RAS** Interface.

NOTE

A **RAS** Interface uses the **Ethernet** Interface, which therefore must be also configured.

IO.RAS.ATCommand

A An **AT** command to send through a socket to force a RAS device to perform a call using the current RAS channel, such as "ATDT6265545".

IO.RAS.CommandTimeoutSec

9 Time to wait for a **CONNECT** message in response to an **AT** command, in seconds.

Serial Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Serial** Interface.

I/O Tags

Tags of a Serial Interface (N2/B2 = 2)

Currently, there are no Tags defined specifically to manage a **Serial** Interface at run time.

Properties

These properties control the configuration of a **Serial** Interface.

IO.Serial.Baudrate

9 Specifies a baud rate of a serial port, such as 9600.

IO.Serial.CTSTimeoutMs

9 Time to wait for a **CTS** signal, in milliseconds. After turning the **RTS** signal on, a timer is started to wait for a **CTS** signal. If this timer expires, a Driver aborts sending bytes through the serial port. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to True.

IO.Serial.DataBits

9 Specifies the number of data bits to configure a serial port. Possible values are **5**: Five data bits, **6**: Six data bits, **7**: Seven data bits, or **8**: Eight data bits.

IO.Serial.DelayAfterMs

9 Number of milliseconds to delay after the last byte is sent through a serial port, but before turning the **RTS** signal off. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DelayBeforeMs

9 Number of milliseconds to delay after turning the **RTS** signal on, but before data is sent. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DTR

A Indicates how a Driver deals with the **DTR** signal. Possible values are **OFF**: **DTR** signal is always turned off or **ON**: **DTR** signal is always turned on.

IO.Serial.InterbyteDelayUs

9 Delay time, in milliseconds (1/1000000 of a second), for each two bytes sent through a **Serial** Interface.

IO.Serial.InterframeDelayMs

9 Delay time, in milliseconds, before sending a packet after the last packet sent or received.

IO.Serial.Parity

A Specifies a parity for the configuration of a serial port. Possible values are **E or Even**: Even parity, **N or None**: No parity, **O or Odd**: Odd parity, **M or Mark**: Mark parity, or **S or Space**: Space parity.

IO.Serial.Port

9 Number of the local serial port. Possible values are **1**: Uses the COM1 port, **2**: Uses the COM2 port, **3**: Uses the COM3 port, or **n**: Uses the COMn port.

IO.Serial.RTS

A Indicates how a Driver deals with the **RTS** signal. Possible values are **OFF**: **RTS** signal always off, **ON**: **RTS** signal always on, or **Toggle**: Turns the **RTS** signal on when transmitting data and turns the **RTS** signal off when not transmitting data.

IO.Serial.StopBits

9 Specifies the number of stop bits for the configuration of a serial port. Possible values are **1**: One stop bit, **2**: One and a half stop bit, or **3**: Two stop bits.

IO.Serial.SuppressEcho

9 Use a value different from 0 (zero) to enable suppressing the echo or 0 (zero) to disable it.

IO.Serial.WaitCTS

☑ Configure to True to force a Driver to wait for the **CTS** signal before sending bytes when the **RTS** signal is turned on. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle**.

Driver's Revision History

VERSION	DATE	AUTHOR	COMMENTS
2.0.8	04/30/2026	M. Ludwig	<ul style="list-style-type: none"> Driver updated to IOKit library version 3.0 and Visual Studio 2022 (<i>Case 38067</i>).
2.0.7	08/27/2015	A. Quites	<ul style="list-style-type: none"> Driver ported to IOKit library version 2.0 (<i>Case 13752</i>). Added the reading of snapshot values (<i>Case 13733</i>). Added reading of harmonic's snapshot values (<i>Case 13943</i>). Fixed an omission in documentation regarding the usage of the <i>P1</i>

VERSION	DATE	AUTHOR	COMMENTS
			parameter of this Driver (Case 16468).
1.7.1	08/24/2012	G. Taschetto	<ul style="list-style-type: none"> Added a protection by code in the protection device (Case 12595).
1.6.1	12/11/2009	C. Mello	<ul style="list-style-type: none"> Adjustments to finish a session and set a device back to its initial Base State (Case 10697). Adjustments to acknowledge a "REJ" message referring to a session failure (Case 10464). Added a property window to configure the Default Slave Address and Internal Retry Error Timer parameters (Case 10547).
1.5.1	04/09/2009	A. Quites	<ul style="list-style-type: none"> Fixed hanging errors in this Driver (Case 9304). Fixed hanging errors when closing this Driver (Case 8101).
		F. Englert	<ul style="list-style-type: none"> Improvements and organization when generating logs in this Driver (Case 9419). Fixed an error in which this Driver returned only 10 records, considering the collecting process finished before returning all data (Case 7459).
1.4.1	10/27/2003	F. Englert	<ul style="list-style-type: none"> Created a Tag to force a reading of event logs. Writing to a Tag with the N2 parameter equal to 998 forces the log of events to be downloaded from a meter and stored on an internal memory of this Driver. A Block Tag with the N2 parameter equal to 3 (three) now simply returns all records stored on this Driver, without generating a new communication (Case 2839). Removed excessive logs and events with an invalid column (Case 2840).

VERSION	DATE	AUTHOR	COMMENTS
1.3.1	10/06/2003	Fabiano/Canopus	<ul style="list-style-type: none">Fixed a problem in which this Driver hangs when receiving a packet if the connection is lost (<i>Case 2646</i>).
1.0.1		Fabiano/Canopus	<ul style="list-style-type: none">Initial version of this Driver.

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)