

# **EUCHNER**

## **Software Manual**

**EKS Data Service PLC**  
Electronic-Key-System EKS

**EN**

## Contents

<b>1.</b>	<b>General notes</b> .....	<b>4</b>
1.1.	Use of the manual.....	4
1.2.	Requirement for the user.....	4
1.3.	System requirements .....	4
<b>2.</b>	<b>General functions of the application</b> .....	<b>5</b>
<b>3.</b>	<b>Purpose</b> .....	<b>6</b>
<b>4.</b>	<b>S7-1200 / S7-1500 TIA Portal</b> .....	<b>7</b>
4.1.	Integrating library into TIA Portal .....	7
4.2.	EKSDataService library .....	7
4.2.1.	EKSDataServiceBase .....	7
4.2.2.	EKSDataServiceExample .....	8
4.3.	EKSDataService block description.....	10
4.3.1.	DB_EKSData1 .....	10
4.3.2.	Data of placed Electronic-Key (type_EKSDatabase) .....	10
4.3.3.	FB_EKSDataService .....	13
<b>5.</b>	<b>S7-300 / S7-400 TIA Portal</b> .....	<b>19</b>
5.1.	Integrating library into TIA Portal .....	19
5.2.	EKSDataService library .....	19
5.2.1.	Standard Siemens blocks used .....	19
5.2.2.	FB_EKSDataService block .....	20
5.2.3.	DB_EKSData1 block.....	20
5.2.4.	Data types type_EKSxxxx .....	20
5.2.5.	FC_EKS_Euchner block .....	20
5.3.	Settings (TCP/IP configuration) .....	22
5.3.1.	Setting up connection in the CPU .....	22
5.4.	EKSDataService block description.....	24
5.4.1.	DB_EKSData1 .....	24
5.4.2.	FB_EKSDataService .....	29
<b>6.</b>	<b>S7-300 / S7-400 SIMATIC STEP7</b> .....	<b>35</b>
6.1.	Integrating library into SIMATIC STEP7.....	35
6.2.	EKSDataService library .....	35
6.2.1.	Standard Siemens blocks used .....	35
6.2.2.	System blocks used.....	36
6.2.3.	FB_EKSDataService block .....	36
6.2.4.	DB_EKSData1 block.....	36
6.2.5.	Data types type_EKSxxxx .....	36
6.2.6.	FC_EKS_Euchner block .....	37
6.3.	NetPro settings (TCP/IP configuration).....	38
6.3.1.	Setting up EKS Data Service PC.....	38
6.4.	EKS Data Service block description.....	42
6.4.1.	DB_EKSData1 .....	42
6.4.2.	FB_EKSDataService .....	46

<b>7.</b>	<b>Status messages .....</b>	<b>51</b>
7.1.	Connection setup (Con_Parametrization_Errorcode) .....	51
7.2.	Data reception (Con_Communication_Errorcode).....	53
7.3.	Status of FB_EKSDataService (Status).....	54
7.4.	Data request status (serviceStatus).....	54

## 1. General notes

### 1.1. Use of the manual

This manual describes the functions of the EKS Data Service PLC software components as part of the EUCHNER EKS Data Service Integration Kit (order no. 8163316).

### 1.2. Requirement for the user

Using EKS Data Service PLC properly requires prior knowledge in the field of PLC programming using SIMATIC Manager STEP7 or TIA Portal. You should also possess prior knowledge in network configuration.

### 1.3. System requirements

The following Siemens control systems and configuration software are supported:

Hardware:	SIMATIC S7-1200
	SIMATIC S7-1200 + CP1243
	SIMATIC S7-1500
	SIMATIC S7-1500 + CP1543
	SIMATIC S7-300 + CP343
	SIMATIC S7-400 + CP443
Software:	SIMATIC S7-300, S7-400, S7-1200, S7-1500 – from TIA Portal V14 SP1
	SIMATIC S7-300, S7-400 – from SIMATIC STEP7 V5.5

## 2. General functions of the application

Data comparison or data retrieval from a central database is currently not established in most EKS applications in the PLC world. This means that the data are almost always read decentrally from the Electronic-Key and then processed individually in the PLC. Access information is thus transmitted exclusively via the Electronic-Key in this case. Many EKS operators wish to implement data comparison from the PLC world using EKM data in the PC world. The central topic is the desire to block Electronic-Keys centrally and to retrieve further data if necessary.

The Electronic-Key-Manager EKM database content is exported to a universally usable file in CSV format in the PC environment. A blocking code behind the Electronic-Key's serial number (KeyID) can be evaluated in this EKM CSV export file. This code is set to "1" as soon as the Electronic-Key is blocked.

Other data elements assigned to the Electronic-Key serial number can also be retrieved. These data elements differ for specific applications.

A request is sent from EKS Data Service PLC to EKS Data Service PC when an Electronic-Key is placed. Based on the KeyID, EKS Data Service PC searches for the entry in the EKM CSV export file and then returns the data to EKS Data Service PLC. The data are now available there to the user for further processing. Additionally, the requested data are stored in an emergency memory. If the connection to the PC is interrupted, the data of previously placed Electronic-Keys are loaded from the emergency-level memory. Up to the last 100 Electronic-Key data items are available in the emergency-level memory (size of the emergency-level memory can be adapted).

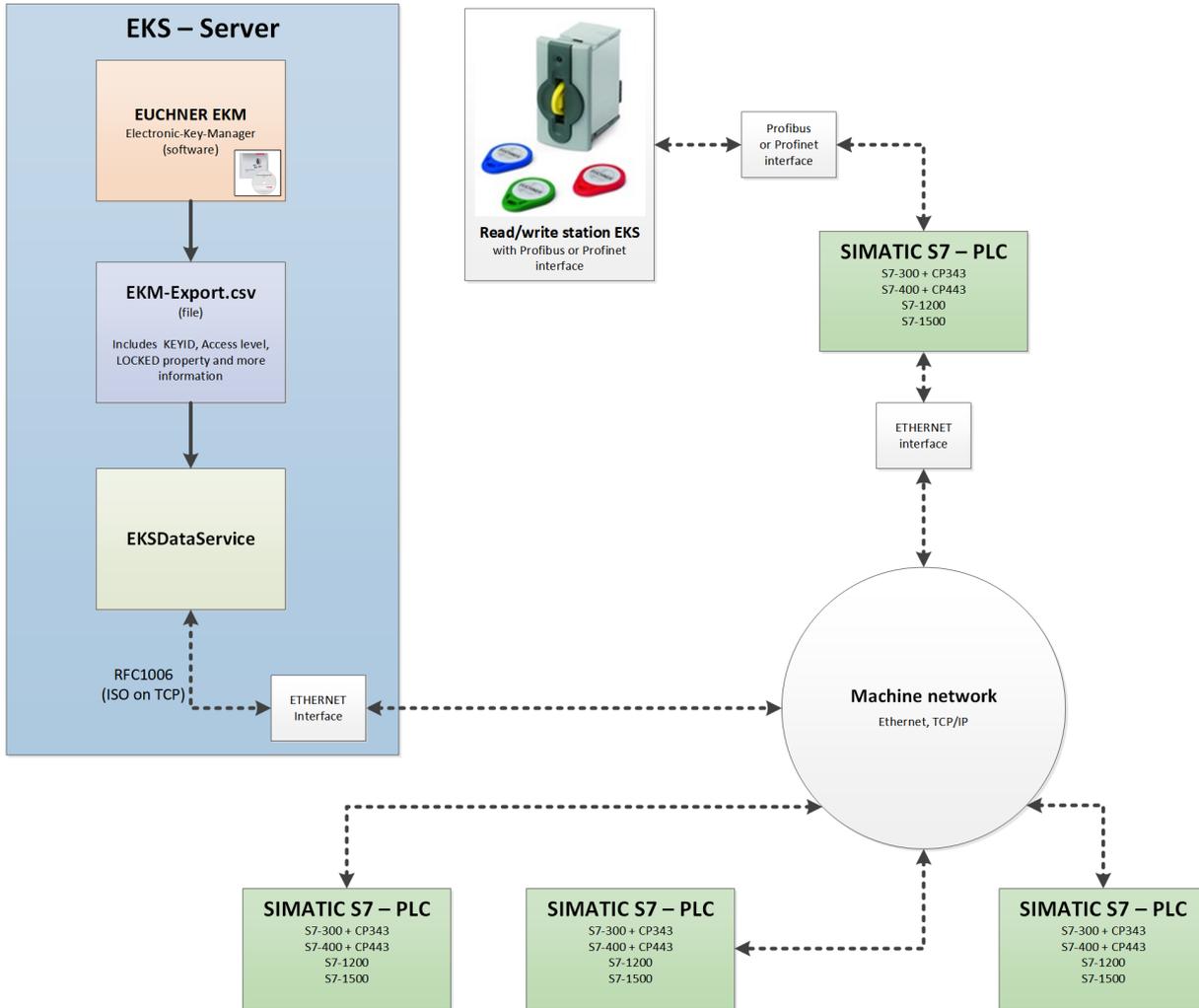
**Notice about the emergency level:** If the emergency-level memory is full and a new Electronic-Key is requested, the Electronic-Key that has not been requested for the longest time will be replaced. Entries of the emergency level are updated cyclically, so it is not necessarily the first Electronic-Key placed that will be replaced.

There are two emergency levels: In case of emergency level 1, EKS Data Service PC cannot access the original EKM CSV export file and instead uses the local backup file of the EKM CSV export file. In case of emergency level 2, EKS Data Service PC cannot provide any data because communication is disrupted. The data of the internal PLC data block are used instead.

### 3. Purpose

The “**EUCHNER EKS Data Service PC**” service is installed on a server, and it is used to distribute EKS Electronic-Key data to SIMATIC control systems that are networked with this server via Ethernet TCP/IP. EUCHNER Electronic-Key-Manager EKM generates the EKS Electronic-Key data.

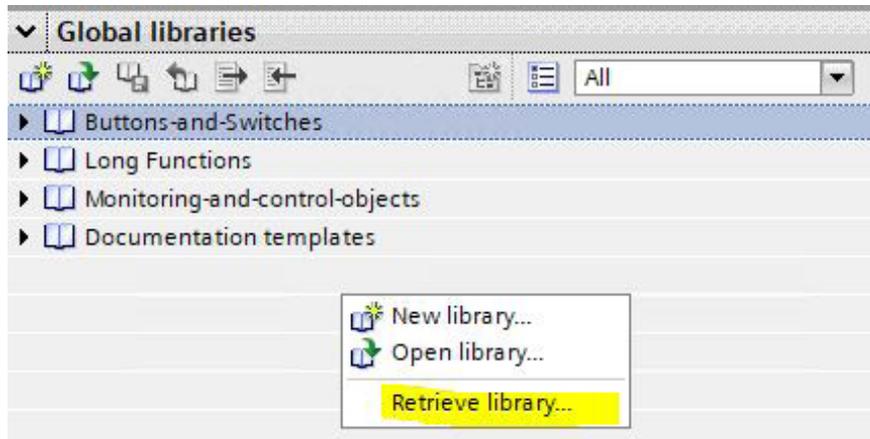
The short identifier is **EKSDataService**. PLC blocks provided in the EKSDatService library for S7-300, S7-400, S7-1200 and S7-1500 are used to set up a connection from the PLC to the service and to request the Electronic-Key data of the placed Electronic-Key.



## 4. S7-1200 / S7-1500 TIA Portal

### 4.1. Integrating library into TIA Portal

You must retrieve the library in order to use it in your project. Right-click in the “Global libraries” area and select *Retrieve library...*



Then navigate to and select the library `PLC_Library\TIA_1200_1500\EKS_Data_Service_PLC_Library_TIA_1200_1500_20190227.zal14`. Subsequently enter the storage location of the retrieved library. If you have a version higher than TIA Portal V14 SP1, you will be asked whether you wish to upgrade the library (this process has been tested successfully with TIA Portal V15). The library will then be available for selection from the *Global libraries* area.

### 4.2. EKSDatasevice library

The EKSDatasevice library contains two folders in the copy templates.

1. EKSDataseviceBase
2. EKSDataseviceExample

#### 4.2.1. EKSDataseviceBase

This folder contains the blocks and data types required for EKS Data Service PLC.

##### FB\_EKSDatasevice block

The FB\_EKSDatasevice block is responsible for exchanging data with EKS Data Service PC and providing the data in the PLC.

##### DB\_EKSDatasevice1 block

The DB\_EKSDatasevice1 data block is an example of the structure for the data required for FB\_EKSDatasevice.

##### Data types (UDTs) type\_EKSxxxx

These data types are required for the declaration of block parameters and data types in the DB.

