Weidmüller

UC20-SL2000-OLAC-EC

Quick Start Guide for using the data recorder in u-create studio

Abstract:
This Quick Start Guide shows how the integrated data recorder of the UC20-SL2000-OLAC-EC can be used to sample a set of data and write it to a CSV file. It shows how this can be achieved out of a u-create studio project.

Hardware reference

No.	Component name	Article No.	Hardware / Firmware version
1	UC20-SL2000-OLAC-EC	2638920000	HW 01.xx.xx / FW 01.18

Software reference

No.	Software name	Article No.	Software version
1	u-create studio	2660130000	1.18a
2	WinSCP	-	-

File reference

No.	Name	Description	Version
1	-	-	-

Contact
Weidmüller Interface GmbH & Co. KG
Klingenbergstraße 26
32758 Detmold, Germany
www.weidmueller.com

For any further support please contact your local sales representative:
https://www.weidmueller.com/countries

Content

1 Warning and Disclaimer

Warning
Controls may fail in unsafe operating conditions, causing uncontrolled operation of the controlled devices. Such hazardous events can result in death and / or serious injury and / or property damage. Therefore, there must be safety equipment provided / electrical safety design or other redundant safety features that are independent from the automation system.

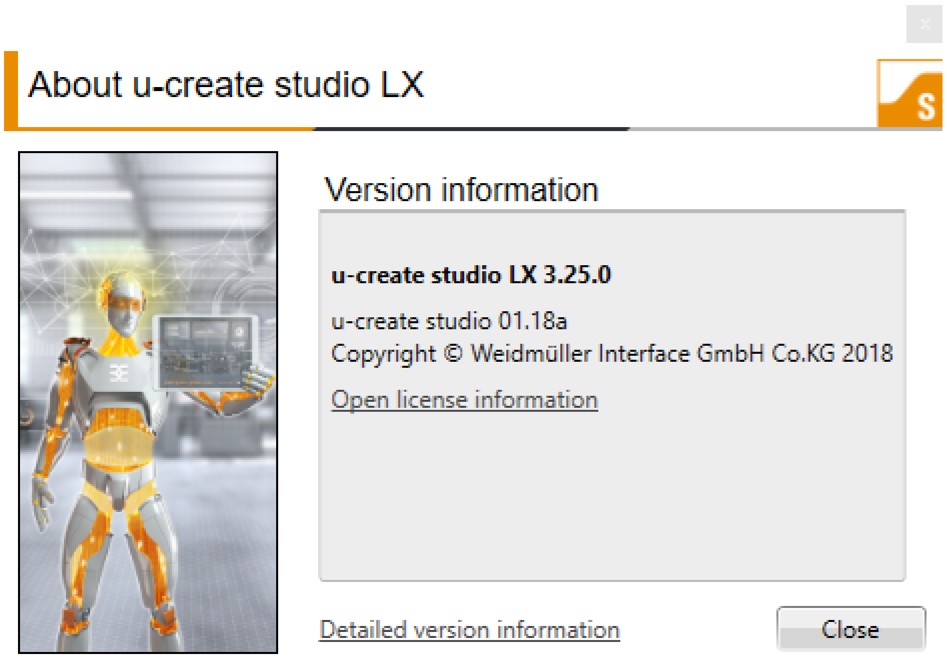
Disclaimer
This Application Note / Quick Start Guide / Example Program does not relieve you of the obligation to handle it safely during use, installation, operation and maintenance. Each user is responsible for the correct operation of his control system. By using this Application Note / Quick Start Guide / Example Program prepared by Weidmüller, you accept that Weidmüller cannot be held liable for any damage to property and / or personal injury that may occur because of the use.

Note
The given descriptions and examples do not represent any customer-specific solutions, they are simply intended to help for typical tasks. The user is responsible for the proper operation of the described products. Application notes / Quick Start Guides / Example Programs are not binding and do not claim to be complete in terms of configuration as well as any contingencies. By using this Application Note / Quick Start Guide / Example Program, you acknowledge that we cannot be held liable for any damages beyond the described liability regime. We reserve the right to make changes to this application note / quick start guide / example at any time without notice. In case of discrepancies between the proposals Application Notes / Quick Start Guides / Program Examples and other Weidmüller publications, like manuals, such contents have always more priority to the examples. We assume no liability for the information contained in this document. Our liability, for whatever legal reason, for damages caused using the examples, instructions, programs, project planning and performance data, etc. described in this Application Note / Quick Start Guide / Example is excluded.

