

A man in a light blue shirt is shown from the side, holding a tablet computer. He is looking at the screen, which displays a software interface with various charts and data. The background is a blurred industrial factory setting with machinery and equipment.

SIEMENS

Application description • 10/2015

Access XML files from S7-1500 Software Controller

SIMATIC ODK 1500S XML Data Access V1.0

<https://support.industry.siemens.com/cs/ww/en/view/109479496>

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.

If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit <http://www.siemens.com/industrialsecurity>.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit <http://support.industry.siemens.com>.

Table of contents

	Warranty and liability.....	2
1	Task.....	4
	1.1 Overview.....	4
	1.2 Needed knowledge.....	4
	1.3 Required Hardware and Software Components	5
	1.3.1 Validity.....	5
	1.3.2 Used Components.....	5
2	Installation	7
	2.1 Quick-start	7
	2.2 Installation on runtime system.....	8
	2.3 Installation on engineering system STEP 7 V13 SP1	9
3	Driver supported functionality	10
	3.1 General overview	10
	3.2 Functional range.....	11
4	Function blocks (FBs).....	12
	4.1 Overview.....	12
	4.2 ODK1500S_XML_Load.....	13
	4.3 ODK1500S_XML_Unload	14
	4.4 ODK1500S_XML_DocLoadFile	15
	4.5 ODK1500S_XML_DocSaveFile	16
	4.6 ODK1500S_XML_AddItems	17
	4.7 ODK1500S_XML_Get.....	18
	4.8 ODK1500S_XML_Set	19
5	Error codes.....	20
	5.1 Error codes of ODK 1500S.....	20
	5.2 Error codes of the ODK 1500S XML Data Access driver.....	22
6	History.....	23

1 Task

1.1 Overview

Introduction

By using S7-150xS PLC it is not possible to read from or write into XML-files.

The add-on ODK 1500S XML Data Access offers this functionality.

1.2 Needed knowledge

To understand this document the following knowledge is needed:

- SIMATIC S7-150xS PLC
- STEP 7 V13 SP1 (TIA-Portal)

1.3 Required Hardware and Software Components

1.3.1 Validity

This application example is valid for

- STEP 7 V13 SP1 or newer
- S7-1500 Software Controller S7-150xS V1.8

1.3.2 Used Components

The application was generated using the following components:

Hardware components

Table 1-1

Component	No.	Order number	Note
ET 200SP Open Controller CPU 1515SP PC + HMI 128PT 4-GB RAM version with WES7 P (Multitouch) 64Bit and 16-GB CFast card	1	6ES7677-2AA41-0FK0	

Standard software components

Table 1-2

Component	No.	Order number	Note
STEP 7 Prof. V13 SP1	1		
SIMATIC ODK 1500S V1	1	6ES7806-2CD00-0YA0	The ODK is not needed to use this driver application!
...			

Note

The ODK (Open Development Kit) is not needed to use this application!

Sample files and projects

The following list includes all files and projects that are used in this example.

Table 1-3

Component	Note
109479496_ODK1500S-XMLDataAccess_Doc_en.pdf	This documentation
Driver \ ODK1500S_XML_DataAccess.dll	The driver's DLL
Driver \ install.bat	Setup batch for the driver
TIAv13-ODK1500S- XML_DataAccess -Example	STEP 7 example project including all function blocks of the driver

