UTC Timestamp Logging

For Flexy as of Firmware 13.2s0

SOLUTION SHEET

KB-0284-01 EN 1.0 ENGLISH

Connecting Devices™

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks SA of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks SA, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks SA. HMS Industrial Networks SA assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks SA will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks SA cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks SA has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

Table of Contents

Page

This page intentionally left blank

1 Preface

1.1 About This Document

This document details how the mechanism responsible for timestamps of the recorded data in the eWON Flexy is operating depending on the user's choice to use local time or UTC based model.

This document concerns the eWON Flexy only.

For additional related documentation and file downloads, please visit www.ewon.biz/support.

1.2 Document History

Version	Date	Description
1.0	2018-10-22	First release

1.3 Related Documents

Document	Author	Document ID

1.4 Trademark Information

2 Introduction

As of firmware **13.2s0**, the eWON Flexy gives the ability to the user to select a time zone through the configuration of the Flexy date & time. This way, Flexy handles the time changes or the daylight saving time on its own.

This feature also implies modifications how the Flexy stores and exports its data (historical logging, alarming, Export Block Descriptors, ...).

The user has the choice to record data either using local time or a UTC based model.

This document covers the following subjects:

- How to choose which timestamps model (local time or UTC) is active? Check *Time Zone Settings, p. 5*
- How is the data exported? Check *Date & Time Format when exporting Data, p. 6*
- What are the differences in exporting data between local time (legacy configuration) and UTC? Check *Recap of Data Export Modifications, p. 15*

3 Time Zone Settings

You can configure the time zone of your Flexy using either the system wizard or the date and time settings window.

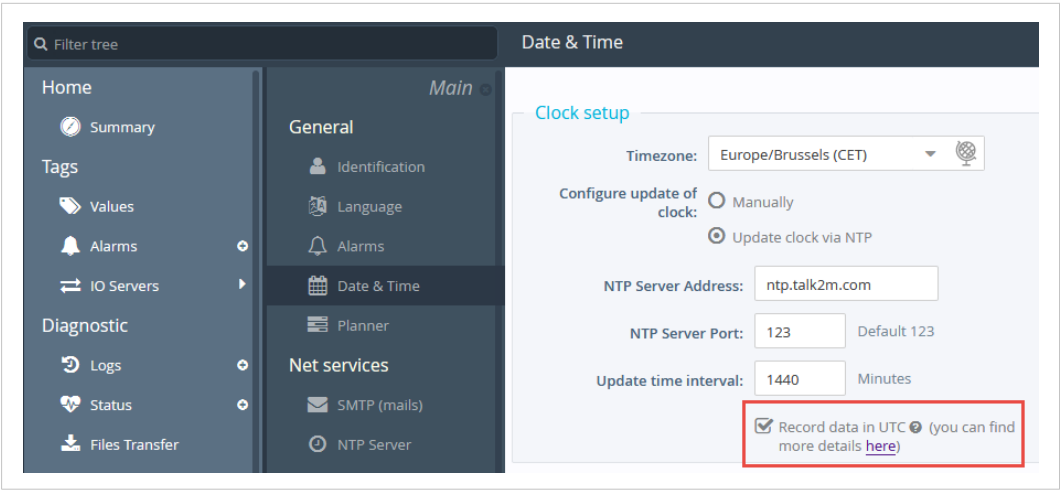

Fig. 1 Date & Time Configuration Panel

Go to **Setup** > **Main** > **General** > **Date & Time** to see the checkbox “Record data in UTC” which is responsible of setting the timestamps model:

- Box **unchecked**: the Flexy uses a timestamp based on its local time.
- Box **checked**: the Flexy uses a timestamp based on UTC format.

You can select the time zone corresponding to the location of your Flexy.

You can also define the synchronization of the Flexy time with an NTP server (recommended).

The Flexy displays the local time on its web interface and for alarm notifications (SMS/email) when the box “Record data in UTC” is checked. However, the data stored internally is using UTC timestamps.

4 Date & Time Format when exporting Data

The eWON Flexy can display/export data in different ways/methods. Data can be exported either in:

- UTC time.
- Local time.

In the following sub-chapters, we explain which model is applied as the “Record data in UTC” option is checked or not.

4.1 eWON Flexy Web Pages

The time is always the **local time** in the web interface of the Flexy.