Security notes
In order to protect equipment, systems, machines and networks against cyber threats, it is necessary to implement (and maintain) a complete state-of-the-art industrial security concept. The customer is responsible for preventing unauthorized access to his equipment, systems, machines and networks. Systems, machines and components should only be connected to the corporate network or the Internet if necessary and appropriate safeguards (such as firewalls and network segmentation) have been taken.

2 Requirements

This Quick Start Guide was created with u–create studio version 1.18.a. Another version may work, but the author does not take any responsibility.


About u-create studio LX

Version information

u-create studio LX 3.25.0

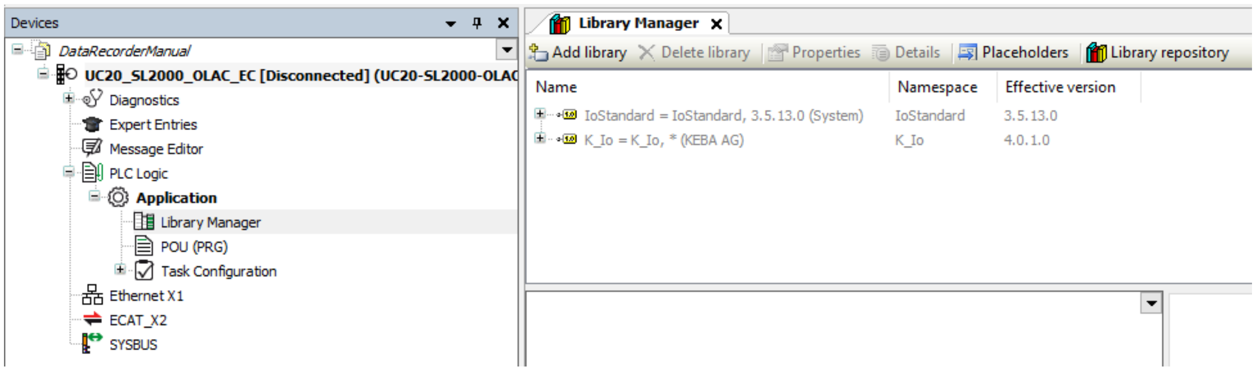
u-create studio 01.18a
Copyright © Weidmüller Interface GmbH Co.KG 2018

The library `K_DataRec` is necessary to interact with the data recorder from the IEC runtime. This example is tested with version 4.0.1.0. Any other version might work as well, but is not yet tested and as so, same results while using any other version can't be guaranteed.

3 Adding the library K_DataRec to the u-create studio project

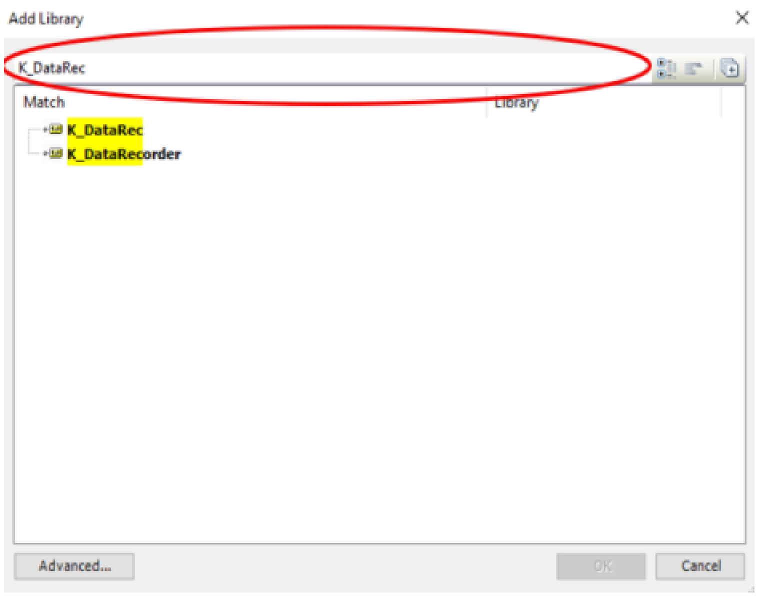
The interface to the data recorder is the library `K_DataRec`.
To able to use the data recorder, this library needs to be included in the u-create studio project.
To add the library:

1) Open the Library Manager.
2) Click on the “Add library” button.