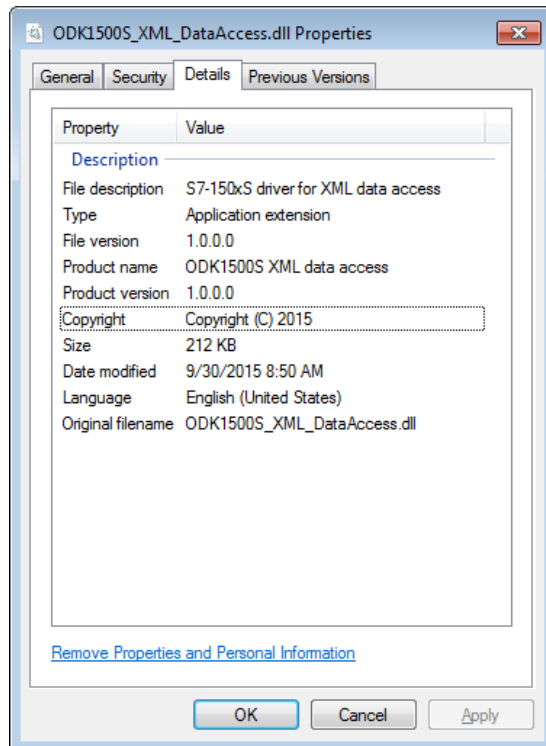
Version of the driver

The driver DLL is located in the “ODK 1500S ODK” directory:

<\$ProgramData>\Siemens\Automation\ODK1500S\ODK1500S_XML_DataAccess.dll

You can identify the version of the driver DLL in the file properties (Windows explorer → right click → properties)

Figure 1-1 Properties of the driver DLL



NOTICE

If the version of the installed DLL does not match the installed STEP 7 function blocks, the following error code may returned:

0x8098 - ODK application could not be executed because the function is not supported.

2 Installation

2.1 Quick-start

Runtime-System

- Install the driver DLL on the run-time system with the **setup.bat**

Engineering system STEP 7 V13 SP1

- De-archive and open the STEP 7-demo-project-file included in delivery
- Copy driver FBs and SFBs into own project
- Integrate function Block ODK1500S_XML_Load
- Call function block ODK1500S_XML_DocLoadFile with fitting file-name(positive edge opens file)
- Call function blocks ODK1500S_XML_Get / ODK1500S_XML_Put for reading / writing (positive edge reads / writes file)

2.2 Installation on runtime system

To install the ODK 1500S XML Data Access driver on the runtime system, run the `setup.bat` batch file as administrator. This will copy the driver DLL to the right folder on the S7-1500 Software Controller and register the DLL in the registry. This installation works from a USB stick, too.

Installation under Windows 7

For copying a file to system32 Administrator privileges are needed. Thus the **setup.bat** has to be started as Administrator (right click – Run as Administrator). The Windows 7 UAC has to be confirmed with “Yes”.

This works also from an USB stick.

2.3 Installation on engineering system STEP 7 V13 SP1

Platform requirements:

- This documentation
- STEP 7 V13 SP1 (TIA-Portal) demo application

The S7 demo application contains the program blocks. You can copy the driver function blocks to your application.

An installation of the driver (**setup.bat**) on the development-system is **not** necessary.

S7-150xS - CPU in the project

If the driver function blocks are copied to a project without a S7-150xS CPU the FBs are marked with an error, because the used ODK SFBs are not supported.

Copy the elements (blocks, data types)

To use the ODK 1500S XML Data Access in a TIA Portal project the following components are needed:

- The driver blocks (folder "Program blocks")
- PLC data types (folder "PLC data types" → XMLItem),

A second instance of the TIA Portal is needed for copying. Open the target project in the second instance and copy the function blocks (e.g. "*drag and drop*") from the demo project (TIAv13-ODK1500S- XML_DataAccess -Example).

Compiling blocks

In some cases the TIA portal may mark the driver function blocks red, because some functions are not recognized right. In this case you have to compile the function blocks again in order to update the function block's interfaces.

