

KB Name	Tag Quality		
Type	KB		
Since revision	6.0s0		
KB Number	KB-0039-0	Build	5
Mod date	23. Jul. 2010		

Tag Quality

1 Purpose

Before firmware version 6.0 it was not easy to know if a tag was read correctly or not. In case of polling problems, the value of the tag was set to 0 or left unchanged, which in most cases could be an acceptable value and thus not obviously indicating an error.

A close examination of the event logs could provide an error message, but notifying a problem to the user had to be made much more straightforward than this.

Up from firmware version 6.0, with the Tag quality feature the user can now see directly on the View IO page if the displayed Tag value is reliable or not.

2 Display of the Tag Quality in the View IO page

The View I/O page now indicates the quality for the value of each polled tag.

Tag Name	Value	New Value	Update	
Pressure1	1.0078	1.0078	Update	Pressure Tank 1
Temperature1	135.5	135.5	Update	Temperature Tank 1
Temperature2	55.5	55.5	Update	Temperature Tank 2

The Tag value which is displayed in red color indicates that the quality for this value is bad.

A new narrow column has been added between the tag's name and the tag's value. When the quality of the displayed value is 'good', the column is left empty; when the quality is 'uncertain', a warning icon is displayed and when the quality is 'bad', an error icon is displayed.

More information about the quality problem can be obtained by placing the mouse cursor on the icon.

As for example:

Temperature2		55.5	55.5
--------------	--	------	------

or

Temperature2		0	0
--------------	--	---	---

Knowledge Base Information

KB Name	Tag Quality		
Type	KB		
Since revision	6.0s0		
KB Number	KB-0039-0	Build	5
Mod date	23. Jul. 2010		

3 IO Servers that don't support the tag quality

Three IO servers do not support the quality :

1. The **MEM IO Server** : as this IO Server doesn't do any communication and only read/writes values in the RAM of the eWON, nothing can go wrong and the quality of the tag is always good.
2. The **EWON IO Server** accesses internal settings and IOs that are directly available on the main board of the eWON. Therefore, the quality of its tags is always good.
3. The **QWAVE IO Server** uses the MODBUS IO server and is therefore more a 'software IO server' than a real IO server. Therefore, the quality of the tags is always set to good. If the user wants to know if there is a communication problem with the device, he should read the QWAVE 'STATUS' registers.

3.1 Write-only tags

As the tag quality is only updated when reading tags, the quality of write-only tags is initially set to good. Users should not rely on the quality information when write-only tags are used.

4 Tag Quality field

The quality used in the eWON is based on the quality defined by the OPC Foundation.

It consists of a 16-bit value where

- Bits 15-8 are vendor-specific quality information
- **Bits 7-6** represent the **major quality (bad / uncertain / good)**
- Bits 5-2 represent the sub-status
- Bits 1-2 represent the limit status

Hereunder you will find the detailed description of each quality information.

For the **vendor-specific information** following formula is applied:

$$\text{TagQuality} = (\text{PreviousQuality} + 255 * \text{SuccessOfLastRead}) / 2$$

where

- TagQuality represents the new quality value
- PreviousQuality represents the previous value of TagQuality
- SuccessOfLastRead is 1 when the last read operation succeeded and 0 otherwise

KB Name	Tag Quality		
Type	KB		
Since revision	6.0s0		
KB Number	KB-0039-0	Build	5
Mod date	23. Jul. 2010		

Knowledge Base Information

The **major quality** can have one of the following values :

- Good (3) : when the last reading happened without problem.
- Bad (0) : when the calculated quality is lower than 32, meaning that at least the three last readings failed or that there is a problem with the configuration of the tag or the IO server.
Details about this situation can be retrieved from the quality sub-status bits.
- Uncertain (1) : when the last reading failed, but that the calculated quality is above 34, meaning that at least one of the last three readings succeeded. In this case, you can decide whether to trust the received value or not.

The **quality sub-status** can have one of the following values :

- Non-specific (0) : means that no additional information than the major quality is provided. This will be returned when the major quality is Good.
- Configuration Error (1) : indicates a configuration error (tag or IO server)
- Device Failure (3) : means that the device returned an error
- Comm Failure (6) : means that the device did not reply
- Out of Service (7) : means that the IO server or tag was disabled.

The **limit status** is currently not used by the eWON.

5 Retrieving the quality in Basic

You can easily read a tag's quality in BASIC in the following way :

```
SETSYS TAG,"load","MyTag"
PRINT GETSYS TAG,"TagValue" : REM The value of the tag
PRINT GETSYS TAG,"TagQuality" : REM The quality of the tag
```

The complete 16-bit quality field will be retrieved.

KB Name	Tag Quality		
Type	KB		
Since revision	6.0s0		
KB Number	KB-0039-0	Build	5
Mod date	23. Jul. 2010		

6 Export block descriptors and related files

6.1 Instant values – text format: dtIV ftT (inst_val.txt)

An additional field is returned. It's called "Quality" and contains the quality value of the tag. The complete 16-bit quality field is used here.

```
"TagId";"TagName";"Value";"AIStatus";"AIType";"Quality"
4;"sysUpTime_ewon";0.000000;0;0;24
6;"ifnumber_pc";3.000000;0;0;65472
13;"ipInReceives_prn";0.000000;0;0;28
```

6.2 Instant values – binary format: dtIV ftB (inst_val.bin)

Before firmware version 6.0s0, the inst_val.bin file was made of a header and tiny data records containing the tag id, the value, the alarm status, the alarm type and an unused field.

Now, the quality value is stored in a part of the previously unused field. The version number contained in the header has been increased.

Providing that the version number is not checked, old software will still be able to parse the new format without any problem. Newer software can interpret 16-bits of the previously unused field as the quality if the version is set to 2.

6.3 Historical Logging – text format: \$dtHL \$ftT (irc_###.txt)

The historical logging file contains now a supplementary field called 'IQuality'. Only the Major Quality information is reported in this column: 3 = Good, 0 = bad, 1 = uncertain

```
"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality"
1279893219;"23/07/2010 13:53:39";1;0;3
1279893219;"23/07/2010 13:53:39";0;0;3
1279893670;"23/07/2010 14:01:10";1;0;0
1279893851;"23/07/2010 14:04:11";1;0;0
1279893857;"23/07/2010 14:04:17";0;124;3
1279893911;"23/07/2010 14:05:11";1;124;0
1279893922;"23/07/2010 14:05:22";0;0;3
```

6.4 Historical Logging – binary format: \$dtHL \$ftB (ircall.bin)

The format has been slightly adapted to be able to reflect the Tag Quality and the Tag Type.

Knowledge Base Information

KB Name	Tag Quality		
Type	KB		
Since revision	6.0s0		
KB Number	KB-0039-0	Build	5
Mod date	23. Jul. 2010		

In fact, the 16 bit Msec field has now been split into 3 different parts: Msec (10 bits), Tag Type (4 bits) and IrcQuality (2 bits).

This should normally allow to minimize the impact of the new information fields for existing software which uses the ircall.bin file.

7 Export Data compatibility

A new parameter called "**PreRev6Compat**" has been added into the Config.txt file to allow backward compatibility to previous firmware versions (5.6).

By default the value of this parameter is set to 0.

When set to 1, the following files will keep the same format as before firmware version 6.0:

- \$dtHL \$ftB
- \$dtHL \$ftT
- \$dtIV \$ftT