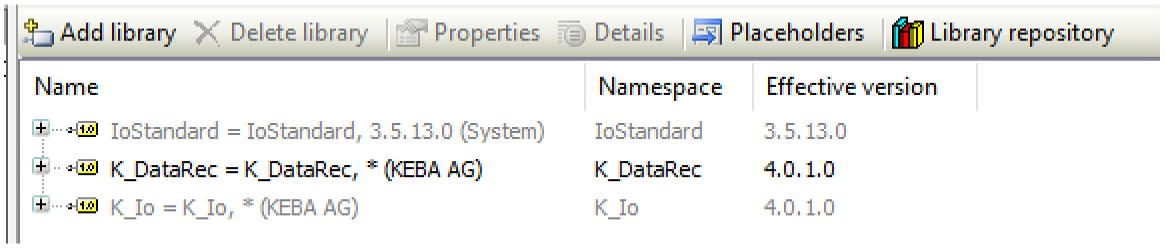
Name	Namespace	Effective version
IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard	3.5.13.0
K_Io = K_Io, * (KEBA AG)	K_Io	4.0.1.0

3) Now search for the library by entering the name into the input field at the top pf the window. After the library appears in the match box, the user can add the library by clicking on its name.



Do not add the library `K_DataRecorder`, as it provides another interface to the data recorder.

4) If everything went all right, the Library Manager should now contain the library.



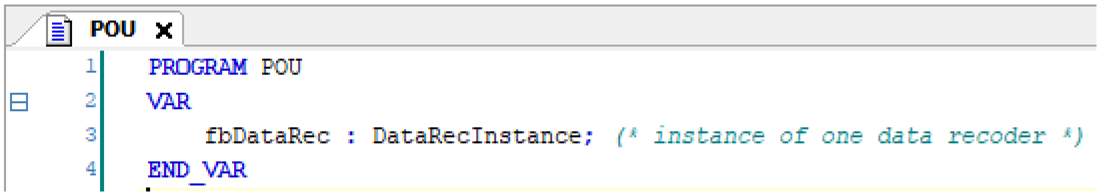
Name	Namespace	Effective version
IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard	3.5.13.0
K_DataRec = K_DataRec, * (KEBA AG)	K_DataRec	4.0.1.0
K_Io = K_Io, * (KEBA AG)	K_Io	4.0.1.0

4 Using the library K_DataRec to control the data recorder

The scope of this Quick Start Guide is the access to the data recorder by using the library `K_DataRec`. This library contains function blocks to work with the data recorder in an object-oriented manner. However, there exists the library `K_DataRecorder` that allows the user to control a data recorder in a more procedural manner by using functions. This library is **NOT** in the scope of this document.

4.1 Configuration of the data recorder

The access to a recorder is represented by the function block `DataRecInstance`. This function block is the origin for any further action the user performs on the data recorder. For every data recorder that shall be used in the project, an instance of this function block needs to be declared.



```
PROGRAM POU
VAR
    fbDataRec : DataRecInstance; (* instance of one data recoder *)
END_VAR
```

Steps in the PLC program:

1) Create a data recorder in the runtime system. Therefore, the function block offers the method `Create(…)`. All necessary parameters are part of the method call and need to be added.

```
stErrorID := fbDataRec.Create(  RecorderName := 'Test Recorder',
                                MaxSampleCnt := 4093,
                                MaxVarCnt := 1,
                                BufferType := BufferType.SingleShot,
                                Autostart := FALSE,
                                RecorderId => stRecorderID);
```

The data types of the shown variables are part of the library. The following table gives the user an overview how the variables need to be declared.