This concerns the following pages and/or sections:

- current time (displayed on the status bar at the bottom of the window),
- alarm summary,
- alarm history,
- event log,
- real-time log,
- scheduled actions table.

4.2 eWON Flexy Data Files

The data files of a Flexy can be displayed by FTP – reaching the FTP server of your Flexy – or by browsing the “File transfer” page of your Flexy web interface.

4.2.1 Historical Data Files

This concerns the file(s) labeled as “irc_TAGNAME.txt”.

This type of file contains two elements of “time” notion:

- **TimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).
- **TimeStr**: in string format “DD/MM/YYYY HH:MM:SS”.

The time format used for those two elements:

	“Record data in UTC” checked	**“Record data in UTC” unchecked**
TimeInt	UTC time	local time
TimeStr	local time	local time

Example 1: Historical Data as “Record data in UTC” is checked

```
"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality"
1537338593;"19/09/2018 08:29:53";0;555;3
```

4.2.2 Alarm Real Time File

This concerns the file labeled as “rt_alm.txt”.

This file contains two elements of “time” notion:

- **AlarmTime**: timestamp of the begin of alarm (string format).
- **StatusTime**: timestamp of the AlStatus that is currently shown.

The time format used for the two elements regardless of the “Record data in UTC” checkbox state:

- AlarmTime: **local time**.
- StatusTime: **local time**.

Example 2: Alarm Real Time

```
"TagId";"AlarmTime";"TagName";"AlStatus";"AlType";"StatusTime";
"UserAck";"Description";"AlHint"

1;"19/09/2018 08:34:08";"Tag_001";"ALM";"HIHI";"19/09/2018 08:34:08";
"";"Tag_001";"Alarm On analog 1"
```

4.2.3 Alarm Historical File

This concerns the file labeled as “hst_alm.txt”.

This file contains a single element of “time” notion:

- **EventDate**: timestamp of the event in string.

The time format used for the element regardless of the “Record data in UTC” checkbox state:

- EventDate: **local time**.

Example 3: Alarm Historical

```
"EventDate";"TagName";"Status";"Type";"UserAck";"Description"
"19/09/2018 08:33:55";"Bool_021";"RTN";"";"";"Bool_021"
"19/09/2018 08:33:57";"Bool_022";"ALM";"LVL";"";"Bool_022"
```

4.2.4 Event Log

This concerns the file labeled as “events.txt”.

This file contains two elements of “time” notion:

- **EventTimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).
- **EventTimeStr**: timestamp in string format “DD/MM/YYYY HH:MM:SS”.

The time format used for those two elements:

	“Record data in UTC” checked	**“Record data in UTC” unchecked**
EventTimeInt	UTC time	local time
EventTimeStr	local time	local time

Example 4: Alarm Real Time as “Record data in UTC” is unchecked

```
"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";
"ThreadId";"Event"
```

```
1537338489;"19/09/2018 08:28:09";"wanmgt-Close WAN interface";"http";
79307;1073772970
```

4.2.5 Binary Historical Data File

This concerns the file labeled as “ircall.bin”.

This file contains a single element of “time” notion:

- **LogTime** (32 bits): Integer format (number of seconds since 01/01/1970 - UNIX Epoch time).

The time format used for this element:

	“Record data in UTC” checked	**“Record data in UTC” unchecked**
LogTime	UTC time	local time

4.3 Export Block Descriptors

You can configure how to export the data when it is exported through EBD (Export Block Descriptor).

We added a new EBD modificator: **$ts**, associated with 4 options ($ts#):

- **no modificator:** local time (local time format)
  - string format “DD/MM/YYYY HH:MM:SS”
  - Example: $ts renders “19/09/2018 12:48:12”
- **O:** local time (UTC format)
  - string format “DD/MM/YYYY HH:MM:SS”
  - Example: $ts renders “19/09/2018 10:48:12”
- **U:** timestamps are formatted in ISO 8601 ZULU (UTC format)
  - string format “YYYY-MM-DD**T**HH:MM:SS**Z**”
  - Example: $tsU renders “2018-09-19T10:48:12Z”
- **L:** timestamps are formatted in ISO 8601 local (local time format)
  - string format “YYYY-MM-DD**T**HH:MM:SS***±000***”
  - Example: $tsL renders “2018-09-19T12:48:12+0200”

4.3.1 Historical Logging

This concerns the following EBD syntax: **$dtHL**.