3 Driver supported functionality

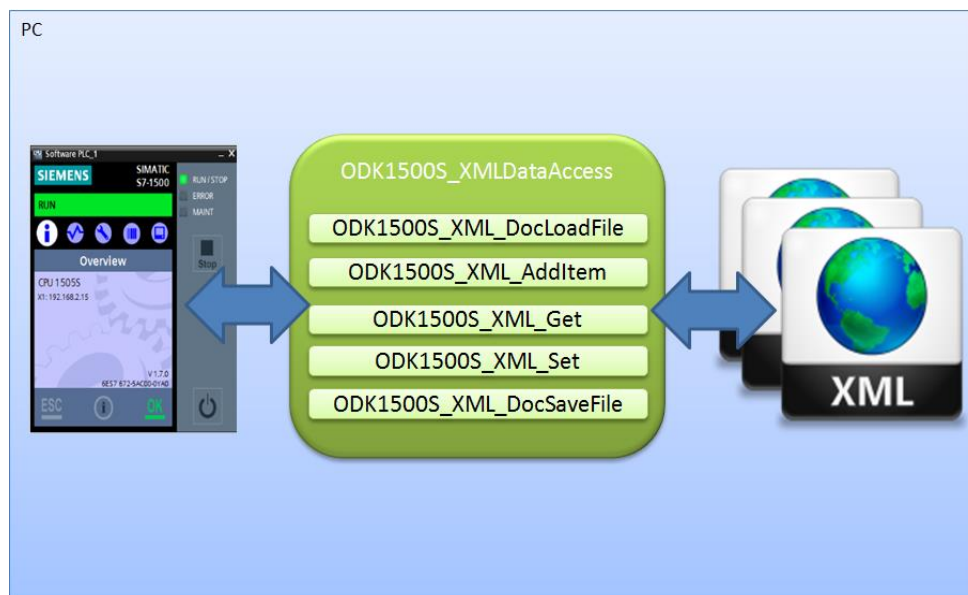
3.1 General overview

The solution consists of two parts:

- STEP 7-function blocks for reading and writing XML-tags
- Driver-DLL which realizes this function

The user has to install the DLL on his target-system(run-time-system). On the engineering-station (TIA Portal) only the new FBs are needed for the customer-project.

Image 3-1 Structure of solution



3.2 Functional range

This driver has been written for those cases of use, in which any (or just single) data has to be written into or read from XML-files.

It is not about to write the complete XML-file in a data block, but to especially edit specific information from the XML-file(reading / writing).

For example the user gets a really big XML-data from the plant operator with any kind of information. The S7-150xS just needs some specific data.

In the world of XML the XPath Syntax lends itself to access data in an XML-file. (see e.g. http://www.w3schools.com/xpath/xpath_syntax.asp)

The data is being defined by XPath.

For parsing the XML-file PugiXML is being used.

PugiXML is licensed by MIT License:

Copyright (c) 2006-2015 Arseny Kapoulkine

„ Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

4 Function blocks (FBs)

4.1 Overview

Following functions are being realized:

- ODK1500S_XML__Load → driver initialisation
- ODK1500S_XML_DocLoadFile → file opening / parsing
- ODK1500S_XML_AddItems → add new items(tags) for Set/Get
- ODK1500S_XML_Get → value requesting with XPath
- ODK1500S_XML_Set → value writing with XPath
- ODK1500S_XML_DocSaveFile → saving the XML-file
- ODK1500S_XML__Unload → unload driver

4.2 ODK1500S_XML__Load

Before using the ODK 1500S XML Data Access driver, the driver has to be loaded using the following function block.

Figure 4-1 Function block ODK1500S_XML__Load

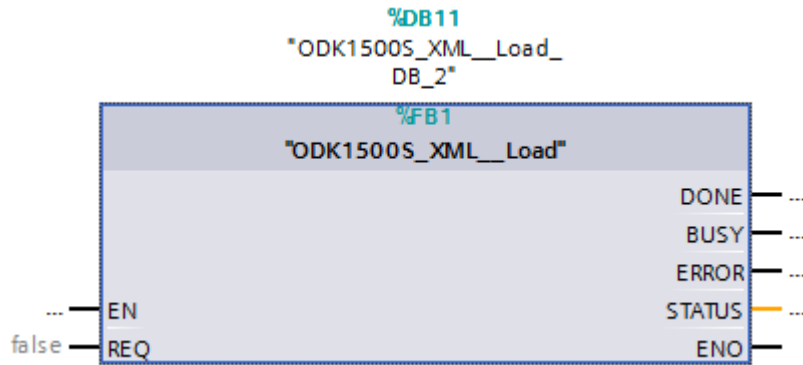


Table 4-1 Parameter of ODK1500S_XML__Load

Parameter	In/Out	Type	Description
REQ	In	Bool	Positive edge triggers the function
DONE	Out	Bool	Function finished
BUSY	Out	Bool	Function still busy
ERROR	Out	Bool	Error occurred during function execution (ODK call)
STATUS	Out	Int	Status of the function execution. (ODK call

PC Start-up

The ODK 1500S XML Data Access is implemented as a Windows DLL.