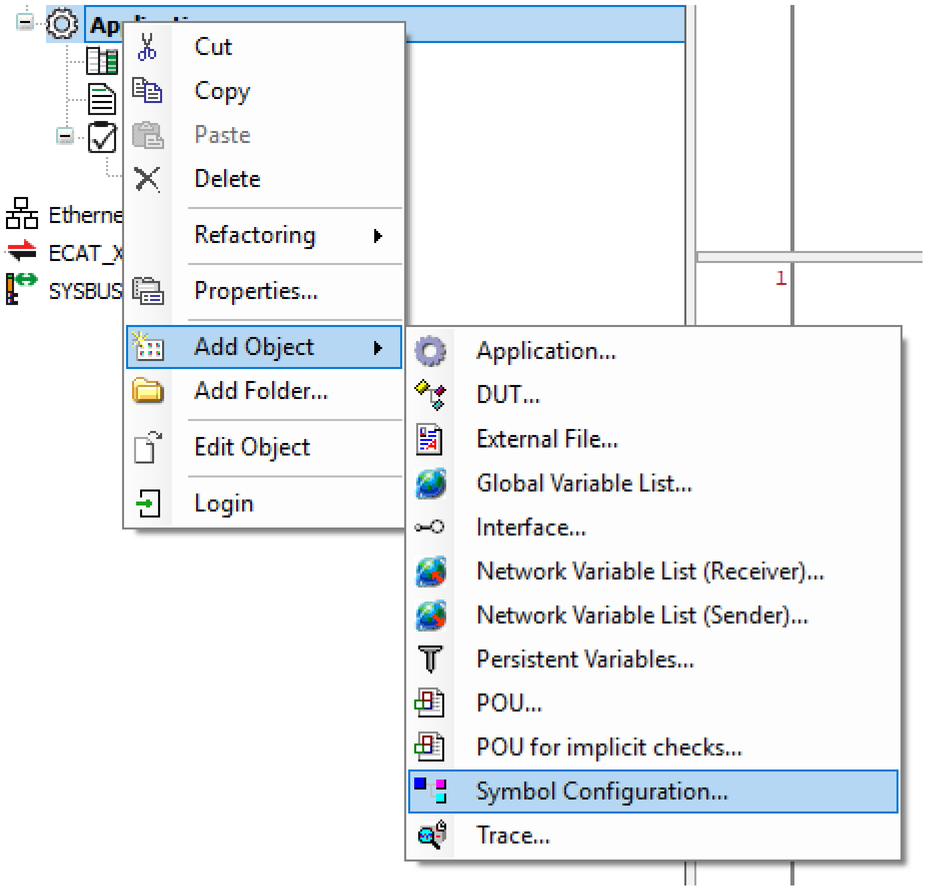
Variable name	Datatype
`stErrorID`	K_DataRec.TErrorID
`stRecorderID`	RecorderId

`RecorderName`: It represents the name of the recorder and needs to be unique on the complete controller platform.
`MaxSampleCnt`: It defines how many data points the recorder can store. The maximum amount is “4093”.
`MaxVarCnt`:It defines the number of variables that can be added to the data recorder. The maximum amount is “255”.

`Autostart:` It is only necessary in the context of persistent data recorders, what is **NOT** in scope of this document.
`BufferType:` It has the enumeration datatype. The meaning of its values is summarized in the following table.

Value	Meaning
`BufferType.SingleShot`	Recording until the buffer size is reached
`BufferType.Continuous`	Continuous recording (ring buffer)