The result of this EBD contains 2 elements of “time” notion:

- **TimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).
- **TimeStr**: timestamp in string format (syntax depending on $ts modificator).

The time format used for those two elements:

Historical Logging with “Record data in UTC” checked

Modificator	TimeInt	TimeStr	Example
No modificator	UTC time	local time	"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality" 1537354078;"19/09/2018 12:47:58";0;556;3 1537354092;"19/09/2018 12:48:12";0;500;3
$tsO	UTC time	UTC time	"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality" 1537354078;"19/09/2018 10:47:58";0;556;3 1537354092;"19/09/2018 10:48:12";0;500;3
$tsU	UTC time	UTC time	"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality" 1537354078;"2018-09-19T10:47:58Z";0;556;3 1537354092;"2018-09-19T10:48:12Z";0;500;3
$tsL	UTC time	local time	"TimeInt";"TimeStr";"IsInitValue";"Value";"IQuality" 1537354078;"2018-09-19T12:47:58+0200";0;556;3 1537354092;"2018-09-19T12:48:12+0200";0;500;3

Historical Logging with “Record data in UTC” unchecked

Modificator	TimeInt	TimeStr
No modificator	local time	local time
$tsO	local time	UTC time
$tsU	local time	UTC time
$tsL	local time	local time

4.3.2 Historical Table

This concerns the following EBD syntax: **$dtHT**.

The result of this EBD contains 2 elements of “time” notion:

- **TimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).
- **TimeStr**: timestamp in string format (syntax depending on $ts modificator).

The time format used for those two elements:

Historical Table with “Record data in UTC” checked

Modificator	TimeInt	TimeStr	Example
No modificator	UTC time	local time	"TimeInt";"TimeStr";"Oil_temperature";"Tag_alarm1" 1537354931;"19/09/2018 13:02:11";0;505 1537354948;"19/09/2018 13:02:28";156;505
$tsO	UTC time	UTC time	"TimeInt";"TimeStr";"Oil_temperature";"Tag_alarm1" 1537354931;"19/09/2018 11:02:11";0;505 1537354948;"19/09/2018 11:02:28";156;505
$tsU	UTC time	UTC time	"TimeInt";"TimeStr";"Oil_temperature";"Tag_alarm1" 1537354931;"2018-09-19T11:02:11Z";0;505 1537354948;"2018-09-19T11:02:28Z";156;505
$tsL	UTC time	local time	"TimeInt";"TimeStr";"Oil_temperature";"Tag_alarm1" 1537354931;"2018-09-19T13:02:11+0200";0;505 1537354948;"2018-09-19T13:02:28+0200";156;505

Historical Table with “Record data in UTC” unchecked

Modificator	TimeInt	TimeStr
No modificator	local time	local time
$tsO	local time	UTC time
$tsU	local time	UTC time
$tsL	local time	local time

4.3.3 Real Time Logging

This concerns the following EBD syntax: **$dtRL**.

The result of this EBD contains 2 elements of “time” notion:

- **TimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).
- **TimeStr**: timestamp in string format (syntax depending on $ts modificator).

The time format used for those two elements:

Real Time Logging with “Record data in UTC” checked

Modificator	TimeInt	TimeStr	Example
No modificator	UTC time	local time	"TimeInt";"TimeStr";"Value" 1537355066;"19/09/2018 13:04:26";506 1537355076;"19/09/2018 13:04:36";506
$tsO	UTC time	UTC time	"TimeInt";"TimeStr";"Value" 1537355066;"19/09/2018 11:04:26";506 1537355076;"19/09/2018 11:04:36";506
$tsU	UTC time	UTC time	"TimeInt";"TimeStr";"Value" 1537355067;"2018-09-19T11:04:27Z";506 1537355077;"2018-09-19T11:04:37Z";506
$tsL	UTC time	local time	"TimeInt";"TimeStr";"Value" 1537355066;"2018-09-19T13:04:26+0200";506 1537355076;"2018-09-19T13:04:36+0200";506

Real Time Logging with “Record data in UTC” unchecked

Modificator	TimeInt	TimeStr	Example
No modificator	local time	local time	"TimeInt";"TimeStr";"Value" 1537355066;"19/09/2018 13:04:26";506 1537355076;"19/09/2018 13:04:36";506
$tsO	local time	UTC time	"TimeInt";"TimeStr";"Value" 1537355066;"19/09/2018 11:04:26";506 1537355076;"19/09/2018 11:04:36";506
$tsU	local time	UTC time	"TimeInt";"TimeStr";"Value" 1537355067;"2018-09-19T11:04:27Z";506 1537355077;"2018-09-19T11:04:37Z";506
$tsL	local time	local time	"TimeInt";"TimeStr";"Value" 1537355066;"2018-09-19T13:04:26+0200";506 1537355076;"2018-09-19T13:04:36+0200";506

4.3.4 Alarm History

This concerns the following EBD syntax: **$dtAH**.

