SIEMENS

SIMATIC

Process Control System PCS 7 Advanced Process Functions (APF)

Function Manual

Preface

Communication 1

Material management 2

Material lot management 3

Storage location management 4

Parameter management 5

Order management 6

Archive management 7

Best before / UTC time supply 8

Data structures and data formats 9

Additional information 10

09/2014
A5E34466124-AA

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E34466124-AA
℗ 09/2014 Subject to change

Copyright © Siemens AG 2013 - 2014.
All rights reserved

Preface

Purpose of the manual

This document contains reference information for configuration of the Advanced Process Functions (APF) within the environment of SIMATIC PCS 7.

You can find additional information on APF in the following manuals:

- APF Installation and Configuration
- APF Engineering Manual
- APF Operating Manual

Core content

The following core issues are covered in this document:

- Functions and connections of the APF function blocks
- Structure of the APF user archives
- Available APF user interfaces
- Data structures

Validity

This documentation is valid for V2.0 of the APF software package.

Table of contents

Preface **3**

1 Communication **11**

1.1 FB_UA_RM 11
1.1.1 Description 11
1.1.2 Functions 12
1.1.3 I/Os 13
1.1.4 Error handling 17
1.1.5 Messaging 17

1.2 FB_UA_WM 18
1.2.1 Description 18
1.2.2 Functions 20
1.2.3 I/Os 22
1.2.4 Error handling 25
1.2.5 Messaging 26

1.3 User archives 27

1.4 Operator control and monitoring 27

2 Material management **29**

2.1 FB_XXXMA 29
2.1.1 Description 29
2.1.2 Generation 30
2.1.3 Functions 31
2.1.4 I/Os 32
2.1.5 Error handling 35
2.1.6 Messaging 36

2.2 User archives 37
2.2.1 FB_MATCLASS 37
2.2.2 FB_MATTYPE_XXX, FB_MATTYPE 38
2.2.3 FB_MATATTRIB_XXX 38
2.2.4 FB_MATERIAL_XXX 39

2.3 Operator control and monitoring 40
2.3.1 Block icons 40
2.3.2 Picture objects 41
2.3.3 Messages 41

3 Material lot management **43**

3.1 FB_MLOT 43
3.1.1 Description 43
3.1.2 Generation 44
3.1.3 Functions 45
3.1.4 I/Os 48
3.1.5 Error handling 52
3.1.6 Messaging 53

3.2 User archives....55
3.2.1 FB_MATERIAL_LOT....55
3.2.2 FB_MATATTRIB_LOT....56
3.3 Operator control and monitoring....58
3.3.1 Block icons....58
3.3.2 Picture objects....58
3.3.3 Messages....59
4 Storage location management....61
4.1 FB_STOR (FB_STORI, FB_STORR)....61
4.1.1 Description....61
4.1.2 Generation....62
4.1.3 Functions....64
4.1.4 I/Os....67
4.1.5 Error handling....71
4.1.6 Messaging....72
4.2 FB_SACT....73
4.2.1 Description....73
4.2.2 Generation....74
4.2.3 Functions....75
4.2.4 I/Os....78
4.2.5 Error handling....80
4.2.6 Messaging....81
4.3 FB_ST_AC....82
4.3.1 Description....82
4.3.2 Generation....82
4.3.3 Functions....83
4.3.4 I/Os....85
4.3.5 Error handling....88
4.3.6 Messaging....90
4.4 FB_STLOC....91
4.4.1 Description....91
4.4.2 Generation....92
4.4.3 Functions....93
4.4.4 I/Os....94
4.4.5 Error handling....101
4.4.6 Messaging....102
4.5 FB_STMAT....104
4.5.1 Description....104
4.5.2 Generation....105
4.5.3 Functions....105
4.5.4 I/Os....108
4.5.5 Error handling....116
4.5.6 Messaging....117
4.6 User archives....118
4.6.1 FB_STORGROUP....118
4.6.2 FB_STORATTRIB....119
4.6.3 FB_STORVALUE....121
4.6.4 FB_STORAGE....122