- type\_EKSDatabase
- type\_EKSDataseviceID
- type\_EKSDescription
- type\_EKSKeyID

### 4.2.2. EKSDataServiceExample

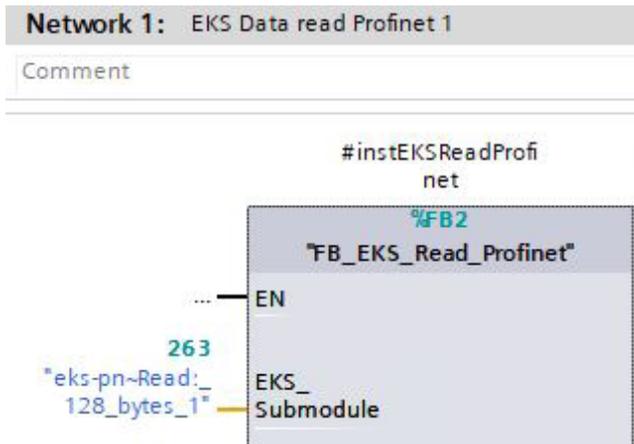
This folder contains a complete example of how to use EKS Data Service PLC. Two EKS readers (PROFINET, PROFIBUS) are used. One instance of FB\_EKSDataService is created for each EKS. The data requested by FB\_EKSDataService are stored in a DB\_EKSData1 data block.

#### EKS\_Euchner block

The block is called in the Main (OB1) program, and it administers all Euchner blocks in connection with EKS Data Service PLC.

The example uses hardware addresses that are created in the device configuration with the declaration of the readers. The correct addresses can be obtained after hardware configuration in the configuration section.

#### Address reference FB\_EKS\_Read\_Profinet



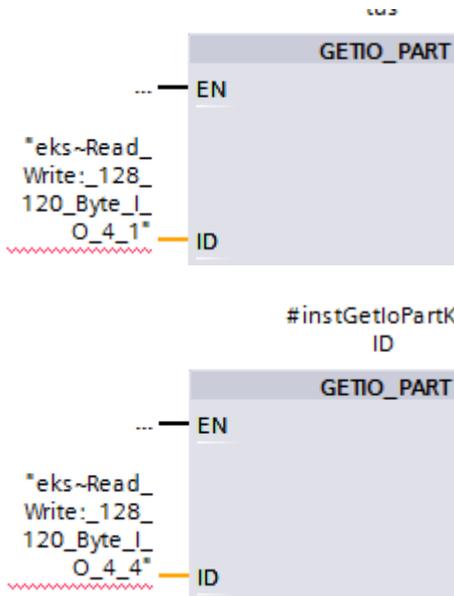
Network overview				Connections	I/O communication	VPN
Network view	Device		Type	Address in subnet		
	▼ S71500/ET200MP station_1	S71500/ET200MP station				
	▶ PLC_1	CPU 1516F-3 PN/DP				
	▼ GSD device_1	GSD device				
	▶ eks-pn	EKS-A-IIXA-G01-ST02/I0...				
	▼ GSD device_2	GSD device				
	eks	EKS-A-IDX-G01-ST09/O3		3		

GSD device_1 [Device]			
General	IO tags	System constants	Texts
Show hardware system constant ▾			
	Name	Type	Hardware identi
<input checked="" type="checkbox"/>	eks-pn-IODevice	Hw_Device	262
<input checked="" type="checkbox"/>	eks-pn-Interface~Port_1	Hw_Interface	260
<input checked="" type="checkbox"/>	eks-pn-Interface	Hw_Interface	259
<input checked="" type="checkbox"/>	eks-pn-Proxy	Hw_SubModule	258
<input checked="" type="checkbox"/>	eks-pn-Head	Hw_SubModule	261
<input checked="" type="checkbox"/>	eks-pn-Read:_128_bytes_1	Hw_SubModule	263
<input checked="" type="checkbox"/>	eks-pn-Write:_128_bytes_1	Hw_SubModule	265

## Address reference FB\_EKS\_Read\_Profibus

If an EKS Profibus is used, the library contains an example block (FB\_EKS\_Read\_Profibus) to read out a Profibus EKS. The input address range of the Profibus EKS must be configured to configure the block.



Network overview		Connections	I/O communication	VPN	TeleCont
Device	Type	Address in subnet	Subnet		
▶ GSD-Geraet_1	GSD device				
▶ GSD-Geraet_2	GSD device				
▶ S71500/ET200MP-Station_1	S71500/ET200MP station				
▼ GSD device_1	GSD device				
eks	EKS-AIDX-G01-ST09/03	3			PROFIBUS_1

D device_1 [Device]			
General	IO tags	System constants	Texts
how hardware system constant			
Name	Type	Hardware identi.	
eks~DPSlave	Hw_DpSlave	263	
eks~Head	Hw_Interface	265	
eks~Read_Write:_128_120_Byte_I_O_4_2	Hw_SubModule	267	
eks~Read_Write:_128_120_Byte_I_O_4_3	Hw_SubModule	268	
eks~Read_Write:_128_120_Byte_I_O_4_4	Hw_SubModule	269	
eks~Read_Write:_128_120_Byte_I_O_4_1	Hw_SubModule	266	

### 4.3. EKSDataService block description

The control system requires two blocks to work with EKS Data Service PC.

The first block is an FB (FB\_EKSDataService) in which all functions are programmed. The second block is a DB (DB\_EKSData1) containing the data for the emergency level, the current data for the placed Electronic-Key and the data declaration.

#### 4.3.1. DB\_EKSData1

All data stored here can also be distributed to other data blocks. All required data are combined in one DB in the template.

#### 4.3.2. Data of placed Electronic-Keys (type\_EKSDatabase)

Data type *type\_EKSDatabase* describes the structure of the data requested from the database.

All Electronic-Key data required in the PLC by the EKM database must be defined in a structure here. The structure must match the description *type\_EKSDescription* (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

type_EKSDatabase			
	Name	Data type	Default value
1	KeyID	String[16]	"
2	LOCKED	Bool	false
3	Key_Bit	Bool	false
4	Key_ShortInt	SInt	0
5	Key_Byte	Byte	16#0
6	Key_SmallInt	Int	0
7	Key_Word	Word	16#0
8	Key_Integer	DInt	0
9	Key_Float	LReal	0.0
10	Key_String	String[12]	"
11	Key_StringBlankFilled	String[10]	"
12	Key_Time	Time	T#0ms
13	Key_TimeAscii	String[8]	"
14	Key_Date	Date	D#1990-01-01

A separate database for saving the data of the currently placed Electronic-Key can be created in the DB for each EKS.

DB_EKSDData1	
Name	Data type
Static	
Data_Key_Reader_Profinet1	"type_EKSDatabase"
Data_Key_Reader_Profinet2	"type_EKSDatabase"

### Description of the requested data (type\_EKSDescription)

The data are requested from EKS Data Service PC. This request is defined via a structure corresponding to the structure of type `type_EKSDatabase` (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

type_EKSDatabase			
	Name	Data type	Default value
1	KeyID	String[16]	"
2	LOCKED	Bool	false
3	Key_Bit	Bool	false
4	Key_Shortint	SInt	0
5	Key_Byte	Byte	16#0
6	Key_Smalint	Int	0
7	Key_Word	Word	16#0
8	Key_Integer	DInt	0
9	Key_Float	LReal	0.0
10	Key_String	String[12]	"
11	Key_StringBlankFilled	String[10]	"
12	Key_Time	Time	T#0ms
13	Key_TimeAscii	String[8]	"
14	Key_Date	Date	D#1990-01-01

The default values are important and can be set in EKS Data Service PC.

The description of the type *EKS Description* is assigned to a variable one time in the DB:

DB_EKSDData1	
Name	Data type
Static	
Data_Key_Reader_Profinet1	"type_EKSDatabase"
Data_Key_Reader_Profinet2	"type_EKSDatabase"
Description	"type_EKSDescription"

### ID array for administering the data (type\_EKSDataID)

The ID is required to save additional information such as the time stamp for Electronic-Key placement or the time stamp of the last refresh operation, the KeyID and the checksum of EKS Data Service. This additional information is entered and stored in the DB for each Electronic-Key placed.

This data type must not be changed and is saved as an array in the DB.

An ID array of the same size is required for each data saving operation (data array).

Since the data are required after a power failure/restart as well, they must be marked as remanent.



#### Notice about the checksum

EKS Data Service uses the requested data to calculate a checksum (also called "CRC" for short below) to ensure the integrity of the requested data. These data are not processed by user. This checksum must not be confused with the checksum calculated by the Electronic Key-Manager EKM software. EKM's checksum (also called "Key CRC" for short below) always refers to a certain data range that can be configured in EKM. EKM's CRC is additionally stored on the Electronic-Key and can be used exclusively on "On-Key" fields. By contrast, EKS Data Service calculates the CRC over all requested data (both "On-Key" and EKM database values).

type_EKSDataID	
Name	Data type
timestampPlaced	DTL
timestampRefresh	DTL
KeyID	*type_EKSKeyID*
CRC	Word

Example from the library:

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profinet1	*type_EKSDatabase*
Data_Key_Reader_Profinet2	*type_EKSDatabase*
Description	*type_EKSDescription*
ID	Array[0..5] of *type_EKSData...

### Data array for saving the emergency mode data (type\_EKSDatabase)

The same data type for data of the placed Electronic-Keys is also used for saving the data for the emergency level. All data from the placed Electronic-Keys are entered into the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed in the EKS Data Service PC request, the Electronic-Key will be deleted in the data array as well. All instances of FB\_EKSDataService can access the same data (Description, ID, Data) in DB\_EKSData1. If the data are to be available after a power failure as well, they must be marked as remanent.

Example from the library:

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profinet1	*type_EKSDatabase*
Data_Key_Reader_Profinet2	*type_EKSDatabase*
Description	*type_EKSDescription*
ID	Array[0..5] of *type_EKSDataID*
Data	Array[0..5] of *type_EKSDatabase*

### 4.3.3. FB\_EKSDataService

FB\_EKSDataService for the S7-1200 / S7-1500 possesses the following interface description.

INPUT		
Name	Data type	Description
Key_Present	Bool	Electronic-Key is placed
Key_Data_updated	Bool	Electronic-Key data are updated on placement
Interface_ID	HW_ANY	Profinet interface hardware ID
Con_ID	CONN_OUC	Unique connection ID
Timeout_Time	Time	Communication timeout
Refresh_Time	Time	Refresh time for Electronic-Key data
IP_EKSDataService	String[20]	IP address of EKS Data Service
Local_TSAP	String[32]	TSAP of local connection
Remote_TSAP	String[32]	TSAP of EKS Data Service
OUTPUT		
Name	Data type	Description
Con_Parametrization_Errorcode	Word	Communication parametrization error
Con_Communication_Errorcode	Word	Server service communication error
DB_Error	Bool	Database length wrong
Data_Error	Bool	Data error from server service
Emergency_Level_1	Bool	Emergency level 1 active
Emergency_Level_2	Bool	Emergency level 2 active
KEY_Not_Found	Bool	Electronic-Key ID not found
KEY_Checksum_Error	Bool	CRC error (of requested data)
KEY_Valid	Bool	Electronic-Key placed and data available
Error	Bool	Block error
Status	Word	Block status
serviceStatus	Word	Status of server service
INOUT		
Name	Data type	Description
Key_ID	type_EKSKeyID	Serial number of placed Electronic-Key
EKS_Data_Key_Reader	Variant	Data storage location for the placed Electronic-Key
EKS_Data_Description	Variant	Data of the description
EKS_Data	Variant	Data array of PLC data
EKS_KEY_ID	Variant	ID data for the data structure
EKS_Data_Temp	Variant	Data memory for buffering



#### Notice

FB\_EKSDataService is protected by a password to prevent tampering. In some cases, FB\_EKSDataService must be recompiled if you have incorporated it into your project. You require a password to compile the block. Please do not modify the implementation of FB\_EKSDataService.

Password: EKS1234

**Key\_Present**

When the EKS reader reads the placed Electronic-Key, this produces a signal confirming the read Electronic-Key serial number. This signal is required for requesting the Electronic-Key data. A data request is sent to EKS Data Service PC when Key\_Present changes from FALSE to TRUE.

**Key\_Data\_updated**

If TRUE is assigned to the parameter, the Electronic-Key data of the placed Electronic-Key are always updated with the database. This means that the data can change during placement.

Example:

An Electronic-Key is placed in EKS, the requested data are already saved in the PLC (DB\_EKSData1.Data\_Key\_Reader\_Profinet) and data refresh (Refresh\_Time) is activated. In other words, the data of the emergency level are cyclically updated in the background. If the updated value of the placed Electronic-Key differs from the saved value, the value of the currently placed Electronic-Key (DB\_EKSData1.Data\_Key\_Reader\_Profinet) will be overwritten under Key\_Data\_updated = TRUE. Otherwise, the updated value will be loaded only when the Electronic-Key is placed again.

