

WinAC

File-I/O

User documentation

V1.5 • February 2010

Applikationen & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

<http://support.automation.siemens.com/WW/view/en/48172192>

If you have any questions concerning this document please e-mail us to the following address:

online-support.automation@siemens.com

SIEMENS

SIMATIC WinAC File I/O

Basic information and data	1
Quick start	2
Interface	3
Function Mode of AccessDB Driver	4
Project Configuration in WinAC	5
Application in the external user-program	6
Error codes	7
Example of use	8
History	9

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 Siemens Industry Sector.

Table of Contents

Warranty and Liability	4
Foreword.....	6
1 Basic information and data.....	7
1.1 Reference System.....	7
1.2 Installation	7
2 Quick start	8
3 Interface.....	9
4 Function of the WinAC extension File-I/O.....	10
5 Projecting	12
5.1 Initialisation.....	12
5.2 „Normal“ processing.....	12
6 Step7 demo program.....	13
6.1 Overview	13
6.2 Usage of the demo program	13
7 Examples of use	14
8 Error codes.....	15
9 History.....	17

Foreword

Objective of the application

The WinAC extension **File-I/O** allows file operation (read / write / delete) in a PLC program in an easy way. It is possible to write whole data blocks or parts of it in files. Furthermore you can read the files into data blocks or delete files.

The files are written in binary form.

If you want to read a file into a data block you have consider the structure of the data block. The structure of the data block range saving data to file must correspond with the structure of the data block range loading data from file.

1 Basic information and data

1.1 Reference System

The described application in this user documentation is based on the following reference system:

WinAC RTX-F 2009

Simatic Field PG with
Simatic Manager V5.4, SP4

Windows XP /

Windows XPE SP3 with Delivery-Image for SIMATIC IPC427C

1.2 Installation

The installation is carried out by calling the supplied Setup. This installs all necessary system files (drivers and libraries) as well as the Step7 programmes (library and demo – programme).

During the installation the user must specify an index for filing the required files. This index is used – among other things – as an archive for filing a Step7 demo programme (Client_b.zip). In order to use this it has to be retrieved by using the Simatic Manager.

2 Quick start

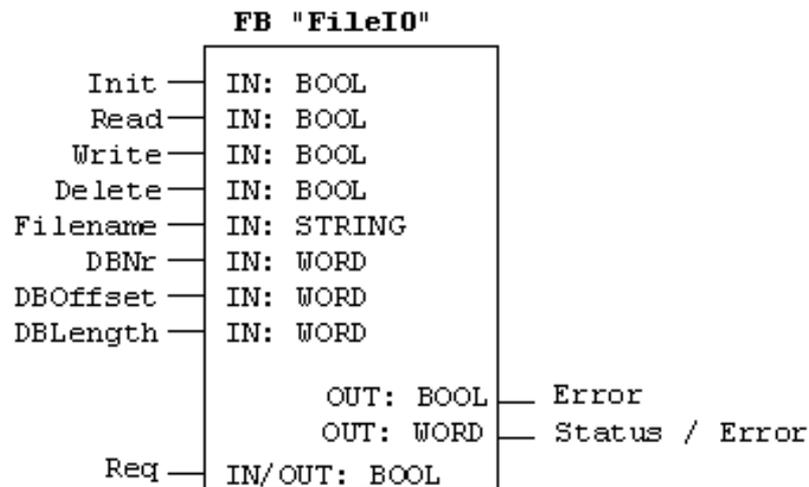
- Execute setup.bat
- Copy the FB File/I/O from the demo into the user project.
- Initialise Fil-I/O (see Example OB100)
- Use the functions of the FB File/I/O (see demo)

3 Interface

The interface of the WinAC extension **File-I/O** is realized by a function block. This function block is included in the Step7 demo program.

Function of File-I/O

The function block **File-I/O** allows to process file operations.



The flag **init** may only be set during one call of the FB (e.g. in **OB100**).

You can activate the desired functions by setting one of the flags „**Read**“, „**Write**“ or „**Delete**“.

Attention You may only use one flag at same time.

You may only process one file at same time.

The parameter „**Filename**“ defines the name of the file, which is used for the desired function.

The parameters „**DBNr**“, „**DBOffset**“ and „**DBLength**“ specify the range of a data block, which has to be written to file or which has to read from file.