The result of this EBD contains a single element of “time” notion:

- **EventDate**: timestamp of the event in string format (syntax depending on $ts modificator)

The time format used for this element:

Real Time Logging with “Record data in UTC” checked & unchecked

Modificator	EventDate	Example
No modificator	local time	"EventDate";"TagName";"Status";"Type";"UserAck";"Description" "19/09/2018 09:59:11";"Tag_alarm1";"ALM";"LO";"";"" "19/09/2018 10:02:21";"Tag_alarm1";"ALM";"LO";"";""
$tsO	UTC time	"EventDate";"TagName";"Status";"Type";"UserAck";"Description" "19/09/2018 07:59:11";"Tag_alarm1";"ALM";"LO";"";"" "19/09/2018 08:02:21";"Tag_alarm1";"ALM";"LO";"";""
$tsU	UTC time	"EventDate";"TagName";"Status";"Type";"UserAck";"Description" "2018-09-19T07:59:11Z";"Tag_alarm1";"ALM";"LO";"";"" "2018-09-19T08:02:21Z";"Tag_alarm1";"ALM";"LO";"";""
$tsL	local time	"EventDate";"TagName";"Status";"Type";"UserAck";"Description" "2018-09-19T09:59:11+0200";"Tag_alarm1";"ALM";"LO";"";"" "2018-09-19T10:02:21+0200";"Tag_alarm1";"ALM";"LO";"";""

4.3.5 Alarm Real Time

This concerns the following EBD syntax: **$dtAR**.

The result of this EBD contains 2 elements of “time” notion:

- **AlarmTime**: timestamp of the begin of alarm (in string format depending on $ts modificator)
- **StatusTime**: timestamp of the AlStatus that is currently shown (in string format depending on $ts modificator)

The time format used for those two elements:

Real Time Logging with “Record data in UTC” checked & unchecked

Modificator	AlarmTime	StatusTime	Example
No modificator	local time	local time	"TagId";"AlarmTime";"TagName";"AlStatus";"AlType";"Status-Time";"UserAck";"Description";"AlHint" 54;"19/09/2018 13:34:18";"Tag_alarm1";"ALM";"HIHI";"19/09/2018 13:34:18";"";"";""
$tsO	UTC time	UTC time	"TagId";"AlarmTime";"TagName";"AlStatus";"AlType";"Status-Time";"UserAck";"Description";"AlHint" 54;"19/09/2018 11:34:18";"Tag_alarm1";"ALM";"HIHI";"19/09/2018 13:34:18";"";"";""
$tsU	UTC time	UTC time	"TagId";"AlarmTime";"TagName";"AlStatus";"AlType";"Status-Time";"UserAck";"Description";"AlHint" 54;"2018-09-19T11:34:18Z";"Tag_alarm1";"ALM";"HIHI";"2018-09-19T11:34:18Z";"";"";""
$tsL	local time	local time	"TagId";"AlarmTime";"TagName";"AlStatus";"AlType";"Status-Time";"UserAck";"Description";"AlHint" 54;"2018-09-19T13:34:18+0200";"Tag_alarm1";"ALM";"HIHI";"2018-09-19T13:34:18+0200";"";"";""

4.3.6 Event File

This concerns the following EBD syntax: **$dtEV**.

The result of this EBD contains 2 elements of “time” notion:

- **EventTimeInt**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time)
- **EventTimeStr**: Timestamp in string format (syntax depending on $ts modificator)

The time format used for those two elements:

Real Time Logging with “Record data in UTC” checked

Modificator	EventTi-meInt	EventTimeStr	Example
No modificator	UTC time	local time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"19/09/2018 14:19:19";"main-Real Time Clock updated";"unact";79301;1073762139
$tsO	UTC time	UTC time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"19/09/2018 12:19:19";"main-Real Time Clock updated";"unact";79301;1073762139
$tsU	UTC time	UTC time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"2018-09-19T12:19:19Z";"main-Real Time Clock updated";"unact";79301;1073762139
$tsL	UTC time	local time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"2018-09-19T14:19:19+0200";"main-Real Time Clock updated";"unact";79301;1073762139.