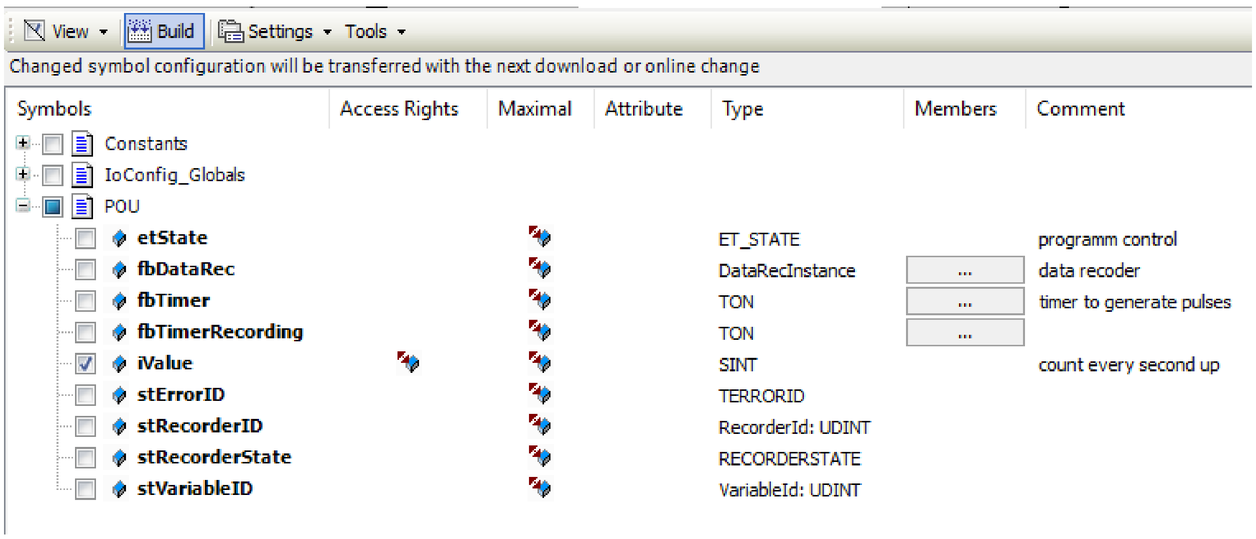
2) After the data recorder instance on the runtime system is created, the user needs to add variables to it. Adding IEC variables to the variable server needs to be done via a Symbol Configuration. If there is no Symbol Configuration in the project, it needs to be created.



The user needs to add variables to the data recorder because the data recorder is not just a subsystem of the IEC part of the runtime system, but of the complete u-create middleware and accessible also via C/C++, all the data that shall be recorded needs to be part of the variable server.

Once the Symbol Configuration has been created, the user can add variables that shall be used by a data recorder to it.

3) The following screenshot shows how to insert the variable `iValue` to the Symbol Configuration.



Changed symbol configuration will be transferred with the next download or online change

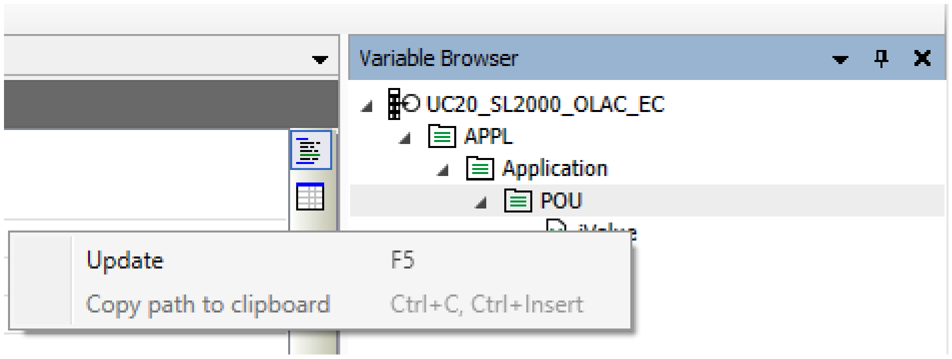
Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
Constants						
IoConfig_Globals						
POU						
etState				ET_STATE		programm control
fbDataRec				DataRecInstance	...	data recoder
fbTimer				TON	...	timer to generate pulses
fbTimerRecording				TON	...	
✓ iValue				SINT		count every second up
stErrorID				TERRORID		
stRecorderID				RecorderId: UDINT		
stRecorderState				RECORDERSTATE		
stVariableID				VariableId: UDINT		

4) Adding the variable to the data recorder is done by calling the method `AddVariable(…)`.

```
stErrorID := fbDataRec.AddVariable( VariableName := 'APPL.Application.POU.iValue',
                                    VariableID => stVariableID );
```

The output `VariableID` is from the datatype `VariableId`, which is a part of the library.
The input parameter `VariableName` is from the datatype string and represents the path to the variable on the variable server.