The flag „**Req**“ is an IN/OUT value. To start the function the flag is set. When finished the operation the flag is reset by the **File-I/O** programm.

If the flag is reseted you can interpret „**Error**“ error „**Status**“ (see “Error codes”).

4 Function of the WinAC extension File-I/O

File-I/O does work **asynchronously**. The user has to connect the FB with the required parameters (file name, DB, ...) and the desired command (read, write, delete).

By setting the flag **Req** the command is started in background. **File-I/O** resets **Req** when processing is finished.

Process of the Step7 function block

For easy use **File-I/O** encapsulates the WinAC calls CREA_COMM and EXEC_COM.

Initializing

With activated **Init**-flag the CREA_COMM FB is called. The returned handle is stored in the static area of the instance DB.

Copying parameters

Only with set **Req** flag the FB does "normal" operation. The parameters (file name,...) are stored in the static area of the instance DB. An corresponding ANY pointer is created. Afterwards the DB parameters are checked.

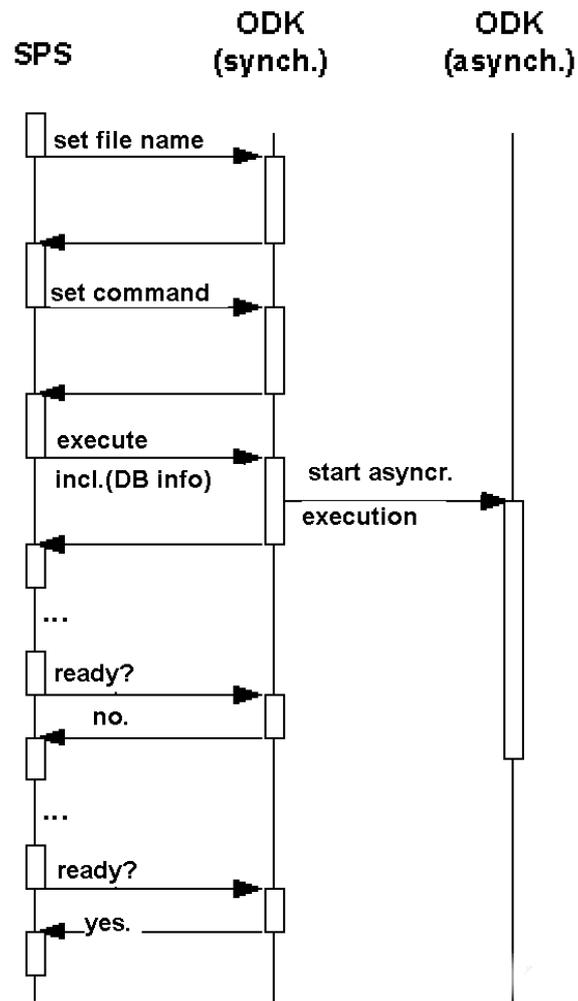
Process flow

The function of **File-I/O** is realised by some EXEC_COM calls. The main function is done asynchronously. The last step in the FB is waiting for finishing the command by the background process. The return values are set and the **Req** flag is resetted.

Process flow

The processing is carried out asynchronous, i.e. it is started from outside and takes place in parallel to the other unimpaired processes.

The processing is shown in the following diagram:



:

5 Projecting

The projecting of **File-I/O** includes two main steps: initialisation and normal processing:

5.1 Initialisation

File-I/O has to be initialised one time (e.g. in **OB100**). There for **Init** has to be set TRUE.

```
CALL "TestFileIO" , DB100
  Init      :=TRUE
  Read      :=
  Write     :=
  Delete    :=
  Filename:=
  DBNr      :=
  DBOffset:=
  DBLenght:=
  Error     :="Error"
  Status    :="Status"
  Req       :=
```

5.2 „Normal“ processing

When using **File-I/O** the flag **Init** must be FALSE. All parameters has to be set to valid values. Only one command bit has to be set to TRUE.

```
CALL "TestFileIO" , DB100
  Init      :=FALSE
  Read      :="Read"
  Write     :="Write"
  Delete    :="Delete"
  Filename:= "FileIO-Param".Filename
  DBNr      :=110
  DBOffset:=10
  DBLenght:=6
  Error     :="Error"
  Status    :="Status"
  Req       :="Req"
```

NOTICE

The WinAC File-I/O has no multitasking capability. If you need to process more than one file, you have to wait finishing one operation before starting the next one.