Real Time Logging with “Record data in UTC” unchecked

Modificator	EventTi-meInt	EventTimeStr	Example
No modificator	local time	local time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"19/09/2018 14:19:19";"main-Real Time Clock updated";"unact";79301;1073762139
$tsO	local time	UTC time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"19/09/2018 12:19:19";"main-Real Time Clock updated";"unact";79301;1073762139
$tsU	local time	UTC time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"2018-09-19T12:19:19Z";"main-Real Time Clock updated";"unact";79301;1073762139
$tsL	local time	local time	"EventTimeInt";"EventTimeStr";"EventStr";"ThreadStr";"-ThreadId";"Event" 1537359559;"2018-09-19T14:19:19+0200";"main-Real Time Clock updated";"unact";79301;1073762139.

4.3.7 Scheduled Status

This concerns the following EBD syntax: **$dtSS**.

The result of this EBD contains 2 elements of “time” notion:

- **Start time**: timestamp in string format (syntax depending on $ts modificator)
- **Endt ime**: timestamp in string format (syntax depending on $ts modificator)

The time format used for those two elements:

Real Time Logging with “Record data in UTC” checked & unchecked

Modificator	Start time	End time	Example
No modificator	local time	local time	41;"Send Mail";0;"Success";"19/09/2018 13:25:01";"19/09/2018 13:25:38" 42;"Send Mail";0;"Success";"19/09/2018 13:25:10";"19/09/2018 13:26:17"
$tsO	UTC time	UTC time	41;"Send Mail";0;"Success";"19/09/2018 11:25:01";"19/09/2018 13:25:38" 42;"Send Mail";0;"Success";"19/09/2018 11:25:10";"19/09/2018 13:26:17"

Real Time Logging with “Record data in UTC” checked & unchecked (continued)

Modificator	Start time	End time	Example
$tsU	UTC time	UTC time	41;"Send Mail";0;"Success";"2018-09-19T11:25:01Z";"2018-09-19T11:25:38Z" 42;"Send Mail";0;"Success";"2018-09-19T11:25:10Z";"2018-09-19T11:26:17Z"
$tsL	local time	local time	41;"Send Mail";0;"Success";"2018-09-19T13:25:01 +0200";"2018-09-19T13:25:38+0200" 42;"Send Mail";0;"Success";"2018-09-19T13:25:10 +0200";"2018-09-19T13:26:17+0200"

4.3.8 Real Time Diagnostic

This concerns the following EBD syntax: **$dtRE**.

The result of this EBD contains a single element of “time” notion:

- **Time**: integer format (number of seconds since 01/01/1970 - UNIX Epoch time).

The time format used for this elements:

	“Record data in UTC” checked	“Record data in UTC” unchecked
Time	UTC time	local time

The operator “$ts” is not of any use for this EBD.

Example 5: Real Time Diagnostic

```
Time Clock Src Id Message
1537344815   41929   5   9   Initialization Sequence Completed
```

4.4 Data Management: DataMailbox

When the Flexy exports data to the DataMailbox, the following date format is used:

- The DataMailbox timestamps are formatted in ISO 8601 ZULU (UTC): “2018-09-20T07:26:27Z”.

If the “Record data in UTC” option is checked, the DMWeb API will add an extra parameter in the JSON feed that is returned for each DMWeb request. This extra parameter tells the applications using the DMWeb API that the data is recorded in UTC.

This extra parameter is called “timeZone” and returns the UTC + timezone of your Flexy.

If your Flexy does not record in UTC, as the “Record data in UTC” option is unchecked, the “timeZone” parameter is missing from the DMWeb JSON feed.

You can check the “Talk2M SDK” package which contains the “RG-0005-0: DataMailbox”. In the “RG-0005-0: DataMailbox”, you will find detailed explanation how the timestamps are dealt by DataMailbox. For further information, check eWON Developer website.

In the meantime, you can already have a summary of the situation with the below content.