When the PC is switched on, the S7-1500 Software Controller is ready some time before the Windows OS has booted completely. In this early phase it makes no sense to load a Windows based driver.

Therefore the _Load function must take place after the Windows boot is completed. To achieve this, the Diagnostic Error Interrupt (OB 82) can be utilized.

4.3 ODK1500S_XML__Unload

The explicit unload of the driver is only needed in a few cases, e.g. when updating the driver DLL.

The explicit unload of the driver can be done using the following function:

Figure 4-2 Function block ODK1500S_XML__Unload

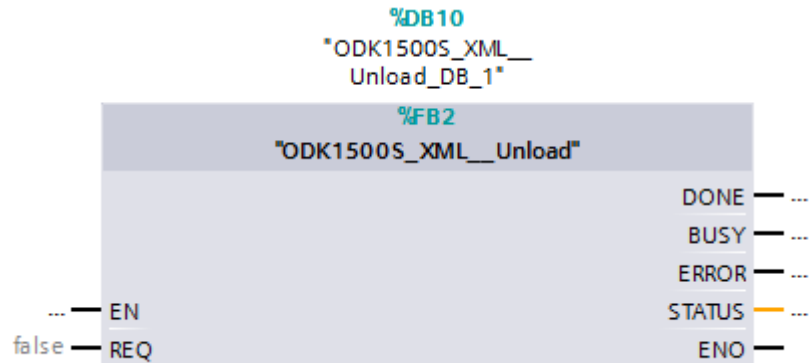


Table 4-2 Parameter of ODK1500S_XML__Unoad

Parameter	In/Out	Type	Description
REQ	In	Bool	Positive edge triggers the function
DONE	Out	Bool	Function finished
BUSY	Out	Bool	Function still busy
ERROR	Out	Bool	Error occurred during function execution (ODK call)
STATUS	Out	Int	Status of the function execution. (ODK call)

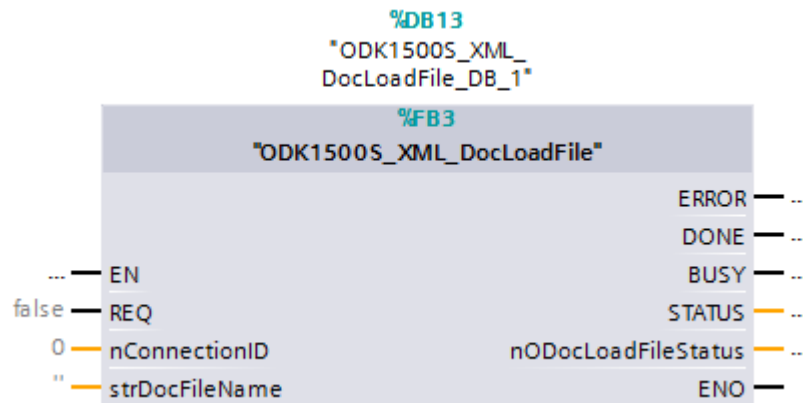
4.4 ODK1500S_XML_DocLoadFile

With this FB an XML-file can be opened.

The call is configured asynchronous, i.e. an edge at REQ signals the job and a TRUE at DONE signals the end of a job.

By the ID the particular file is being responded. Up to 32 files are being supported at the same time.

Figure 4-3 Function block ODK1500S_XML__DocLoadFile



The call is configured asynchronous, i.e. an edge at REQ signals the job and a high-level DONE signal the end of a job.