5) To get the path of a variable on the variable server, the Variable Browser (open through the view menu) can be used while u-create studio is connected to the controller. The user can navigate to the variable and add the path through a right click to the clipboard (see picture below).



4.2 Starting and stopping the data recorder

Once the data recorder is created and configured, it is ready to be used in many ways.
This document will cover two ways of sampling data.

1) Sampling via a trigger pulse from the IEC program.
2) Sampling with a continuous sampling rate.

There are some more possibilities to record data, but these possibilities are **NOT** in the scope of this document.

To record data in a continuous manner, the recorder needs to be parametrized to do so. Therefore, the user needs to call a method with the name `SetRecordingContext(…).`This method has the input parameters `IntervalUs` and `Priority`.
`IntervalUs:` It defines the sampling rate in microseconds, a manual triggering is enforced by setting this parameter to zero.
`Priority`: The priority parameter defines the priority of the sampling task. It can be set from “4” (high priority) to “31” (low priority).

```
(* specify a sampling rate *)
stErrorID := fbDataRec.SetRecordingContext(IntervalUs := 2500, Priority := 20);
```

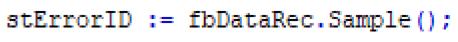
Before any data can be recorded, the data logger needs to be started. This is achieved with the method `StartRecording()`. The method call changes the recorder state from “Stopped” to “Recording”.

```
ET_State.startRecording:
    (* request current state of recorder *)
    stErrorID := fbDataRec.GetState( RecorderState => stRecorderState );
    (* check recorder to be recording *)
    IF ( stRecorderState = K_DataRec.RecorderState.Recording ) THEN
        (* recorder is recording *)
        IF ( stErrorID = K_DataRec.TErrorID.eOk ) THEN
            (* start waiting on sampling periode exceeded *)
            etState := ET_State.waitRecordingFinished;
        ELSE
            (* error occured, change to idle state *)
            etState := ET_State.idle;
        END_IF
    ELSE
        (* if the recorder is not recording, start the recorder *)
        stErrorID := fbDataRec.StartRecording();
        (* start the timer for sampling periode *)
        fbTimerRecording.IN := TRUE;
    END_IF
```

To estimate the current state of the data recorder, the function block offers the method `GetState(…)`. The methods output value “`RecorderState`” has the enumeration datatype `K_DataRec.RecorderState` which is part of the library.

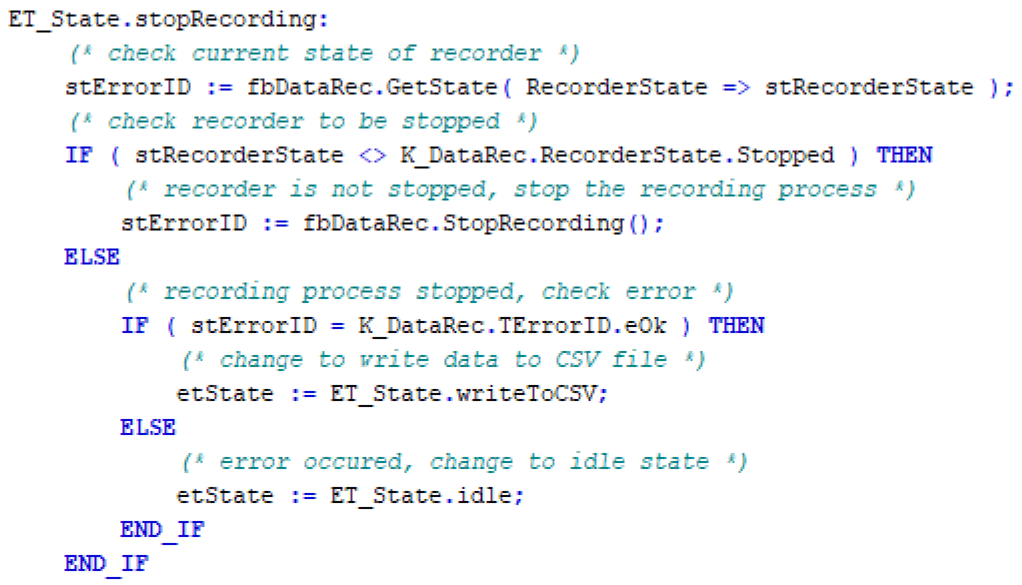
Value of RecorderState	**Meaning**
Stopped	Recorder is stopped. Sample() will be ignored.
Waiting	Not supported at the moment
Recording	Started and recording samples