4.6.5 FB_STOR_ST_STATUS....125
4.6.6 FB_STOR_MAT_STATUS....126
4.6.7 FB_PROCEDURES....126

4.7 Operator control and monitoring....127
4.7.1 Block icons....127
4.7.2 Picture objects....128
4.7.3 Messages....129

4.8 Other....132
4.8.1 User-defined filter/sorting....132
4.8.1.1 Basic procedure....132
4.8.1.2 Configuration example C function....134
4.8.1.3 Default values....135
4.8.1.4 Restrictions....136
4.8.2 Two-step operation....136
4.8.3 Use of remainder....136
4.8.3.1 "Split" point....137
4.8.3.2 Selection of multiple storage locations....137
4.8.3.3 Output of selected storage locations....138
4.8.3.4 Switching the active storage location....138
4.8.3.5 Reserving material quantities....138
4.8.3.6 Plausibility check for storage locations....138
4.8.3.7 Search with use of remainder....138
4.8.4 Material class / material type releases....139
4.8.5 Static and dynamic procedure enables....141
4.8.6 Storage / material state....142

5 Parameter management....143

5.1 FB_XXXPA....143
5.1.1 Description....143
5.1.2 Generation....144
5.1.3 Functions....145
5.1.4 I/Os....149
5.1.5 Error handling....153
5.1.6 Messaging....153

5.2 FB_PA_ST....155
5.2.1 Description....155
5.2.2 Generation....155
5.2.3 Functions....156
5.2.4 I/Os....157

5.3 C functions for normalization/recalculation....158

5.4 User archives....160
5.4.1 FB_PARAMGROUP....160
5.4.2 FB_PARAMATTRIB_XXX....160
5.4.3 FB_PARAM_XXX....162
5.4.4 FB_PARAM_CALC_XXX....163

5.5 Operator control and monitoring....164
5.5.1 Block icons....164
5.5.2 Picture objects....164
5.5.3 Messages....165

6 Order management....169

6.1 FB_JOB....169
6.1.1 Description....169
6.1.2 Generation....171
6.1.3 Functions....172
6.1.4 I/Os....186
6.1.5 Error handling....197
6.1.6 Messaging....198

6.2 User archives....205

6.3 Operator control and monitoring....205
6.3.1 Block icons....205
6.3.2 Picture objects....206
6.3.3 Messages....207

7 Archive management....209

7.1 FB_XXXA (FB_XXXAI, FB_XXXAR)....209
7.1.1 Description....209
7.1.2 Generation....210
7.1.3 Functions....212
7.1.4 I/Os....214
7.1.5 Error handling....217
7.1.6 Messaging....218

7.2 C functions for export and backup....219
7.2.1 Overview....219
7.2.2 Data export....220
7.2.3 Data backup....223

7.3 User archives....224
7.3.1 FB_ ARCHGROUP....224
7.3.2 FB_ARCHVALUE_XXX....224
7.3.3 FB_ARCHIVING_XXX....228
7.3.4 FB_JOB_STATUS, FB_STORAGE_STATUS, FB_MATERIAL_STATUS....229
7.3.5 AENUMERATION (name in ENUM_MAT)....230

7.4 Operator control and monitoring....230

8 Best before / UTC time supply....233

8.1 FB_BBD....233
8.1.1 Description....233
8.1.2 Functions....234
8.1.3 I/Os....236
8.1.4 Error handling....237

8.2 User archives....238

8.3 Operator control and monitoring....238

9 Data structures and data formats....239

9.1 UDT_UA_R_ST....239

9.2 UDT_UA_R_REQ_ST....240

9.3 UDT_UA_W_ST....241

9.4 UDT_UA_W_REQ_ST....242
9.5 UDT_ALARM_ST....244
9.6 UDT_PA_REQ_ST....245
9.7 UDT_PA_CON_ST....245
10 Additional information....247
10.1 Configuration file....247
10.2 Data backup of the user archives....247
10.2.1 General information....247
10.2.2 Storage at the backup path....248
10.2.3 Data backup procedure....249
10.3 User archives....249

Communication

1

1.1 FB_UA_RM

1.1.1 Description

Object properties

The following table shows the object properties of the "FB_UA_RM" block:

Name	Number	Family	Module
FB_UA_RM	FB1775	APF	Communication

Area of application

The block executes the following functions:

- Reading one or more data records from the OS user archive to a data block on request
- Transferring read data to the requesting block
- Monitoring communication (generation of alarms)

Startup characteristics

The "FB_UA_RM" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

Communication on the AS level between the requesting block and the "FB_UA_RM" block is implemented with the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

The data is read through the "BRCV" block, which occurs twice in the "FB_UA_RM" block, enabling the establishment of a redundant connection.