By the parameter nConnectionID the particular file is being responded. Up to 32 files are being supported at the same time.

Table 4-3 Parameter of the ODK1500S_XML__DocLoadFile

Parameter	In/Out	Type	Description
REQ	In	bool	Positive edge triggers the function
nConnectionID	In	int	Ident for this file 0..31
strDocFileName	In	String	Name of XML document
BUSY	Out	BOOL	Job running
DONE	Out	BOOL	Job finished without error
ERROR	Out	BOOL	Error occurred
nODocLoadFileStatus	Out	DWORD	Error information

4.5 ODK1500S_XML_DocSaveFile

With this FB an opened XML-file is being saved.

The call is configured asynchronous, i.e. an edge at REQ signals the job and a high-level DONE signal the end of a job.

By the ID the particular file is being responded. Up to 32 files are being supported at the same time.

Figure 4-4 Function block ODK1500S_XML_DocLoadFile

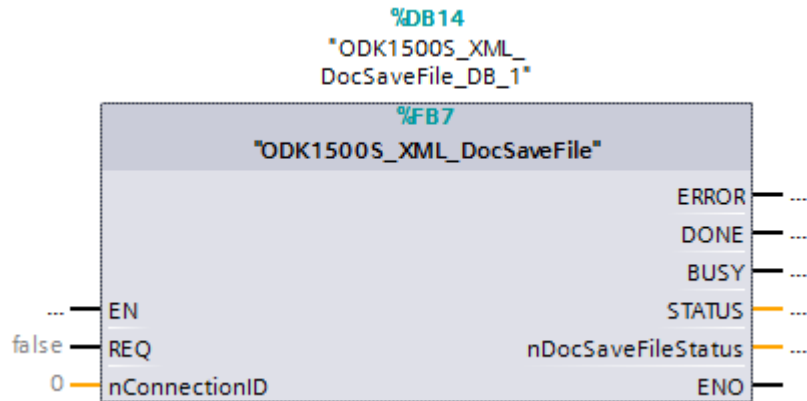


Table 4-4 Parameter of the ODK1500S_XML_DocSaveFile

Parameter	In/Out	Type	Description
REQ	In	BOOL	Positive edge triggers the function
nConnectionID	In	INT	Ident for this file 0..31
strDocFileName	In	String	
BUSY	Out	BOOL	Job running
DONE	Out	BOOL	Job finished without error
ERROR	Out	BOOL	Error occurred
nDocSaveFileStatus	Out	DWORD	Error information

4.6 ODK1500S_XML_AddItems

With this FB a list of XML items-descriptions will be loaded into the driver.

All Items are defined with a XPath term.

These items can be read or written by ODK1500S_XML_Get or ODK1500S_XML_Set.

Figure 4-5 Function block ODK1500S_XML_Get

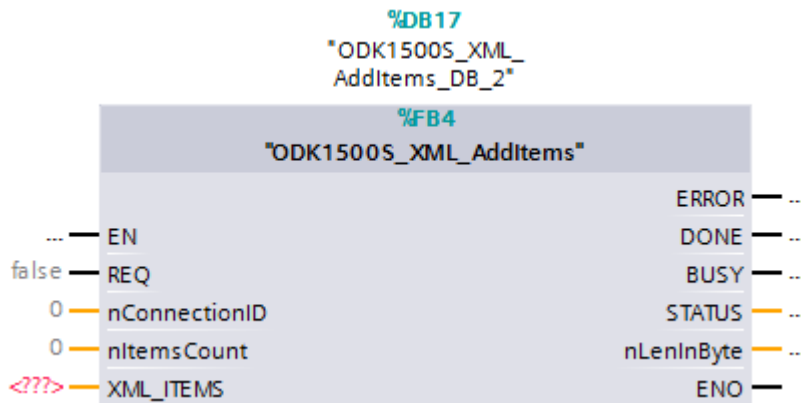


Table 4-5 Parameter of the ODK1500S_XML__Get

Parameter	In/Out	Type	Description
REQ	In	bool	Positive edge opens file
nConnectionID	In	int	Ident for this file 0..31
nItemCount	In	int	Item counts
XML_ITEMS	InOut	array of string	XML item list as Xpath descriptions
BUSY	Out	bool	Job running
DONE	Out	bool	Job finished without error
ERROR	Out	bool	Error occurred
STATUS	Out	word	ODK error status
nLenInByte	Out	int	Length in byte of requested XML Data

4.7 ODK1500S_XML_Get

With this FB the XML-data is being read and saved in the PLC.