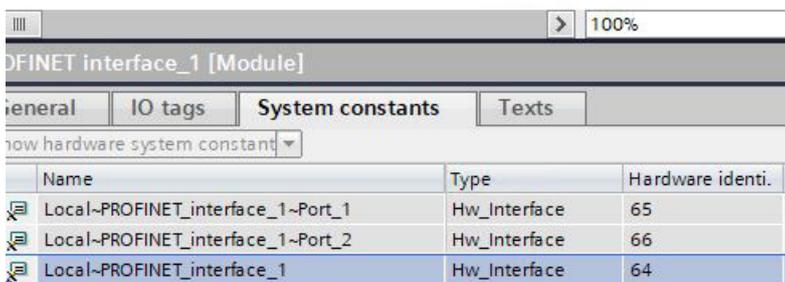
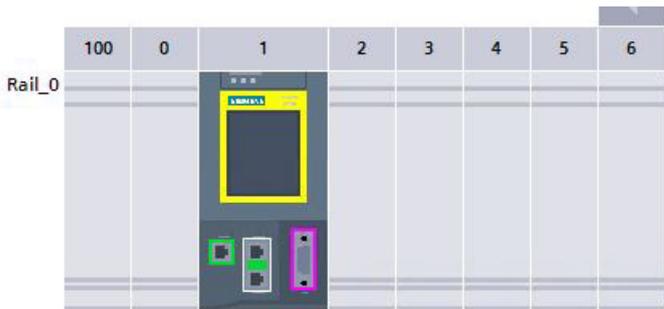
**Interface\_ID**

Hardware ID of the local interface via which communication with EKS Data Service PC is to take place.



**Notice**

Several FB\_EKSDataService instances can communicate via the same interface.



**Con\_ID**

The connection ID must be unambiguous for each connection within a PLC. This is also necessary if several instances are used.

Example:

Instance A = 1

Instance B = 2



## Timeout\_Time

If a timeout occurs during communication with EKS Data Service PC, an error is generated that cancels communication and issues an error with status. In the event of a fault, emergency level 2 is triggered and the requested data are loaded from the emergency-level memory (DB\_EKSDData1.Data).

#200ms — Timeout\_Time

## Refresh\_Time

In this time cycle, EKS Data Service PC retrieves the data of the listed Electronic-Keys in DB\_EKSDData1.Data and checks whether they are up to date. Otherwise, the values in DB\_EKSDData1.Data will be updated. In other words, already requested data will be updated in the background without Electronic-Key placement. The block moves on to the next entry after every cycle and requests the data from EKS Data Service PC. With a setting of 10 s, the service requires 100 s to request 10 stored items of Electronic-Key data.

Data refresh can be switched off using the value 0 ms. If several FB\_EKSDDataService instances are required and if all instances access the same data, only one instance must perform the refresh operation.

Example:

Instance A: Refresh\_Time = 10 s (responsible for refresh)

Instance B: Refresh\_Time = 0 ms (refresh switched off)

Instance C: Refresh\_Time = 0 ms (refresh switched off)

#10s — Refresh\_Time

## IP\_EKSDDataService

The IP address of EKS Data Service PC is entered here.

'192.168.12.101' — IP\_EKSDDataService

The IP address is entered as a string enclosed in single quotation marks. ( '192.168.12.101' )

## Local\_TSAP

The local TSAP address is entered here. This is used to identify communication and is displayed in EKS Data Service PC. Different descriptions must be used for several instances. This is also useful for debugging in EKS Data Service PC; the name is used for the log entry here.

'EKSCIENT1500\_Profinet' — Local\_TSAP Example for the Profinet reader

'EKSCIENT\_1500\_Profibus' — Local\_TSAP or Profibus reader

The TSAP is entered as a string enclosed in single quotation marks. ( 'EKSCIENT1500\_Profinet' )

## Remote\_TSAP

Remote TSAP is a fixed address for EKS Data Service PC. It must always be entered as follows:

'EKSSERV' — Remote\_TSAP

The TSAP is entered as a string enclosed in single quotation marks. ( 'EKSSERV' )

### **Con\_Parametrization\_Errorcode**

If an error is generated during connection setup, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.1. *Connection setup (Con\_Parametrization\_Errorcode)*.

### **Con\_Communication\_Errorcode**

If an error is generated during data reception, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.2. *Data reception (Con\_Communication\_Errorcode)*.

### **DB\_Error**

The error bit becomes TRUE if the length of structure *type\_EKSDatabase* does not match the sent data size from EKS Data Service PC.

### **Data\_Error**

EKS Data Service PC outputs the *Data\_Error* error if the specified data structure does not match the structure of EKS Data Service PC. If EKS Data Service PC reports this error, *Data\_Error* will be set to TRUE.

### **Emergency\_Level\_1**

Emergency level 1: EKS Data Service PC cannot access the original EKM CSV export file and instead uses the local backup file of the EKM CSV export file.

### **Emergency\_Level\_2**

Emergency level 2: EKS Data Service PC cannot provide any data because communication is disrupted. The data of the internal data block (*DB\_EKSData1.Daten*) are used instead.

### **KEY\_Not\_Found**

The placed Electronic-Key was not found. This information can originate from EKS Data Service PC, or from the PLC in case of emergency level 2 if the requested Electronic-Key is not stored in the memory (*DB\_EKSData1.Data*).

### **KEY\_Checksum\_Error**

The CRC calculated via the requested data does not match. This error is output if the CRC of the sent data does not match, or if the CRC of the saved data is incorrect in case of emergency level 2.

### **KEY\_Valid**

The data of the placed Electronic-Key are available. This also applies to the emergency levels.

### **Error**

An error occurred during block processing. The error will be deleted during the next processing operation and then possibly output again.

### **Status**

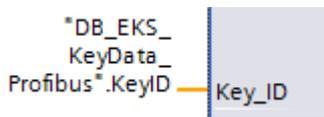
Status messages that are generated as part of processing in *FB\_EKSDataService*. A list of status messages can be found in the chapter 7.3. *Status of FB\_EKSDataService (Status)*.

### **serviceStatus**

Status codes that are transmitted from EKS Data Service PC to *FB\_EKSDataService*. A list of status messages can be found in the chapter 7.4. *Data request status (serviceStatus)*.

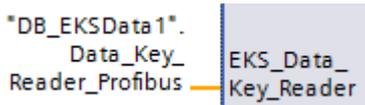
## Key\_ID

The serial number of the placed Electronic-Key must be specified here to request the data from EKS Data Service PC. Data type *type\_EKSKeyID* must be used for this purpose.



## EKS\_Data\_Key\_Reader

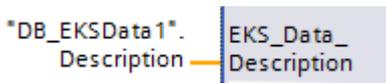
The data of the placed Electronic-Key are copied to this address. Data type *type\_EKSDatabase* must be used.



▶ Data_Key_Reader_Profinet	"type_EKSDatabase"
▶ Data	Array[0..100] of "type_EKSDatabase"

## EKS\_Data\_Description

The data description is transmitted as the data type variant. This data structure is used to request the required data from EKS Data Service PC. The data structure of the data type *type\_EKSDescription* must be used.



▶ Description	"type_EKSDescription"
---------------	-----------------------

## EKS\_Data

The data array for data backup on the PLC is created as the data type variant (see data `DB_EKSData1`).



▶ Data	Array[0..100] of "type_EKSDatabase"
--------	-------------------------------------

The data array must be the same size as `EKS_KEY_ID` Array.

## EKS\_KEY\_ID

The ID range is available for administering data backup (see ID `DB_EKSData1`).



▶ ID	Array[0..100] of "type_EKSDataID"
------	-----------------------------------

The data array must be the same size as `EKS_Data` Array.

### EKS\_Data\_Temp

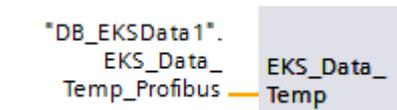
The block requires a buffer outside the block of the type `type_EKSDatabase`. The buffer is used for copying tasks. The structure is transferred as a data type variant and, depending on application, can have a different content. However, the internal copying processes can copy only variants and require a memory location with the same structure.

The memory location can also be created in DB\_EKSData1. A memory location for every instance is important here.

Profinet instance:



Profibus instance:



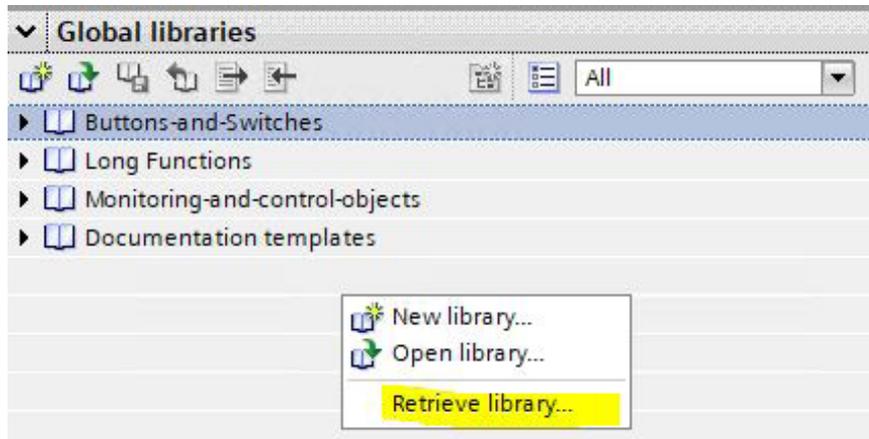
Storage location in DB\_EKSData1:

		EKS_Data_Temp_Profinet	"type_EKSDatabase"
		EKS_Data_Temp_Profibus	"type_EKSDatabase"

## 5. S7-300 / S7-400 TIA Portal

### 5.1. Integrating library into TIA Portal

You must retrieve the library in order to use it in your project. Right-click in the “Global libraries” area and select *Retrieve library...*



Subsequently navigate to and select the library (PLC\_Library\TIA\_300\_400\EKS\_Data\_Service\_PLC\_Library\_TIA\_300\_400\_20190503.zal14). Subsequently enter the storage location of the retrieved library. If you have a version higher than TIA Portal V14 SP1, you will be asked whether you wish to upgrade the library (this process has been tested successfully with TIA Portal V15). The library will then be available for selection from the *Global libraries* area.

### 5.2. EKSDataService library

The EKSDataService library contains all blocks for the CPU 300 in the S7-300 folder and all blocks for the CPU 400 in the S7-400 folder.

#### 5.2.1. Standard Siemens blocks used

Various blocks from the Siemens library are used for implementation.

- AG\_SEND (AG\_LSEND)
- AG\_RECV (AG\_LRECV)
- EQ\_STRNG
- AG\_CNTRL
- GT\_DT
- MID
- NE\_STRNG
- REPLACE
- STRNG\_I

These blocks are essential, and they should not be changed in the numbering (exception in case of symbolic priority). The CPU 300 and CPU 400 blocks have different functions. The correct blocks of the CPU must be used.

Example – block family:

CPU 300

**Information**

Title:	<input type="text"/>
Comment:	<input type="text"/>
Version:	1.4
Family:	CP_300
Author:	SIMATIC
User-defined ID:	AG_CNTRL

CPU 400

**Information**

Title:	<input type="text"/>
Comment:	<input type="text"/>
Version:	1.0
Family:	CP_400
Author:	SIMATIC
User-defined ID:	AG_CNTRL

### 5.2.2. FB\_EKSDataService block

The block (FB\_EKSDataService) is responsible for exchanging data with EKS Data Service PC and providing the data in the PLC.

### 5.2.3. DB\_EKSData1 block

The (DB\_EKSData1) data block is an example of the structure for the data required for FB\_EKSDataService.

### 5.2.4. Data types type\_EKSxxxx

The data type UDTs are required for the declaration of block parameters and data types in the DB.

- type\_EKSDatabase
- type\_EKSDatalID
- type\_EKSDescription
- type\_EKSKeyID

### 5.2.5. FC\_EKS\_Euchner block

The block is called in the Main (OB1) program, and it administers all Euchner blocks in connection with EKS Data Service PC.

The example uses hardware addresses that are created in the device configuration with the declaration of the EKS readers. The correct addresses can be obtained after hardware configuration in the configuration section.