In case the sampling interval is set to zero, the recorder does not collect the data automatically. That means sampling needs to be triggered manually by the method `Sample()`. This method is called every time a sample should be taken and allows the IEC application to control the whole process of data recording.



```
stErrorID := fbDataRec.Sample();
```

Stopping the recorder is also possible and it is achieved by the call of the method `StopRecording()`. After calling this method, the recorder changes to the state Stopped.



```
ET_State.stopRecording:
    (* check current state of recorder *)
    stErrorID := fbDataRec.GetState( RecorderState => stRecorderState );
    (* check recorder to be stopped *)
    IF ( stRecorderState <> K_DataRec.RecorderState.Stopped ) THEN
        (* recorder is not stopped, stop the recording process *)
        stErrorID := fbDataRec.StopRecording();
    ELSE
        (* recording process stopped, check error *)
        IF ( stErrorID = K_DataRec.TErrorID.eOk ) THEN
            (* change to write data to CSV file *)
            etState := ET_State.writeToCSV;
        ELSE
            (* error occured, change to idle state *)
            etState := ET_State.idle;
        END_IF
    END_IF
```

4.3 Creating a CSV file with the content of the data recorder

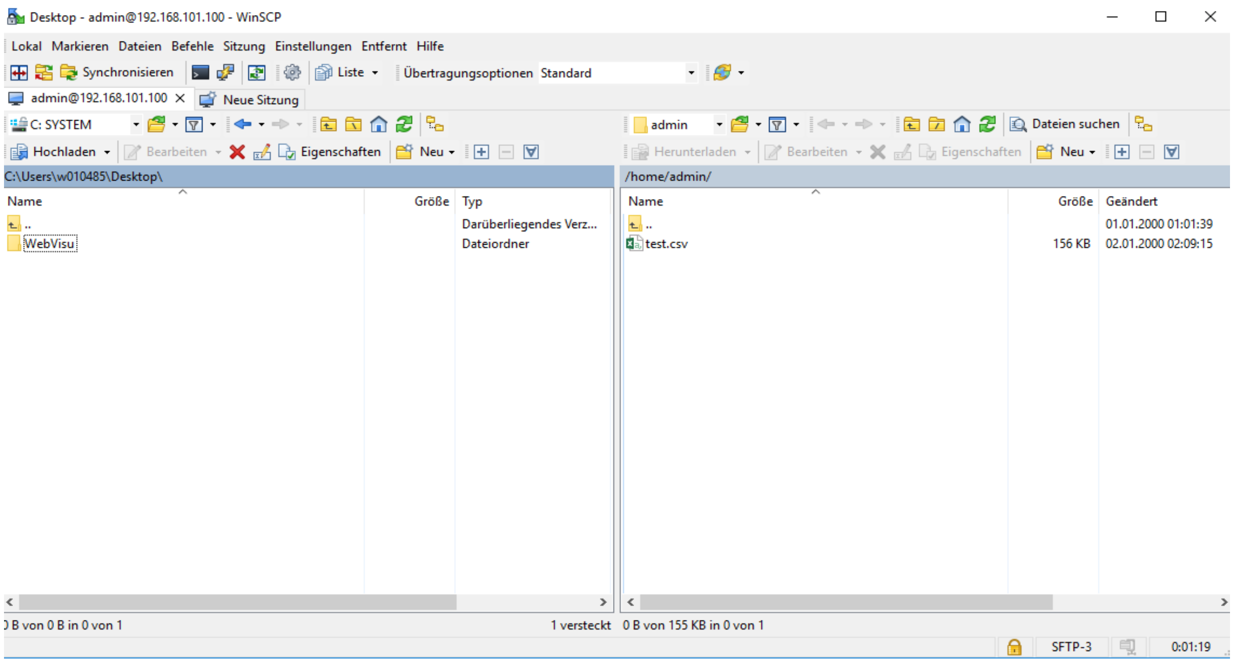
After the data recorder has collected some data, it remains in the random-access memory of the controller. To store it longer then the next power cycle or use it in another program e.g. Excel, it can be written to the controllers file system.