6 Step7 demo program

6.1 Overview

Relevant DBs / FBs are shown in the following table:

FB100	The File-I/O function block.
DB100	The corresponding instance DB
OB100	initialised the File-I/O
OB1	cyclic usage of File-I/O
DB101	collection of file names
DB110 / DB111	test data block with uniform data type (INT)
DB112 / DB113	test data block with different data types (structure is shifted by one INT value)

In addition some system function calls are used:

SFB65001	Init custom code extension
SFB65002	Process custom code extension
SFC24	TEST_DB

6.2 Usage of the demo program

You have to adjust some parameters of **File-I/O** in **OB1**:

- File name
- DBNr, DBOffset, DBLength

The variable table **VAT_1** includes some relevant variables. You can activate functions of File-I/O in an easy way and observe the return values.

By setting **Read**, **Write** or **Delete** you select a function (set only one to TRUE!). By setting **Req** you start the command.

Error and **Status** contain the return values.

7 Examples of use

File-I/O can be used in example in every form of data storing and restoring, which has to be executed directly from the SPS.

8 Error codes

The following table includes the error codes returned by the FB FileIO.

General errors WinAC open development kit

- 0x807F** An internal error occurred.
- 0x8001** An exception occurred.
- 0x8002** Input: the ANY pointer is invalid.
- 0x8003** Input: the ANY pointer range is invalid.
- 0x8004** Output: the ANY pointer is invalid.
- 0x8005** Output: the ANY pointer range is invalid.
- 0x8006** More bytes were written into the output buffer by the extension object than were allocated.
- 0x8007** ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
- 0x8008** The supplied handle value does not correspond to a valid extension object.
- 0x8009** More bytes were written into the input buffer by the extension object than were allocated.
- 0x8102** The call to CLSIDFromProgID failed.
- 0x8103** The call to CoInitializeEx failed.
- 0x8104** The call to CoCreateInstance failed.
- 0x8105** The library failed to load.
- 0x8106** A Windows response timeout occurred.
- 0x8107** Controller is in an invalid state for scheduling an OB.
- 0x8108** Schedule information for OB is invalid.
- 0x8109** Instance ID for SFB65001 call is invalid.
- 0x810A** Controller could not load proxy DLL.
- 0x810B** The WinAC controller could not create or initialize shared memory area.
- 0x810C** Attempt to access unavailable option occurred.

General error codes (from 0x8000)

- 0x0000** no error
- 0x8002** ODK-Init-error (e.g. DLL not found)
- 0x8201** internal error: unknown command
- 0x8250** internal error: job list overflow
- 0x8252** internal error: invalid counter
- 0x8255** internal error: memory range invalid
- 0x8256** internal error: invalid data block number
- 0x8259** internal error: data range invalid
- 0x825F** internal error: address error
- 0x8261** internal error: hardware error
- 0x8401** internal error: invalid asynchronous event

0x8402 internal error: queue asynchr. processor empty
0x8403 internal error: queue asynchr. processor overflow

Special error codes from File-I/O (from 0x9000)

0x9001 file name / path to long
0x9002 internal error: unknown command code
0x9011 internal error: asynchr. event not present
0x9012 internal error: Service handle not defined
0x9021 internal error: S7-read
0x9022 internal error: S7-write
0x9031 internal error: undefined status asynchr. event
0x9032 internal error: unknown status asynchr. event
0x9041 internal error: invalid call: asynchr. event is running
0x9051 error creating file for writing
0x9052 error opening file for reading
0x9053 error while writing
0x9054 error while reading
0x9055 error deleting file
0x9061 internal error: ODK-Read
0x9062 internal error: ODK-Write
0x9091 DB does not exist
0x9092 DB length exceeded
0x9999 internal error: undefined error code

9 History

Version	Date	
V 1.1	05-16-06	System requirements added
V1.2	05-09-07	Contact info on front Page Export code
V1.3	11-16-07	Change front page Change some formats and headlines Add general ODK error codes Arguments of FB now WORD (not INT)
V1.4	04-02-09	-tested with WinAC RTX 2008
V1.5	02-10-10	-tested with WinAC RTX-F 2009