## Address reference EKS\_Euchner (Profibus)

### Network 1: Read EKS Profibus (Status information)

Comment



### Network 2: Read EKS Profibus (KeyID)

Comment



Module	Rack	Slot	I address	Q address	Ty
EKS-Slave	0	0	2043*		El
Read/Write: 128/120 Byte I/...	0	1	256...287	256...287	Re
Read/Write: 128/120 Byte I/...	0	2	288...319	288...319	Re
Read/Write: 128/120 Byte I/...	0	3	320...351	320...351	Re
Read/Write: 128/120 Byte I/...	0	4	352...383	352...375	Re



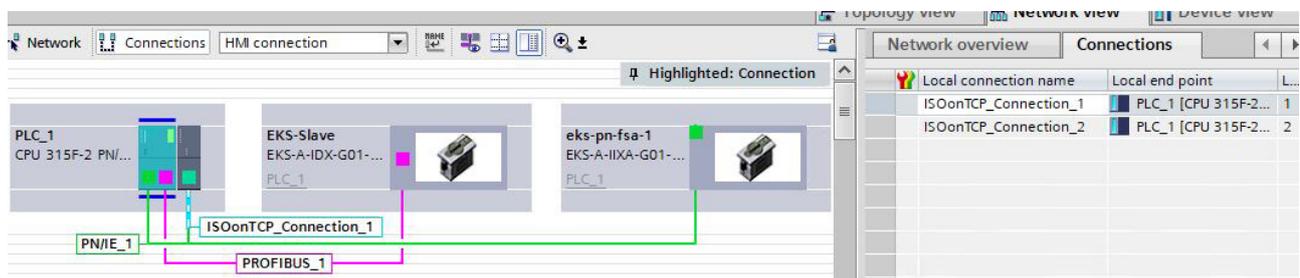
### Notice

The E-address must be specified to LADDR as a HEX value

### 5.3. Settings (TCP/IP configuration)

#### 5.3.1. Setting up connection in the CPU

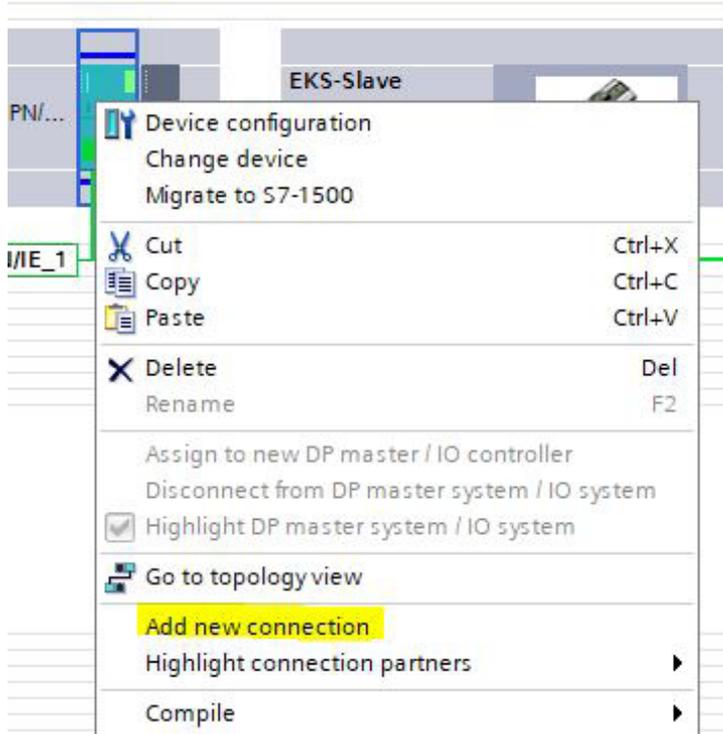
A connection to EKS Data Service PC must be set up in the CPU for each instance of the FB\_EKSDataService block.



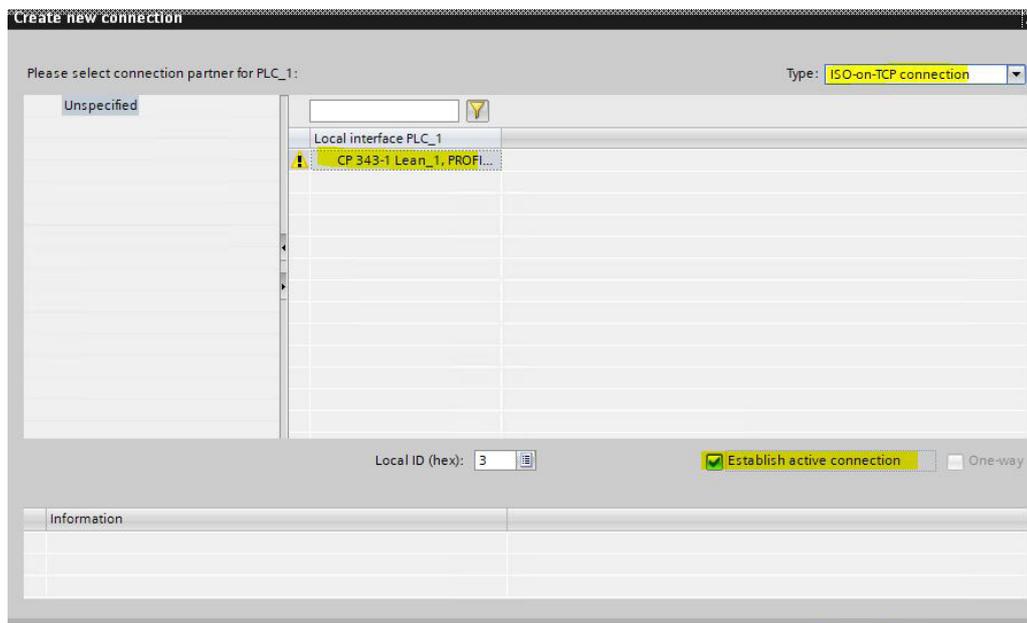
Network overview					
Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner	Connection type
ISOonTCP_Connection_1	PLC_1 [CPU 315F-2...	1		Unknown	ISO-on-TCP connection
ISOonTCP_Connection_2	PLC_1 [CPU 315F-2...	2		Unknown	ISO-on-TCP connection

## Adding a new connection

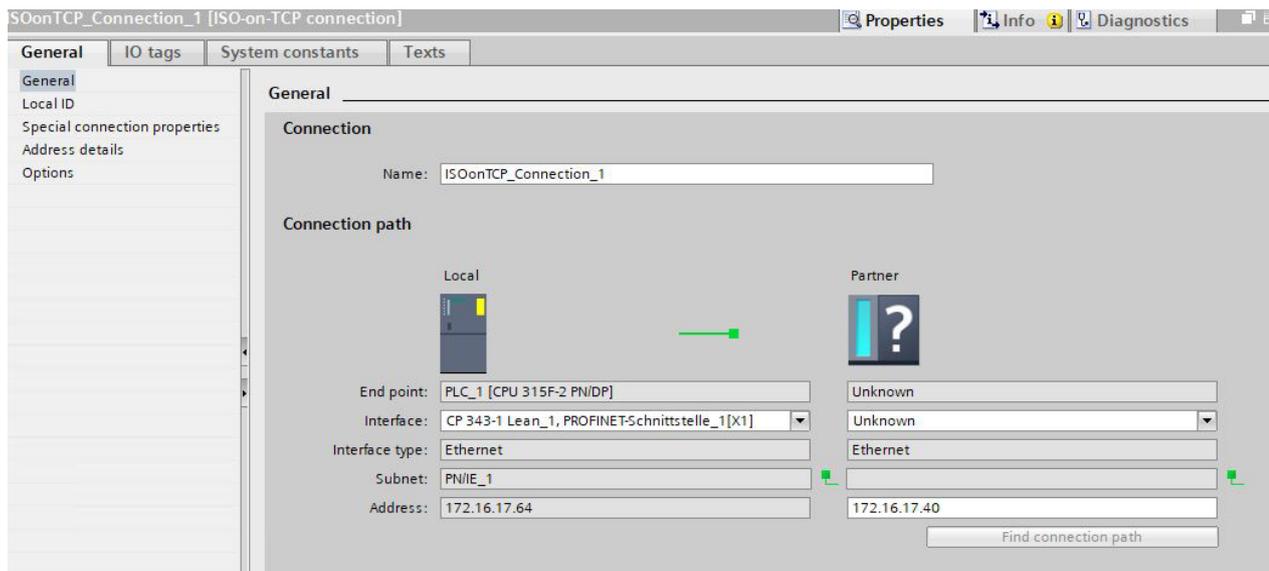
Select the *Connections* view in the *Devices & Networks* menu and then right-click the CPU and select *Add new connection*.



Select *ISO-on-TCP connection type* and *Establish active connection* and click *Add*.



## CPU 300 connection and block parameters



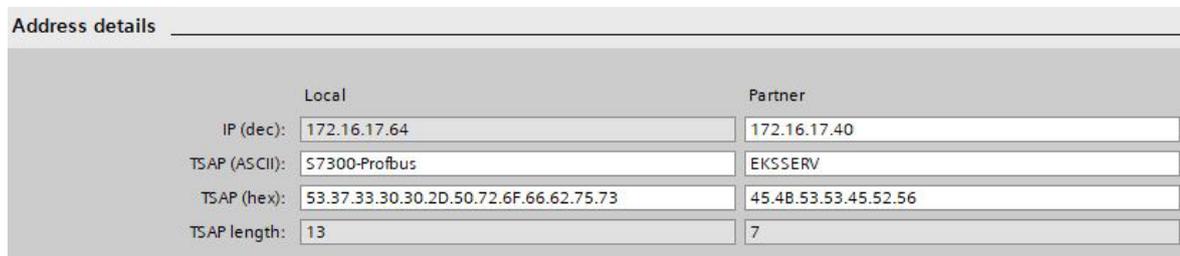
- › The name of the connection can be changed.
- › The active connection setup must be marked (special connection properties).
- › Block parameters are important for programming.

### Addresses

The TSAP must be set here.

The “Local TSAP” is used to identify communication and is displayed in EKS Data Service PC. Different descriptions must be used for several instances. This is also useful for debugging in EKS Data Service PC; the name is used for the log entry here.

The IP address of EKS Data Service PC must be entered as the partner. The partner’s TSAP must be *EKSSERV*.



## 5.4. EKSDataService block description

The control system requires two blocks to work with EKS Data Service PC.

The first block is an FB (FB\_EKSDataService) in which all functions are programmed. The second block is a DB (DB\_EKSData1) containing the data for the emergency level, the current data for the placed Electronic-Key and the data declaration.

### 5.4.1. DB\_EKSData1

All data stored here can also be distributed to other data blocks. All required data are combined in one DB in the template.

#### Data of placed Electronic-Keys (type\_EKSDatabase)

Data type *type\_EKSDatabase* describes the structure of the data requested from the database.

All data required in the PLC by the database can be defined in a structure here. The structure must match the description *type\_EKSDescription* (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

type_EKSDatabase		
Name	Data type	Default value
KeyID	String[16]	''
LOCKED	Bool	false
Key_Bit	Bool	false
Key_Shortint	SInt	0
Key_Byte	Byte	16#0
Key_SmallInt	Int	0
Key_Word	Word	16#0
Key_Integer	DInt	0
Key_Float	LReal	0.0
Key_String	String[12]	''
Key_StringBlankFilled	String[10]	''
Key_Time	Time	T#0ms
Key_TimeAscii	String[8]	''
Key_Date	Date	D#1990-01-01

A separate database for saving the data of the currently placed Electronic-Key can be created in the DB for each EKS.

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profibus	*type_EKSDatabase*

### Description of the requested data (type\_EKSDescription)

The data are requested from EKS Data Service PC. This request is defined via a structure corresponding to the structure of type `type_EKSDatabase` (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

Name	Data type	Default value
KeyID	String[16]	''
LOCKED	Bool	false
Key_Bit	Bool	false
Key_ShortInt	SInt	0
Key_Byte	Byte	16#0
Key_SmallInt	Int	0
Key_Word	Word	16#0
Key_Integer	DInt	0
Key_Float	LReal	0.0
Key_String	String[12]	''
Key_StringBlankFilled	String[10]	''
Key_Time	Time	T#0ms
Key_TimeAscii	String[8]	''
Key_Date	Date	D#1990-01-01

The default values are important and can be set in EKS Data Service PC.

The description of the type *EKS Description* is assigned to a variable one time in the DB:

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profinet1	*type_EKSDatabase*
Data_Key_Reader_Profinet2	*type_EKSDatabase*
Description	*type_EKSDescription*

## ID array for administering the data (type\_EKSDataID)

The ID is required to store additional information such as the time stamp for Electronic-Key placement or the time stamp of the last update, the KeyID and the checksum. This additional information is entered and stored in the DB for each Electronic-Key placed.

This data type must not be changed and is saved as an array in the DB.

An ID array of the same size is required for each data saving operation (Data Array).

Since the data are required after a power failure/restart as well, they must be marked as remanent.



### Notice about the checksum

EKS Data Service uses the requested data to calculate a checksum (also called "CRC" for short below) to ensure the integrity of the requested data. These data are not processed by user. This checksum must not be confused with the checksum calculated by the Electronic Key-Manager EKM software. EKM's checksum (also called "Key CRC" for short below) always refers to a certain data range that can be configured in EKM. EKM's CRC is additionally stored on the Electronic-Key and can be used exclusively on "On-Key" fields. By contrast, EKS Data Service calculates the CRC over all requested data (both "On-Key" and EKM database values).

type_EKSDataID				
Name	Data type	Offset	Default value	
timestampPlaced	Date_And_Time	0.0	DT# 1990-01-01-0	
timestampRefresh	Date_And_Time	8.0	DT# 1990-01-01-0	
KeyID	*type_EKSKeyID*	16.0		
CRC	Word	24.0	16#0	

Example from the library:

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profibus	*type_EKSDatabase*
Data_Key_Reader_Profinet	*type_EKSDatabase*
Description	*type_EKSDescription*

**Data Array for saving the emergency mode data (type\_EKSDatabase)**

The same data type for data of the placed Electronic-Keys is also used for saving the data for the emergency level. All data from the placed Electronic-Keys are entered into the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed in the EKS Data Service PC request, the Electronic-Key will be deleted in the data array as well. All instances of FB\_EKSDataService can access the same data (Description, ID, Data) in DB\_EKSData1. If the data are to be available after a power failure as well, they must be marked as remanent.

Example from the library:

DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profibus	"type_EKSDatabase"
Data_Key_Reader_Profinet	"type_EKSDatabase"
Description	"type_EKSDescription"
ID	Array[0..5] of "type_EKSDataID"
Data	Array[0..5] of "type_EKSDatabase"

## 5.4.2. FB\_EKSDataService

FB\_EKSDataService for the S7-300 / S7-400 possesses the following interface description.