Therefore, the function block offers the method `DumpToCSV(…)`

```
ET_State.writeToCSV:
    (* dump all the data to a CSV file *)
    stErrorID := fbDataRec.DumpToCSV(FileName := '/home/admin/test.csv');
    etState := ET_State.finished;
```

The methods input parameter `FileName` is from the datatype string and contains the paths and the filename.

To access the controller's filesystem and to copy the CSV file to a PC, the user can use the tool "WinSCP". A key file to authenticate at the controller is needed, please refer u-create studio Quick Start Guide(QGUI_UC20-studio_en / Document-Nr. 2729550000 / Revision 01 / April 2021) for any further information on how this key file needs to be generated.



C:\Users\w010485\Desktop\

Name	Größe	Typ
..		Darüberliegendes Verz...
WebVisu		Dateiordner

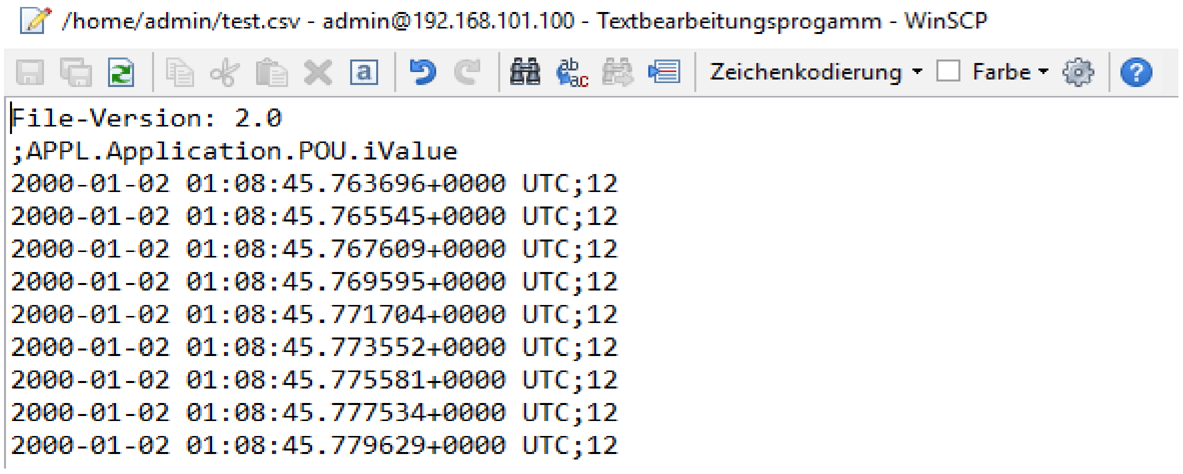
/home/admin/

Name	Größe	Geändert
..		01.01.2000 01:01:39
test.csv	156 KB	02.01.2000 02:09:15

0 B von 0 B in 0 von 1 — 1 versteckt — 0 B von 155 KB in 0 von 1

SFTP-3 — 0:01:19

The data in the CSV file is stored in the following manner.


/home/admin/test.csv - admin@192.168.101.100 - Textbearbeitungsprogamm - WinSCP
Zeichenkodierung
Farbe

```
File-Version: 2.0
;APPL.Application.POU.iValue
2000-01-02 01:08:45.763696+0000 UTC;12
2000-01-02 01:08:45.765545+0000 UTC;12
2000-01-02 01:08:45.767609+0000 UTC;12
2000-01-02 01:08:45.769595+0000 UTC;12
2000-01-02 01:08:45.771704+0000 UTC;12
2000-01-02 01:08:45.773552+0000 UTC;12
2000-01-02 01:08:45.775581+0000 UTC;12
2000-01-02 01:08:45.777534+0000 UTC;12
2000-01-02 01:08:45.779629+0000 UTC;12
```