The selection, which XML-data has to be read, is being carried out by an XPath-expression. The selection of many homogenous elements is possible. On the output side an array is being returned.

Figure 4-6 Function block ODK1500S_XML_Get

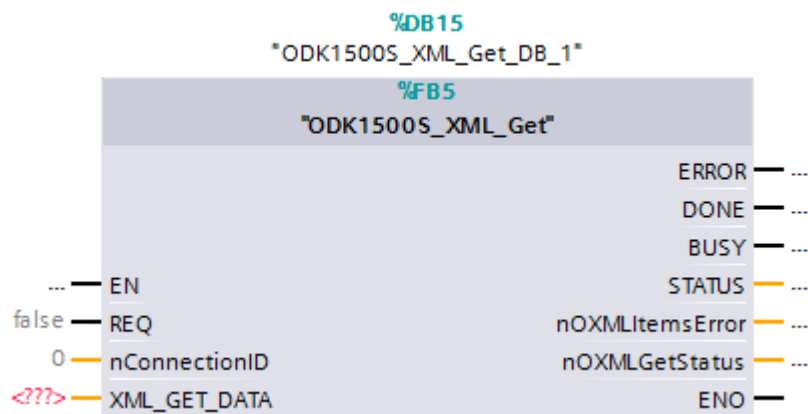


Table 4-6 Parameter of the ODK1500S_XML_Get

Parameter	In/Out	Type	Description
REQ	In	bool	Positive edge opens file
nConnectionID	In	int	Ident for this file 0..31
XML_GET_DATA	InOut	Any	DB into XML data will be written
BUSY	Out	bool	Job running
DONE	Out	bool	Job finished without error
ERROR	Out	bool	Error occurred
STATUS	Out	word	ODK error status
nXMLItemsError	Out	int	First error item
nXMLGetStatus	Out	int	Status of XML Get function

NOTICE XML_GET_DATA should be enough long to receive all read data. Otherwise not all items can be read.

4.8 ODK1500S_XML_Set

With this FB the XML-data with the values are written from the PLC.

The selection, which XML-data has to be read, is being carried out by an XPath-expression. The selection of many homogenous elements is possible. On the output side an array is being returned.

Figure 4-7 Function block ODK1500S_XML_Set

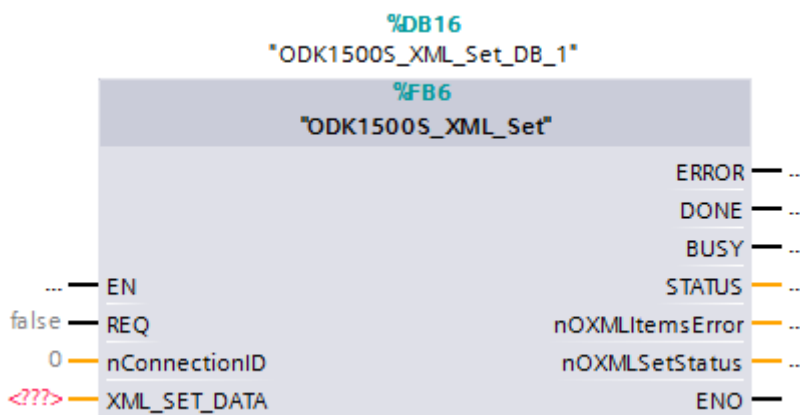


Table 4-7 Parameter of the ODK1500S_XML_Set

Parameter	In/Out	Type	Description
REQ	In	bool	Positive edge opens file
nConnectionID	In	int	Ident for this file 0..31
XML_SET_DATA	InOut	Any	From this DB XML data will be written
BUSY	Out	bool	Job running
DONE	Out	bool	Job finished without error
ERROR	Out	bool	Error occurred
STATUS	Out	word	ODK error status
nXMLItemsError	Out	int	First error item
nXMLSetStatus	Out	int	Status of XML Get function