INPUT		
Name	Data type	Description
Key_Present	Bool	Electronic-Key is placed
Key_Data_updated	Bool	Electronic-Key data are updated on placement
Interface_ID	Word	Profinet interface hardware ID
Con_ID	Int	Unique connection ID
Timeout_Time	Time	Communication timeout
Refresh_Time	Time	Refresh time for Electronic-Key data

OUTPUT		
Name	Data type	Description
Con_Parametrization_Error	Word	Communication parametrization error
Con_Communication_Error	Word	Server service communication error
DB_Error	Bool	Database length wrong
Data_Error	Bool	Data error from server service
Emergency_Level_1	Bool	Emergency level 1 active
Emergency_Level_2	Bool	Emergency level 2 active
KEY_Not_Found	Bool	Electronic-Key ID not found
KEY_Checksum_Error	Bool	CRC error (of requested data)
KEY_Valid	Bool	Electronic-Key placed and data available
Error	Bool	Block error
Status	Word	Block status
serviceStatus	Word	Status of server service

INOUT		
Name	Data type	Description
Key_ID	type_EKSkeyID	Serial number of placed Electronic-Key
EKS_Data_Key_Reader	Any	Data storage location for the placed Electronic-Key
EKS_Data_Description	Any	Data of the description
EKS_Data	Any	Data array of PLC data
EKS_KEY_ID	Any	ID data for the data structure



### Notice

FB\_EKSDataService is protected by a password to prevent tampering. In some cases, FB\_EKSDataService must be recompiled if you have incorporated it into your project. You require a password to compile the block. Please do not modify the implementation of FB\_EKSDataService.

Password: *EKS1234*

### Key\_Present

When the EKS reader reads the placed Electronic-Key, this produces a signal confirming the read Electronic-Key serial number. This signal is required for requesting the Electronic-Key data. A data request is sent to EKS Data Service PC when Key\_Present changes from FALSE to TRUE.

### Key\_Data\_updated

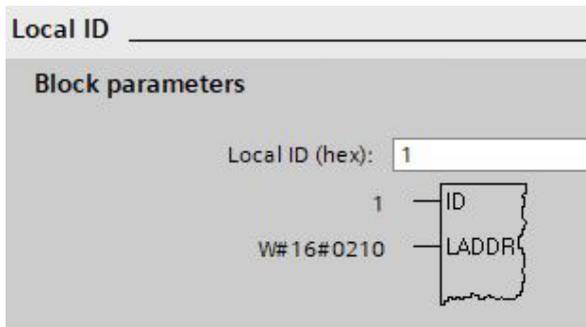
If TRUE is assigned to the parameter, the Electronic-Key data of the placed Electronic-Key are always updated with the database. This means that the data can change during placement.

Example:

An Electronic-Key is placed in EKS, the requested data are already saved in the PLC (DB\_EKSData1.Data\_Key\_Reader\_Profinet) and data refresh (Refresh\_Time) is activated. In other words, the data of the emergency level are cyclically updated in the background. If the updated value of the placed Electronic-Key differs from the stored value, the value of the currently placed Electronic-Key (DB\_EKSData1.Data\_Key\_Reader\_Profinet) will be overwritten under Key\_Data\_updated = TRUE. Otherwise, the updated value will be loaded only when the Electronic-Key is placed again.

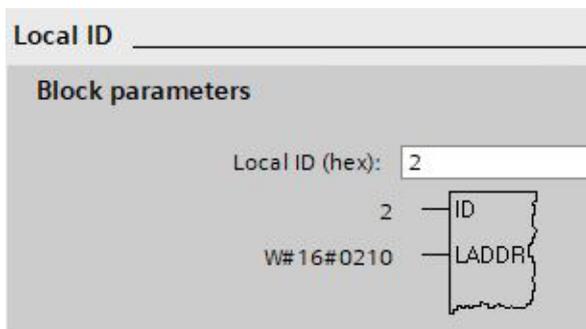
### Interface\_ID

The hardware address is taken from the property window and must be specified on the block.



### Con\_ID

The connection ID is taken from the property window and must be output on the block.



## Timeout\_Time

If a timeout occurs during communication with EKS Data Service PC, an error is generated that cancels communication and issues an error with status. In the event of a fault, emergency level 2 is triggered and the requested data are loaded from the emergency-level memory (DB\_EKSDData1.Data).

T#5S — Timeout\_Time

## Refresh\_Time

In this time cycle, EKS Data Service PC retrieves the data of the listed Electronic-Keys in DB\_EKSDData1.Data and checks whether they are up to date. Otherwise, the values in DB\_EKSDData1.Data will be refreshed. In other words, already requested data will be updated in the background without Electronic-Key placement. The block moves on to the next entry after every cycle and requests the data from EKS Data Service PC. With a setting of 10 s, the service requires 100 s to request 10 stored items of Electronic-Key data.

Data refresh can be switched off using the value 0 ms. If several FB\_EKSDDataService instances are required and if all instances access the same data, only one instance must perform the refresh operation.

Example:

Instance A: Refresh\_Time = 10 s (responsible for refresh)

Instance B: Refresh\_Time = 0 ms (refresh switched off)

Instance C: Refresh\_Time = 0 ms (refresh switched off)

t#10s — Refresh\_Time

### **Con\_Parametrization\_Errorcode**

If an error is generated during connection setup, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.1. *Connection setup (Con\_Parametrization\_Errorcode)*.

### **Con\_Communication\_Errorcode**

If an error is generated during data reception, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.2. *Data reception (Con\_Communication\_Errorcode)*.

### **DB\_Error**

The error bit becomes TRUE if the length of structure *type\_EKSDatabase* does not match the sent data size from EKS Data Service PC.

### **Data\_Error**

EKS Data Service PC outputs the *Data\_Error* error if the specified data structure does not match the structure of EKS Data Service PC. If EKS Data Service PC reports this error, *Data\_Error* will be set to TRUE.

### **Emergency\_Level\_1**

Emergency level 1: EKS Data Service PC cannot access the original EKM CSV export file and instead uses the local backup file of the EKM CSV export file.

### **Emergency\_Level\_2**

Emergency level 2: EKS Data Service PC cannot provide any data because communication is disrupted. The data of the internal data block (*DB\_EKSData1.Daten*) are used instead.

### **KEY\_Not\_Found**

The placed Electronic-Key was not found. This information can originate from EKS Data Service PC, or from the PLC in case of emergency level 2 if the requested Electronic-Key is not stored in the memory (*DB\_EKSData1.Data*).

### **KEY\_Checksum\_Error**

The CRC calculated via the requested data does not match. This error is output if the CRC of the sent data does not match, or if the CRC of the saved data is incorrect in case of emergency level 2.

### **KEY\_Valid**

The data of the placed Electronic-Key are available. This also applies to the emergency levels.

### **Error**

An error occurred during block processing. The error will be deleted during the next processing operation and then possibly output again.

### **Status**

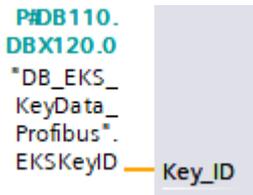
Status messages that are generated as part of processing in *FB\_EKSDataService*. A list of status messages can be found in the chapter 7.3. *Status of FB\_EKSDataService (Status)*.

### **serviceStatus**

Status codes that are transmitted from EKS Data Service PC to *FB\_EKSDataService*. A list of status messages can be found in the chapter 7.4. *Data request status (serviceStatus)*.

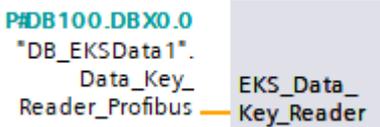
## Key\_ID

The serial number of the placed Electronic-Key must be specified here to request the data from EKS Data Service PC. Data type `type_EKSKeyID` must be used for this purpose.



## EKS\_Data\_Key\_Reader

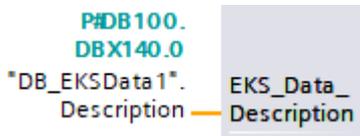
The data of the placed Electronic-Key are copied to this address. Data type `type_EKSDatabase` must be used.



DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profibus	"type_EKSDatabase"
Data	Array[0..5] of "type_EKSDatabase"

## EKS\_Data\_Description

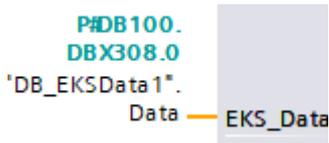
The data description is transmitted as the data type variant. This data structure is used to request the required data from EKS Data Service PC. The data structure of the data type `type_EKSDescription` must be used.



DB_EKSData1	
Name	Data type
Static	
Data_Key_Reader_Profibus	"type_EKSDatabase"
Data_Key_Reader_Profinet	"type_EKSDatabase"
Description	"type_EKSDescription"

### EKS\_Data

The data array for data backup on the PLC is created as a variant (see data DB\_EKSData1).

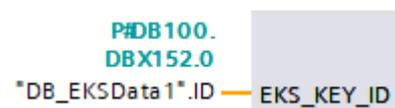


DB_EKSData1	
Name	Data type
Static	
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
▶ Data_Key_Reader_Profinet	"type_EKSDatabase"
▶ Description	"type_EKSDescription"
▶ ID	Array[0..5] of "type_EKSDataID"
▶ Data	Array[0..5] of "type_EKSDatabase"

The data array must be the same size as EKS\_KEY\_ID Array.

### EKS\_KEY\_ID

The ID range is available for administering data backup (see ID DB\_EKSData1).



DB_EKSData1	
Name	Data type
Static	
▶ Data_Key_Reader_Profibus	"type_EKSDatabase"
▶ Data_Key_Reader_Profinet	"type_EKSDatabase"
▶ Description	"type_EKSDescription"
▶ ID	Array[0..5] of "type_EKSDataID"

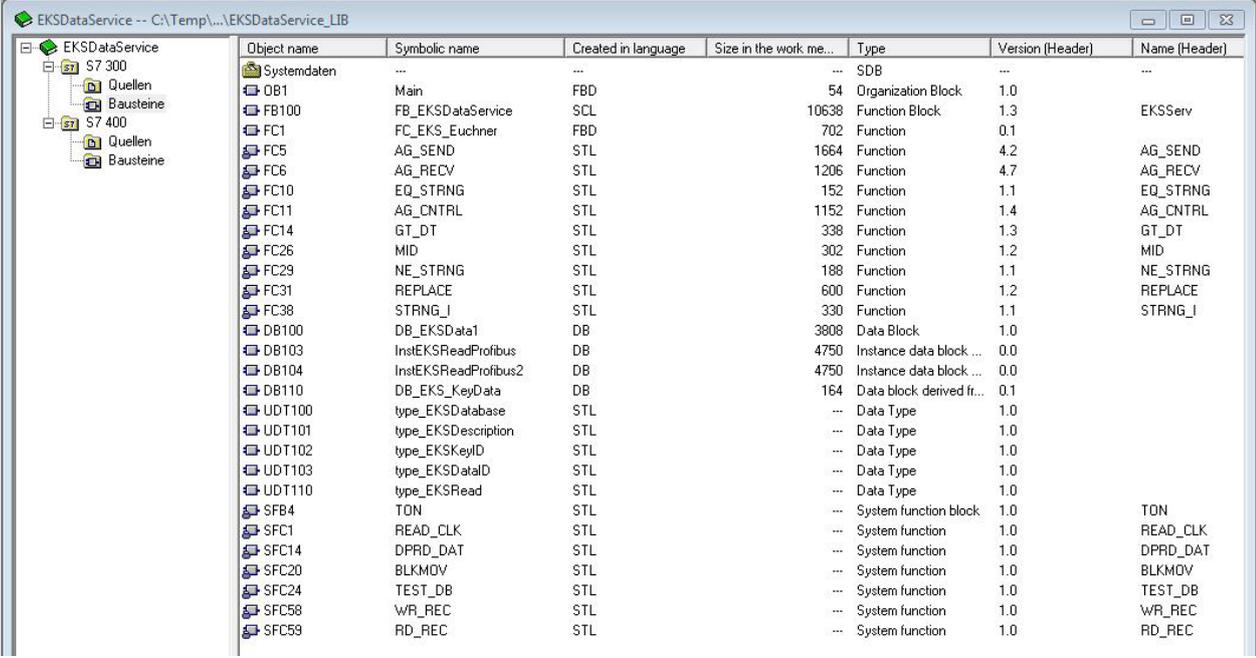
The data array must be the same size as EKS\_Data Array.

## 6. S7-300 / S7-400 SIMATIC STEP7

### 6.1. Integrating library into SIMATIC STEP7

To use the library in SIMATIC STEP7, go to *File | Open...* in SIMATIC Manager. Another window opens. Change to the *Libraries* tab and then click *Search...* Navigate to the library and select it.