4.4.1 getewons

The result of this call contains a single element of “time” notion regardless of the “Record data in UTC” checkbox state:

Item	Time format	Time reference
ewons.[#].lastSynchroDate	UTC	Talk2M server time

4.4.2 getewon

The result of this call contains a single element of “time” notion regardless of the “Record data in UTC” checkbox state:

Item	Time format	Time reference
lastSynchroDate	UTC	Talk2M server time

4.4.3 getdata

The result of this call contains four elements of “time” notion:

Item	Time format with “Record data in UTC” checked	Time format with “Record data in UTC” unchecked	Time reference
ewons.[#].tags.[#].history.[#].date	UTC	local time	eWON time
ewons.[#].tags.[#].alarmState.dateStart	UTC	local time	eWON time
ewons.[#].tags.[#].alarmState.dateStatus	UTC	local time	eWON time
ewons.[#].lastSynchroDate	UTC	UTC	Talk2M server time

4.4.4 syncdata

The result of this call contains five elements of “time” notion:

Item	Time format with “Record data in UTC” checked	Time format with “Record data in UTC” unchecked	Time reference
ewons.[#].tags.[#].history.[#].date	UTC	local time	eWON time
ewons.[#].tags.[#].alarmState.dateStart	UTC	local time	eWON time
ewons.[#].tags.[#].alarmState.dateStatus	UTC	local time	eWON time
ewons.[#].tags.[#].alarmHistory.[#].date	UTC	local time	eWON time
ewons.[#].lastSynchroDate	UTC	UTC	Talk2M server time

5 Recap of Data Export Modifications

The timestamps format of the Flexy data export depends on the “Record data in UTC” checkbox, available in the date & time configuration of your Flexy.

5.1 eWON Flexy Web Pages

All data is displayed in local time.

5.2 eWON Flexy Data Files

The time displayed in string format remains the same. Local time is used regardless of the “Record data in UTC” checkbox state.

The integer representation of the date (like *TimeInt* and *EventTimeInt*) are exported in UTC if “Record data in UTC” is checked.

Filename	Parameter	If “Record data in UTC” is unchecked	If “Record data in UTC” is checked
irc_TAGNAME.txt	TimeInt TimeStr	local time local time	**UTC time** local time
rt_alm.txt	AlarmTime StatusTime	local time local time	local time local time
hst_alm.txt	EventDate	local time	local time
events.txt	EventTimeInt EventTimeStr	local time local time	**UTC time** local time
ircall.bin	LogTime	local time	**UTC time**

5.3 Export Block Descriptors

The integer representation of the date (like *TimeInt* and *EventTimeInt*) depends on the “Record data in UTC” checkbox.

The operator $ts can be used to export the time string in an other format.

EBD	Parameter	If “Record data in UTC” is unchecked	If “Record data in UTC” is checked
$dtHL	TimeInt TimeStr	local time local time	**UTC time** local time
$dtHT	TimeInt TimeStr	local time local time	**UTC time** local time
$dtRL	TimeInt TimeStr	local time local time	**UTC time** local time
$dtAH	EventDate	local time	local time
$dtAR	AlarmTime StatusTime	local time local time	local time local time
$dtEV	EventTimeInt EventTimeStr	local time local time	**UTC time** local time
$dtSS	Start time End time	local time local time	local time local time
$dtRE	Time Clock	local time	**UTC time**

5.4 Data Management: DataMailbox

The Flexy exports its data in a defined format based on the selection of the “Record data in UTC” box.

Function	Parameter	If “Record data in UTC” is unchecked	If “Record data in UTC” is checked
getewons	ewons.[#].lastSynchroDate	UTC time	UTC time
getewon	lastSynchroDate	UTC time	UTC time
getdata	ewons.[#].tags.[#].history.[#].date ewons.[#].tags.[#].alarmState.dateStart ewons.[#].tags.[#].alarmState.dateStatus ewons.[#].lastSynchroDate	local time local time local time UTC time	**UTC time** **UTC time** **UTC time** UTC time
syncdata	ewons.[#].tags.[#].history.[#].date ewons.[#].tags.[#].alarmState.dateStart ewons.[#].tags.[#].alarmState.dateStatus ewons.[#].tags.[#].alarmHistory.[#].date ewons.[#].lastSynchroDate	local time local time local time local time UTC time	**UTC time** **UTC time** **UTC time** **UTC time** UTC time

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se
KB-0284-01 EN 1.0 / 2018-11-13 / 10507