NOTICE XML_SET_DATA should have all data that will be written in XML file. Otherwise not all items can be written

5 Error codes

The ODK 1500S XML Data Access driver can provide different classes of error messages:

- Code in the FB-output ODK_ERROR_CODE according to ODK 1500S (see chapter 5.1 in this document)
- Special error codes of the ODK 1500S XML Data Access driver (see chapter 5.2 in this document)

5.1 Error codes of ODK 1500S

The driver had been developed with the ODK 1500S (Open Development Kit). The ODK can generate error codes, which are returned in the ODK_ERROR_CODE of the FBs.

Table 5-1 ODK 1500S System-error-codes

ODK Code (HEX)	Description
0	Success
8001	An exception occurred.
8002	Input: the ANY pointer is invalid.
8003	Input: the ANY pointer range is invalid.
8004	Output: the ANY pointer is invalid.
8005	Output: the ANY pointer range is invalid.
8006	More bytes were written into the output buffer by the extension object than were allocated.
8007	ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
8008	The supplied handle value does not correspond to a valid extension object.
8009	More bytes were written into the input buffer by the extension object than were allocated.
807F	An internal error occurred.
80C3	Maximum number (32) of parallel jobs/instances exceeded.
8102	The call to CLSIDFromProgID failed.
8103	The call to CoInitializeEx failed.
8104	The call to CoCreateInstance failed.
8105	The library failed to load.
8106	A Windows response timeout occurred.
8107	Controller is in an invalid state for scheduling an OB.
8108	Schedule information for OB is invalid.
8109	Instance ID for SFB65001 call is invalid.
810A	Controller could not load proxy DLL.
810B	The S7-150xS controller could not create or initialize shared memory area.
810C	Attempt to access unavailable option occurred.
8201	The Execute command index could not be found

5 Error codes

ODK Code (HEX)	Description
8250	No more available positions in the job list
8252	The count is invalid
8253	A data type of an item in the list is invalid
8254	The count specified is invalid
8255	A memory area of an item in the list is invalid
8256	A DB number of an item in the list is invalid
8257	A bit number of an item in the list is invalid
8258	A pBuff of an item in the list is invalid
8259	A data quantity is invalid
825A	The area offset parameter is invalid for this type
825B	The frequency value is invalid
825C	The callback pointer is invalid
825D	The job ID pointer is invalid
825E	The job ID is invalid
825F	Job could not be completed because address is incorrect
8260	Job could not be completed because of protection level
8261	Job could not be completed because of hardware issues
8301	Invalid Thread Execution Priority
8401	Invalid Asynchronous Event
8402	Asynchronous Processor Queue is empty
8403	Asynchronous Processor Queue is full

5.2 Error codes of the ODK 1500S XML Data Access driver

Among the general error bit of the driver FBs there is a special error code in the XML_ERROR_CODE, which describes the reason of the problem.

Table 5-2 Error codes of the ODK 1500S XML Data Access driver

```
//No Error:  
0x80000000 // no error occurred  
  
//XML Additional Errors:  
0x80000002 //Error on save data  
0x80000003 //XPath syntax error  
0x80000004 //No Items in XMLGet function  
0x80000005 //Data counts is wrong.  
0x80000006 //No items found in XMLSet function  
0x80000007 //XML Items count is wrong.  
0x80000040 // XML Doc is not open  
0x80000050 // Error on ODK input buffer.  
0x80000052 // Error on ODK output buffer.
```

6 History

Table 67-1 Version History

Version	Date	Modifications
V1.0.0	10.2015	First version