This library is found in `\PLC_Library\STEP_7_300_400\EKS_Data_Service_PLC_Library_STEP_7_190503.zip`



Object name	Symbolic name	Created in language	Size in the work me...	Type	Version (Header)	Name (Header)
Systemdaten	...	...	...	SDB	...	...
OB1	Main	FBD	54	Organization Block	1.0	...
FB100	FB_EKSDataService	SCL	10638	Function Block	1.3	EKSServ
FC1	FC_EKS_Euchner	FBD	702	Function	0.1	...
FC5	AG_SEND	STL	1664	Function	4.2	AG_SEND
FC6	AG_RECV	STL	1206	Function	4.7	AG_RECV
FC10	EQ_STRNG	STL	152	Function	1.1	EQ_STRNG
FC11	AG_CNTRL	STL	1152	Function	1.4	AG_CNTRL
FC14	GT_DT	STL	338	Function	1.3	GT_DT
FC26	MID	STL	302	Function	1.2	MID
FC29	NE_STRNG	STL	188	Function	1.1	NE_STRNG
FC31	REPLACE	STL	600	Function	1.2	REPLACE
FC38	STRNG_I	STL	330	Function	1.1	STRNG_I
DB100	DB_EKSData1	DB	3808	Data Block	1.0	...
DB103	InstEKsReadProfibus	DB	4750	Instance data block ...	0.0	...
DB104	InstEKsReadProfibus2	DB	4750	Instance data block ...	0.0	...
DB110	DB_EKS_KeyData	DB	164	Data block derived fr...	0.1	...
UDT100	type_EKSDatabase	STL	...	Data Type	1.0	...
UDT101	type_EKSDescription	STL	...	Data Type	1.0	...
UDT102	type_EKSKeyID	STL	...	Data Type	1.0	...
UDT103	type_EKSDatalD	STL	...	Data Type	1.0	...
UDT110	type_EKSRead	STL	...	Data Type	1.0	...
SFB4	TON	STL	...	System function block	1.0	TON
SFC1	READ_CLK	STL	...	System function	1.0	READ_CLK
SFC14	DPRD_DAT	STL	...	System function	1.0	DPRD_DAT
SFC20	BLKMOV	STL	...	System function	1.0	BLKMOV
SFC24	TEST_DB	STL	...	System function	1.0	TEST_DB
SFC58	WR_REC	STL	...	System function	1.0	WR_REC
SFC59	RD_REC	STL	...	System function	1.0	RD_REC

### 6.2. EKSDataService library

The library (EKSDataService) contains all blocks for the CPU 300 in the S7-300 folder and all blocks for the CPU 400 in the S7-400 folder.

#### 6.2.1. Standard Siemens blocks used

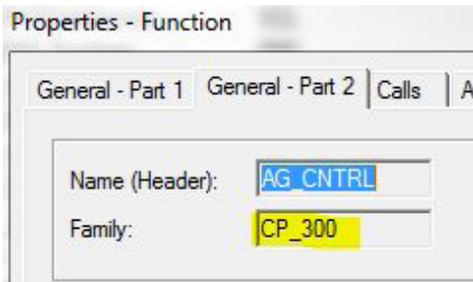
Various blocks from the Siemens library are used for implementation.

- AG\_SEND (AG\_LSEND)
- AG\_RECV (AG\_LRECV)
- EQ\_STRNG
- AG\_CNTRL
- GT\_DT
- MID
- NE\_STRNG
- REPLACE
- STRNG\_I

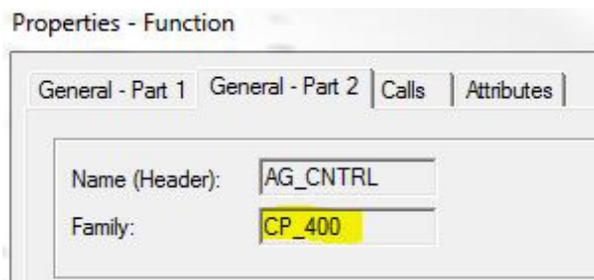
These blocks are essential, and they should not be changed in the numbering (exception in case of symbolic priority). The CPU 300 and CPU 400 blocks have different functions. The correct blocks of the CPU must be used.

Example – block family:

CPU 300



CPU 400



### 6.2.2. System blocks used

System blocks are used in addition to the library blocks.

- SFB4 TON
- SFC1 READ\_CLK
- SFC14 DPRD\_DAT
- SFC20 BLKMOV
- SFC24 TEST\_DB
- SFC58 WR\_REC
- SFC59 RD\_REC

### 6.2.3. FB\_EKSDataService block

The block (FB\_EKSDataService) is responsible for exchanging data with EKS Data Service PC and providing the data in the PLC.

### 6.2.4. DB\_EKSData1 block

The (DB\_EKSData1) data block is an example of the structure for the data required for FB\_EKSDataService.

### 6.2.5. Data types type\_EKSxxxx

The data types are required for the declaration of block parameters and data types in the DB.

- type\_EKSDatabase
- type\_EKSDataID
- type\_EKSDescription
- type\_EKSKeyID

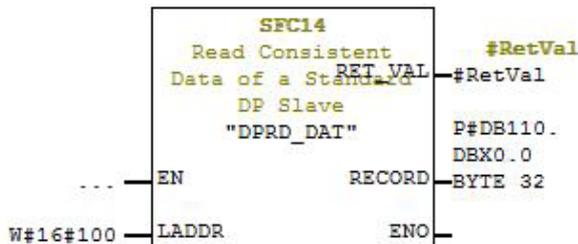
## 6.2.6. FC\_EKS\_Euchner block

The block is called in the Main (OB1) program, and it administers all Euchner blocks in connection with EKS Data Service. The example uses hardware addresses that are created in the device configuration with the declaration of the readers. The correct addresses can be obtained after hardware declaration in the configuration section.

### Address reference EKS\_Euchner

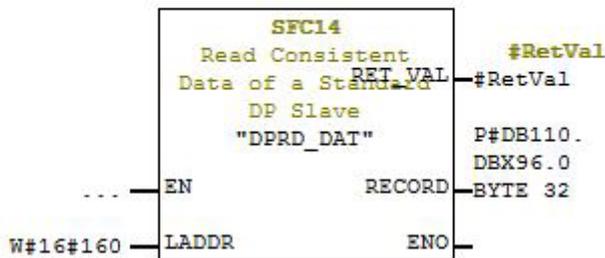
**Network 1:** Read EKS 1 (Status Information)

Comment:



**Network 2:** Read EKS 1 (Key ID)

Comment:



Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	192	Read/Write: 128/120 Byte I/O	256...287	256...287	
2	192	--> Read/Write: 128/120 Byte I/O	288...319	288...319	
3	192	--> Read/Write: 128/120 Byte I/O	320...351	320...351	
4	192	--> Read/Write: 128/120 Byte I/O	352...383	352...375	



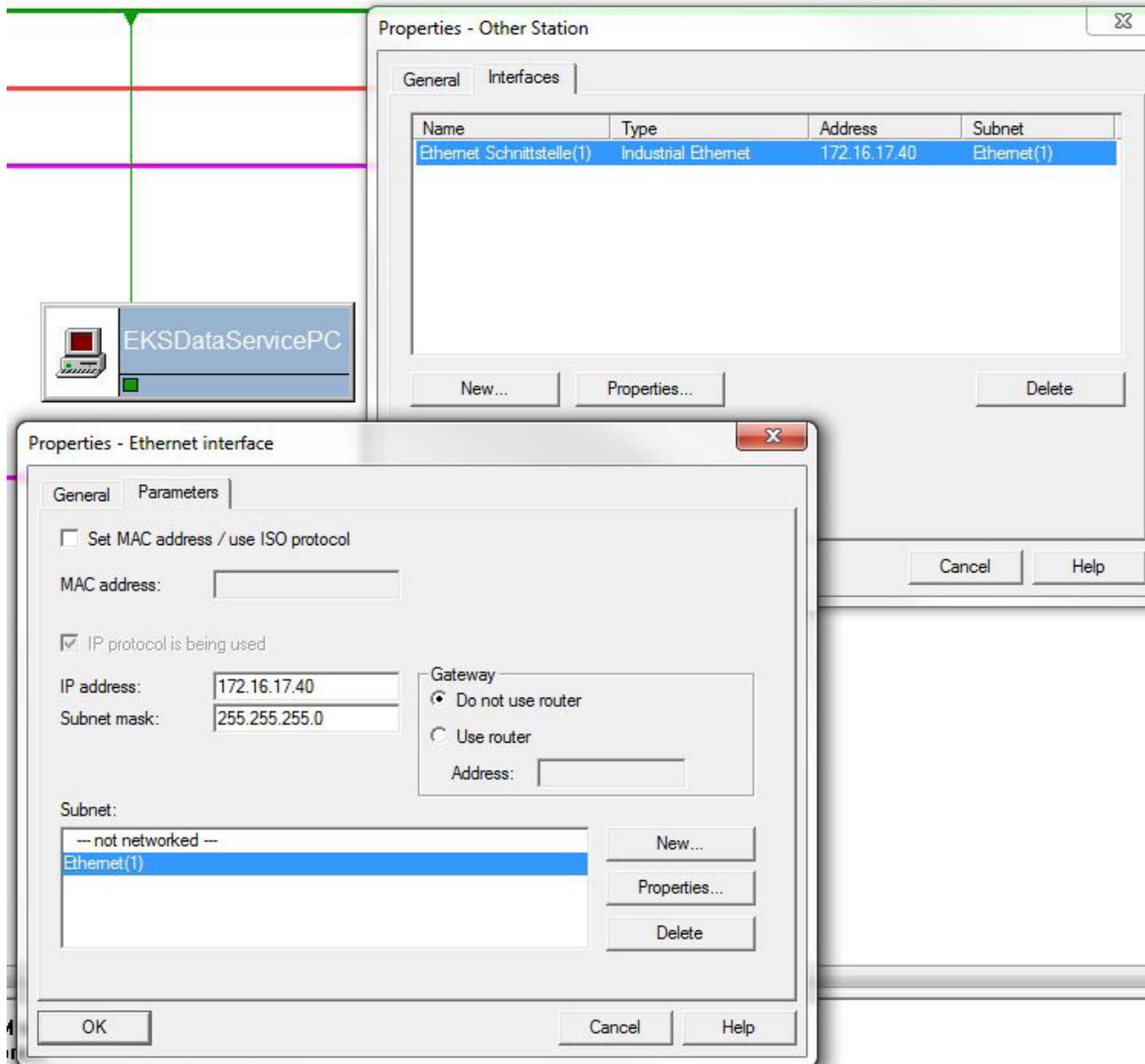
### Notice

The E-address must be specified to LADDR as a HEX value

### 6.3. NetPro settings (TCP/IP configuration)

#### 6.3.1. Setting up EKS Data Service PC

EKS Data Service PC can be set up as *Other Station* in NetPro. If this is not done, the connection must be entered unspecifically. The *Other Station* requires an IP address and must be assigned to an Ethernet subnet.



## Setting up connection in the CPU

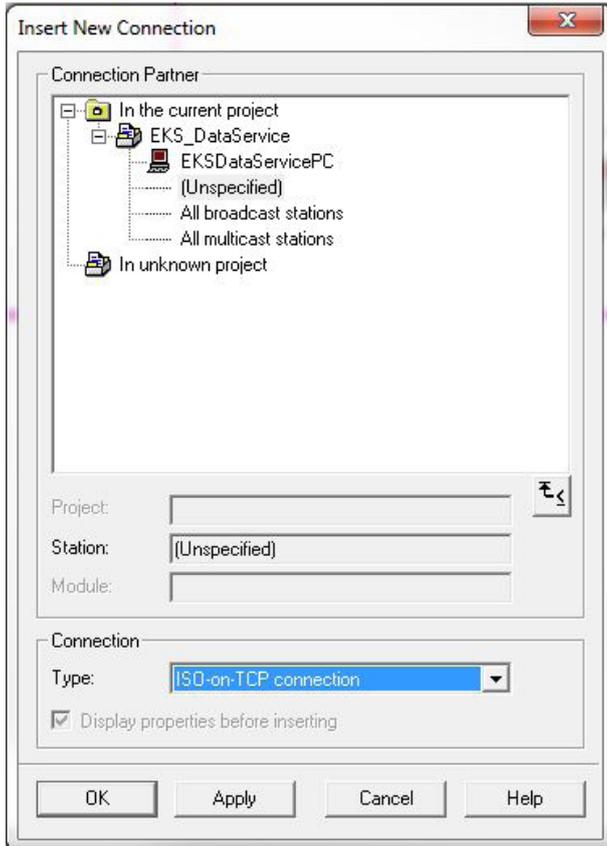
A connection must be set up in the CPU for each instance of the FB\_EKSDataService block. To do this, right-click the required CPU and add a connection in NetPro.

Example – CPU 300:

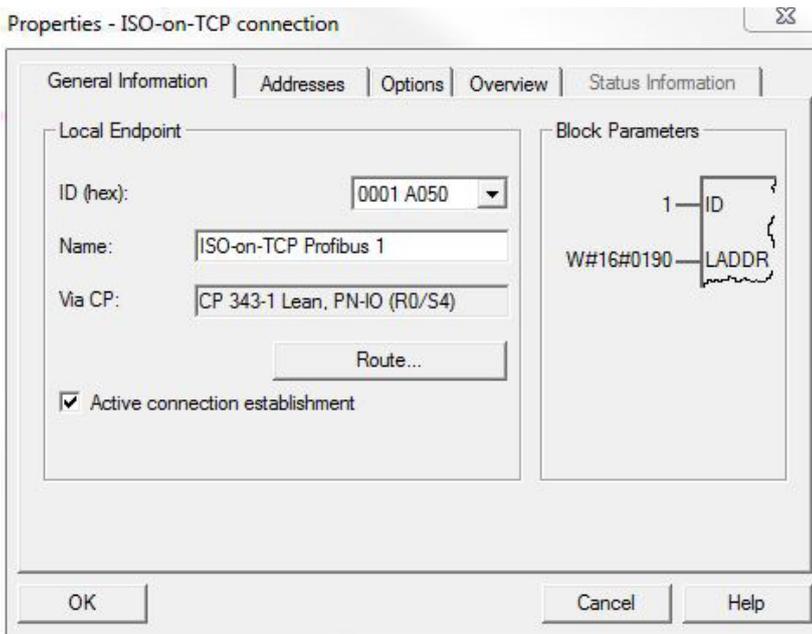
Local ID	Pa	Partner	Type	Active con	Send	Subnet	Local interface	Partner interface	Local address	Partner address
0001 A050		EKSDataServicePC	ISO-on-TCP connection	Yes	-	Ethernet(1) [IE]	PN-IO-2	Ethernet Schnittstelle(1)	172.16.17.64	172.16.17.40
0002 A050		EKSDataServicePC	ISO-on-TCP connection	Yes	-	Ethernet(1) [IE]	PN-IO-2	Ethernet Schnittstelle(1)	172.16.17.64	172.16.17.40

## Adding EKS DataServicePC as an unspecified connection

Select unspecified connection and use ISO-on-TCP connection as the type.

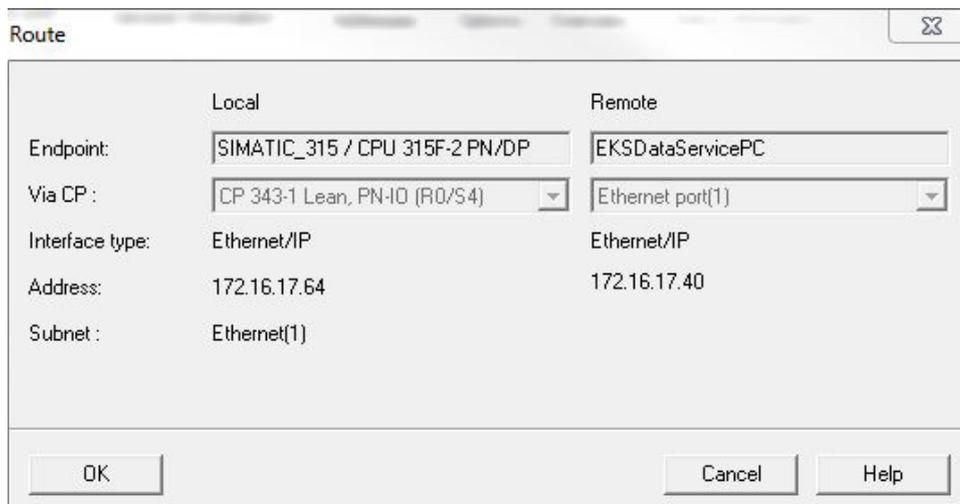


### CPU 300 connection and block parameters



- › The name of the connection can be changed.
- › The active connection setup must be marked.
- › Block parameters are important for programming.

### CPU 300 route selection...



## Addresses

The TSAP must be set here. Any TSAP can be used for the partner *EKSSERV*, Local (entered in log entry in EKS Data Service PC).

Properties - ISO-on-TCP connection ✕

	Local	Remote
IP (dec):	172.16.17.64	172.16.17.40
TSAP (ASC):	S7300_Profibus1	EKSSERV
TSAP (hex):	53.37.33.30.30.5F.50.72.1	45.4B.53.53.45.52.56
TSAP length:	15	7

OK Cancel Help

## 6.4. EKS Data Service block description

The control system requires two blocks to work with EKS Data Service PC.

The first block is an FB (FB\_EKSDataService) in which all functions are programmed. The second block is a DB (DB\_EKSData1) containing the data for the emergency level, the current data for the placed Electronic-Key and the data declaration.

### 6.4.1. DB\_EKSData1

All data stored here can also be distributed to other data blocks. All required data are combined in one DB in the template.

Data of placed Electronic-Keys *type\_EKSDatabase*

Data type *type\_EKSDatabase* describes the structure of the data requested from the database.

All data required in the PLC by the database can be defined in a structure here. The structure must match the description *type\_EKSDescription* (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	KeyID	STRING[16]	''
+18.0	LOCKED	BOOL	FALSE
+18.1	Key_Bit	BOOL	FALSE
+19.0	Key_ShortInt	BYTE	B#16#0
+20.0	Key_Byte	BYTE	B#16#0
+22.0	Key_SmalInt	INT	0
+24.0	Key_Word	WORD	W#16#0
+26.0	Key_Integer	DINT	L#0
+30.0	Key_Float	REAL	0.000000e+00
+34.0	Key_String	STRING[12]	''
+48.0	Key_StringBlankFilled	STRING[10]	''
+60.0	Key_Time	TIME	T#0MS
+64.0	Key_TimeAscii	STRING[8]	''
+74.0	Key_Date	DATE	D#1990-1-1
+76.0	Key_DateAscii	STRING[8]	''
+86.0	Key_BitString	WORD	W#16#0
+88.0	KEYCRC	WORD	W#16#0
+90.0	DB_Bit	BOOL	FALSE
+91.0	DB_ShortInt	BYTE	B#16#0
+92.0	DB_Byte	BYTE	B#16#0
+94.0	DB_SmalInt	INT	0
+96.0	DB_Word	WORD	W#16#0
+98.0	DB_Integert	DINT	L#0
+102.0	DB_Float	REAL	0.000000e+00
+106.0	DB_String	STRING[14]	''
+122.0	DB_StringBlankFilled	STRING[12]	''
+136.0	DB_Time	TIME	T#0MS
+140.0	DB_Date	DATE	D#1990-1-1
+142.0	DB_BitString	BYTE	B#16#0
=144.0		END_STRUCT	

A separate database for saving the data of the currently placed Electronic-Key can be created in the DB for each EKS.

DB100 -- "DB\_EKSData1" -- EKSDataService\S7 300\...\DB100

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"

### Description of the requested data (type\_EKSDescription)

The data are requested from EKS Data Service PC. This request is defined via a structure corresponding to the structure of type `type_EKSDatabase` (see chapter 7.2.2 in the EKS Data Service PC software manual).

Example from the library:

UDT101 -- "type\_EKSDescription" -- EKSDataService\S7 300\...\UDT101

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	KeyID	INT	1
+2.0	LOCKED	INT	2
+4.0	Key_Bit	INT	3
+6.0	Key_ShortInt	INT	4
+8.0	Key_Byte	INT	5
+10.0	Key_SmalInt	INT	6
+12.0	Key_Word	INT	7
+14.0	Key_Integert	INT	8
+16.0	Key_Float	INT	9
+18.0	Key_String	INT	10
+20.0	Key_StringBlankFilled	INT	11
+22.0	Key_Time	INT	12
+24.0	Key_TimeAscii	INT	13
+26.0	Key_Date	INT	14
+28.0	Key_DateAscii	INT	15
+30.0	Key_BitString	INT	16
+32.0	KEYCRC	INT	17
+34.0	DB_Bit	INT	18
+36.0	DB_ShortInt	INT	19
+38.0	DB_Byte	INT	20
+40.0	DB_SmalInt	INT	21
+42.0	DB_Word	INT	22
+44.0	DB_Integert	INT	23
+46.0	DB_Float	INT	24
+48.0	DB_String	INT	25
+50.0	DB_StringBlankFilled	INT	26
+52.0	DB_Time	INT	27
+54.0	DB_Date	INT	28
+56.0	DB_BitString	INT	29
=58.0		END_STRUCT	

The default values are important and can be set in EKS Data Service PC.

The *Description* is entered once in the DB:

DB100 -- "DB\_EKSData1" -- EKSDataService\S7 300\...\DB100

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"
+144.0	Description	"type_EKSDescription"

### ID array for administering the data (type\_EKSDataID)

The ID is required to save additional information such as the time stamp for Electronic-Key placement or the time stamp of the last refresh operation, the KeyID and the CRC. This additional information is entered and stored in the DB for each Electronic-Key placed.

This data type must not be changed and is saved as an array in the DB.

An ID array of the same size is required for each data saving operation (Data Array).

Since the data are required after a power failure/restart as well, they must be marked as remanent.



#### Notice about the checksum

EKS Data Service uses the requested data to calculate a checksum (also called "CRC" for short below) to ensure the integrity of the requested data. These data are not processed by user. This checksum must not be confused with the checksum calculated by the Electronic Key-Manager EKM software. EKM's checksum (also called "Key CRC" for short below) always refers to a certain data range that can be configured in EKM. EKM's CRC is additionally stored on the Electronic-Key and can be used exclusively on "On-Key" fields. By contrast, EKS Data Service calculates the CRC over all requested data (both "On-Key" and EKM database values).

UDT103 -- "type\_EKSDataID" -- EKSDataService\S7 300\...\UDT103

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	timestampPlaced	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+8.0	timestampRefresh	DATE_AND_TIME	DT#90-1-1-0:0:0.000
+16.0	KeyID	"type_EKSKeyID"	
+24.0	CRC	WORD	W#16#0
=26.0		END_STRUCT	

Example from the library:

DB100 -- "DB\_EKSData1" -- EKS\_DataService\SIMATIC 315\CPU 315F-2 PN/DP\...\DB100

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Data_Key_ReaderProfibus	"type_EKSDatabase"	
+70.0	Data_Key_ReaderProfinet	"type_EKSDatabase"	
+140.0	Description	"type_EKSDescription"	
+152.0	ID	ARRAY[0..5]	
+26.0		"type_EKSDataID"	
+308.0	Data	ARRAY[0..5]	
+70.0		"type_EKSDatabase"	
=728.0		END_STRUCT	

## Data array for saving the emergency mode data (type\_EKSDatabase)

The same data type for data of the placed Electronic-Keys is also used for saving the data for the emergency level. All data from the placed Electronic-Keys are entered into the data array. The data will not be overwritten until the number of Electronic-Keys placed exceeds the number of array entries. If data are overwritten, the oldest Electronic-Key entry is always overwritten first. If an Electronic-Key is not listed in the EKS Data Service PC request, the Electronic-Key will be deleted in the data array as well. All instances of FB\_EKSDataService can access the same data (Description, ID, Data) in DB\_EKSData1. If the data are to be available after a power failure as well, they must be marked as remanent.

Example from the library:

DB100 -- "DB\_EKSData1" -- EKS\_DataService\SIMATIC 315\CPU 315F-2 PN/DP\...\DB100

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	Data_Key_ReaderProfibus	"type_EKSDatabase"	
+70.0	Data_Key_ReaderProfinet	"type_EKSDatabase"	
+140.0	Description	"type_EKSDescription"	
+152.0	ID	ARRAY[0..5]	
+26.0		"type_EKSDataID"	
+308.0	Data	ARRAY[0..5]	
+70.0		"type_EKSDatabase"	
=728.0		END_STRUCT	

### 6.4.2. FB\_EKSDataService

FB\_EKSDataService for the S7-300 / S7-400 possesses the following interface description.

INPUT		
Name	Data type	Description
Key_Present	Bool	Electronic-Key is placed
Key_Data_updated	Bool	Electronic-Key data are updated on placement
Interface_ID	Word	Profinet interface hardware ID
Con_ID	Int	Unique connection ID
Timeout_Time	Time	Communication timeout
Refresh_Time	Time	Refresh time for Electronic-Key data

OUTPUT		
Name	Data type	Description
Con_Parametrization_Err	Word	Communication parametrization error
Con_Communication_Error	Word	Server service communication error
DB_Error	Bool	Database length wrong
Data_Error	Bool	Data error from server service
Emergency_Level_1	Bool	Emergency level 1 active
Emergency_Level_2	Bool	Emergency level 2 active
KEY_Not_Found	Bool	Electronic-Key ID not found
KEY_Checksum_Error	Bool	CRC error
KEY_Valid	Bool	Electronic-Key placed and data available
Error	Bool	Block error
Status	Word	Block status
serviceStatus	Word	Status of server service

INOUT		
Name	Data type	Description
Key_ID	type_EKSKeyID	Serial number of placed Electronic-Key
EKS_Data_Key_Reader	Any	Data storage location for the placed Electronic-Key
EKS_Data_Description	Any	Data of the description
EKS_Data	Any	Data array of PLC data
EKS_KEY_ID	Any	ID data for the data structure



**Notice**

FB\_EKSDataService is protected by a password to prevent tampering. In some cases, FB\_EKSDataService must be recompiled if you have incorporated it into your project. You require a password to compile the block. Please do not modify the implementation of FB\_EKSDataService.

Password: *EKS1234*

### Key\_Present

When the EKS reader reads the placed Electronic-Key, this produces a signal confirming the read Electronic-Key serial number. This signal is required for requesting the Electronic-Key data. A data request is sent to EKS Data Service PC when Key\_Present changes from FALSE to TRUE.

### Key\_Data\_updated

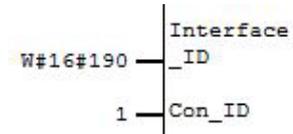
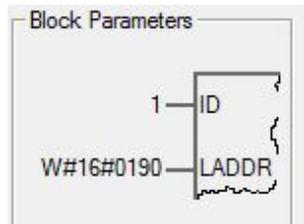
If TRUE is assigned to the parameter, the Electronic-Key data of the placed Electronic-Key are always updated with the database. This means that the data can change during placement.

Example:

An Electronic-Key is placed in EKS, the requested data are already saved in the PLC (DB\_EKSDData1.Data\_Key\_Reader\_Profinet) and data refresh (Refresh\_Time) is activated. In other words, the data of the emergency level are cyclically updated in the background. If the updated value of the placed Electronic-Key differs from the stored value, the value of the currently placed Electronic-Key (DB\_EKSDData1.Data\_Key\_Reader\_Profinet) will be overwritten under Key\_Data\_updated = TRUE. Otherwise, the updated value will be loaded only when the Electronic-Key is placed again.

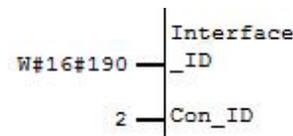
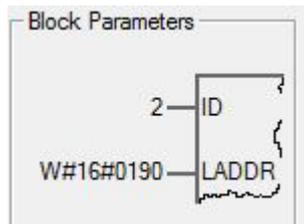
### Interface\_ID

Hardware ID of the local interface via which communication with EKS Data Service PC is to take place.



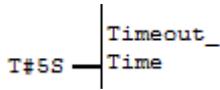
### Con\_ID

The connection ID is taken from the property window and must be output on the block.



### Timeout\_Time

If a timeout occurs during communication with EKS Data Service PC, an error is generated that cancels communication and issues an error with status. In the event of a fault, emergency level 2 is triggered and the requested data are loaded from the emergency-level memory (DB\_EKSData1.Data).



### Refresh\_Time

In this time cycle, EKS Data Service PC retrieves the data of the listed Electronic-Keys in DB\_EKSData1.Data and checks whether they are up to date. Otherwise, the values in DB\_EKSData1.Data will be refreshed. In other words, already requested data will be updated in the background without Electronic-Key placement. The block moves on to the next entry after every cycle and requests the data from EKS Data Service PC. With a setting of 10 s, the service requires 100 s to request 10 stored items of Electronic-Key data.

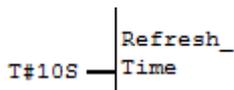
Data refresh can be switched off using the value 0 ms. If several FB\_EKSDataService instances are required and if all instances access the same data, only one instance must perform the refresh operation.

Example:

Instance A: Refresh\_Time = 10 s (responsible for refresh)

Instance B: Refresh\_Time = 0 ms (refresh switched off)

Instance C: Refresh\_Time = 0 ms (refresh switched off)



### Con\_Parametrization\_Errorcode

If an error is generated during connection setup, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.1. *Connection setup (Con\_Parametrization\_Errorcode)*.

### Con\_Communication\_Errorcode

If an error is generated during data reception, the corresponding error code is entered here. A list of error codes can be found in the chapter 7.2. *Data reception (Con\_Communication\_Errorcode)*.

### DB\_Error

The error bit becomes TRUE if the length of structure *type\_EKSDatabase* does not match the sent data size from EKS Data Service PC.

### Data\_Error

EKS Data Service PC outputs the Data\_Error error if the specified data structure does not match the structure of EKS Data Service PC. If EKS Data Service PC reports this error, Data\_Error will be set to TRUE.

### Emergency\_Level\_1

Emergency level 1: EKS Data Service PC cannot access the original EKM CSV export file and instead uses the local backup file of the EKM CSV export file.

### Emergency\_Level\_2

Emergency level 2: EKS Data Service PC cannot provide any data because communication is disrupted. The data of the internal data block (DB\_EKSData1.Daten) are used instead.

### KEY\_Not\_Found

The placed Electronic-Key was not found. This information can originate from EKS Data Service PC, or from the PLC in case of emergency level 2 if the requested Electronic-Key is not stored in the memory (DB\_EKSData1.Data).

## KEY\_Checksum\_Error

The CRC calculated via the requested data does not match. This error is output if the CRC of the sent data does not match, or if the CRC of the saved data is incorrect in case of emergency level 2.

## KEY\_Valid

The data of the placed Electronic-Key are available. This also applies to the emergency levels.

## Error

An error occurred during block processing. The error will be deleted during the next processing operation and then possibly output again.

## Status

Status messages that are generated as part of processing in FB\_EKSDataService. A list of status messages can be found in the chapter 7.3. *Status of FB\_EKSDataService (Status)*.

## serviceStatus

Status codes that are transmitted from EKS Data Service PC to FB\_EKSDataService. A list of status messages can be found in the chapter 7.4. *Data request status (serviceStatus)*.

## Key\_ID

The serial number of the placed Electronic-Key must be specified here to request the data from EKS Data Service PC. Data type `type_EKSKeyID` must be used for this purpose.

```

P#DB110.DB
  X120.0
  "DB_EKS_
  KeyData".
  EKSKeyID — Key_ID
    
```

## EKS\_Data\_Key\_Reader

The data of the placed Electronic-Key are copied to this address. Data type `type_EKSDatabase` must be used.

```

P#DB100.DB
  X0.0
  "DB_
  EKSDat1".
  Data_Key_
  Reader_
  Profibus — EKS_Data_
  Key_
  Reader
    
```

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"
+748.0	Data	ARRAY[0..20]
*144.0		"type_EKSDatabase"

### EKS\_Data\_Description

The data description is transmitted as the data type variant. This data structure is used to request the required data from EKS Data Service PC. The data structure of the data type *type\_EKSDescription* must be used.

```

P#DB100.DB
  X70.0
  "DB_
EKSData1". EKS_Data_
  Descriptio Descripti
n          on
  
```

DB100 -- "DB\_EKSData1" -- EKSDatService\S7 300\...\DB100

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"
+144.0	Description	"type_EKSDescription"

### EKS\_Data

The data array for data backup on the PLC is created as a variant (see DB\_EKSData1).

```

P#DB100.DB
  X238.0
  "DB_
EKSData1".
  Data      EKS_Data
  
```

DB100 -- "DB\_EKSData1" -- EKSDatService\S7 300\...\DB100

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"
+144.0	Description	"type_EKSDescription"
+202.0	ID	ARRAY[0..20]
+26.0		"type_EKSDataID"
+748.0	Data	ARRAY[0..20]
+144.0		"type_EKSDatabase"

The data array must be the same size as EKS\_KEY\_ID Array.

### EKS\_KEY\_ID

The ID range is available for administering data backup (see ID DB\_EKSData1).

```

P#DB100.DB
  X82.0
  "DB_
EKSData1". EKS_KEY_
  ID       ID
  
```

DB100 -- "DB\_EKSData1" -- EKSDatService\S7 300\...\DB100

Address	Name	Type
0.0		STRUCT
+0.0	Data_Key_Reader_Profibus	"type_EKSDatabase"
+144.0	Description	"type_EKSDescription"
+202.0	ID	ARRAY[0..20]
+26.0		"type_EKSDataID"

The data array must be the same size as EKS\_Data Array.

## 7. Status messages

### 7.1. Connection setup (Con\_Parametrization\_Errorcode)

Excerpt from the Siemens description:

STATUS* (W#16#...)	Explanation
0	Connection was set up successfully.
7000	Task processing not active.
7001	Start task processing, set up connection.
7002	Setting up connection (REQ irrelevant).
8085	The connection ID (ID parameter) is already being used by a configured connection.
8086	ID parameter is outside the permissible range.
8087	Maximum number of connections reached; no further connection possible.
8089	The CONNECT parameter does not point to a connection description, or the connection description was created manually.
809A	The structure on the CONNECT parameter is not supported on an integrated interface, or the length is invalid.
809B	The Interfaceld element within the TCON_XXX structure does not reference any hardware ID of a CPU or CM/CP interface or has the value "0."
80A1	The specified connection or the port is already assigned.
80A2	System uses local or remote port. The following local ports are reserved: 20, 21, 80, 102, 135, 161, 162, 443, 34962, 34963, 34964, as well as the range 49152 to 65535.
80A3	ID is being used by a connection created by the user program that uses the same connection description on the CONNECT parameter.
80A4	IP address of the remote end point is invalid or corresponds to the IP address of the local partner.
80A7	Communication error: You executed "TDISCON" before "TCON" was ended.
80B4	Only for TCON_IP_RFC: The local T-selector was not specified, or the first byte does not contain the value 0x0E (only for a T-selector length = 2), or the local T-selector begins with "SIMATIC-."
80B5	Only passive connection setup is permissible for connection type 13 = UDP (ActiveEstablished parameter of the TCON_IP_v4 / TCON_PARAM structure has the value TRUE).
80B6	Parametrization error in the ConnectionType parameter of the data block for the connection description.
	Valid only for TCON_IP_v4: 0x11, 0x0B and 0x13.
	Valid only for TCON_IP_RFC: 0x0C and 0x12
80B7	For TCON_IP_v4:
	TCP (active connection setup): remote port is "0."
	TCP (passive connection setup): local port is "0."
	UDP: local port is "0."
	IP address of the partner end point was set to 0.0.0.0.
	For TCON_IP_RFC:
	Local (LocalTSelector) or remote (RemoteTSelector) T-selector was specified with a length of more than 32 bytes.
	A length of more than 32 bytes was entered for TSelLength of the T-selector (local or remote).
	IP address length error of the respective connection partner.
	IP address of the partner end point was set to 0.0.0.0.

80B8	ID parameter in the local connection description (structure on CONNECT parameter) and ID parameter of the instruction differ.
80C3	All connection resources are assigned, or other applications or connections might be using ports dynamically.
80C4	Temporary communication error:
	The connection cannot be set up at present.
	The connection cannot be set up because firewalls along the connection path are not enabled for the required ports.
	The interface is currently receiving new parameters.
	The configured connection is currently being removed by a "TDISCON" instruction.
80C5	The connection partner refuses connection setup or has cleared down or actively ended the connection.
80C6	The connection partner is unreachable (network error).
80C7	Execution timeout.
80C8	Value on the ID parameter is already being used by a connection created via the user program. The connection uses the same ID but different connection settings on the CONNECT parameter.
80C9	Validation of the connection partner failed. The connection partner that is attempting to set up a connection does not correspond to the defined partner of the structure on the CONNECT parameter.
80CE	The IP address of the local interface is 0.0.0.0.
80D0	In connection with TCP and the active connection end point: the remote_qdn parameter is an empty character string. No connection can be set up in this case.

## 7.2. Data reception (Con\_Communication\_Errorcode)

Excerpt from the Siemens description:

STATUS* (W#16#...)	Explanation
0	Transmission task ended without errors.
7000	Task processing not active.
7001	Task processing begins; data are transmitted.
	The operating system accesses the data in the DATA send area during this processing operation.
7002	Task is being processed (REQ irrelevant).
	The operating system accesses the data in the DATA send area during this processing operation.
8085	LEN parameter exceeds the largest permissible value (65536).
	The DATA and LEN parameters both have the value "0."
8086	The ID parameter is outside the permissible address range (1..0xFFF).
8088	LEN parameter is larger than the range specified in DATA.
80A1	Communication error:
	The specified connection has not been set up yet.
	The specified connection is being terminated. Transmission via this connection is not possible.
	The interface is being reinitialized.
80B3	The configured protocol variant (ConnectionType parameter in the connection description) is UDP. Please use the "TUSEND" instruction for a UDP connection.
80C3	A block with this ID is already being processed in another priority group.
	Internal lack of resources.
80C4	Temporary communication error:
	The connection to the partner cannot be set up at present.
	The interface is currently receiving new parameter settings, or the connection is being set up.
80C5	Communication partner closed the connection.
80C6	Network error. Communication partner unreachable.
80C7	Timeout during execution.

### 7.3. Status of FB\_EKSDataService (Status)

STATUS* (W#16#...)	Explanation
5001	Transmission error
5002	Reception error
5003	Reception timeout
5004	Length error of received data
5005	DATA_ERROR telegram
5006	CRC error during data reception
5007	Transmission timeout
5008	KeyID of received data differs from KeyID of request
6001	Array length for ID and data not identical
6002	Emergency Level 2 is active, and there are no Key data
6003	CRC error during data creation via DB Data

### 7.4. Data request status (serviceStatus)

STATUS* (W#16#...)	Explanation
E001	Electronic-Key unknown
E002	Field unknown in the description
E003	Telegram size exceeded
E004	Format error
E005	EKM CSV file error
F001	Emergency level 1 EKM CSV file unusable



Euchner GmbH + Co. KG  
Kohlhammerstraße 16  
D-70771 Leinfelden-Echterdingen  
info@euchner.de  
www.euchner.com

Edition:  
2528410-02-08/23  
Title:  
Software Manual  
EKS Data Service PLC  
(translation of the original operating instructions)  
Copyright:  
© EUCHNER GmbH + Co. KG, 08/2023

Subject to technical modifications; no responsibility  
is accepted for the accuracy of this information.