OS tags are used to coordinate the data communication between the AS and OS. Appropriate scripts associated with the function block and its functionalities are triggered for operator control at the OS level.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

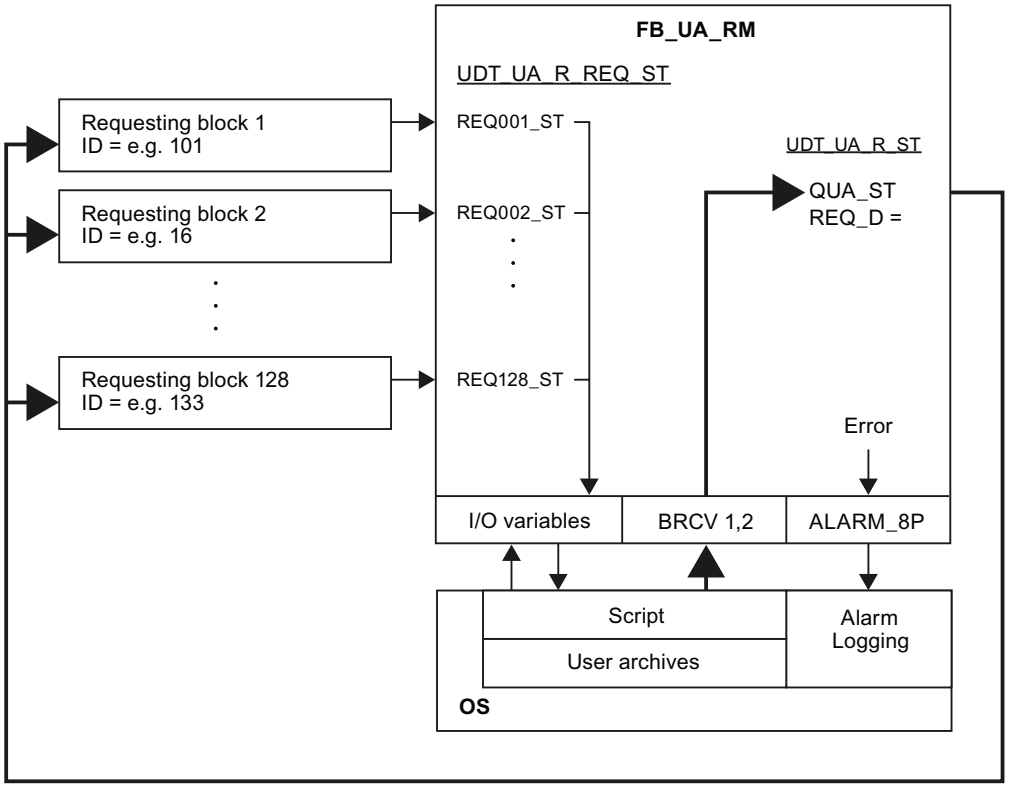
Note

For information on configuring the "FB_UA_RM" block, refer to the APF Engineering Manual.

1.1.2 Functions

Overview

The figure below provides an overview of how the "FB_UA_RM" block works:



How it works

Upon request, the "FB_UA_RM" function block reads data from an OS user archive via the "BRCV" communication block (BlockReceive, SFB13). It transfers the read data to the requesting block in the AS.

Communication between the requesting block and the "FB_UA_RM" function block occurs at the AS level using the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

The "FB_UA_RM" block can manage a maximum of 128 requests. Each request occupies a separate structure input (UDT_UA_R_REQ_ST). The requesting block transfers its uniquely identifying ID (REQ_ID) to the structure.

The read data are made available at an output structure (UDT_UA_R_ST) and identified with the REQ_ID associated with the job. As a result, the requesting block is able to recognize that the data belong to its read request.

The block monitors the 128 input structures for valid requests. A request is valid if it fulfills the following conditions:

- "REC_REQ" = 1
- "REQ_ID" <> 0
- "REC_ID" <> 0
- "ARCH_NA" <>' '
- "COLUMN" <> 0 and "NO_COL" <> 0 or "NO_COL" = 0

Each valid request is allocated a priority in the order of its occurrence. Priority is allocated as consecutive numbers from 1 to 128; 1 is the highest priority and 128 the lowest.

If no read jobs are active and there is at least one valid request pending, the request data of the input structure with the highest priority are entered in the coordination interface to the OS, and the trigger for the OS script is activated. The receive mailbox is monitored for the arrival of new data. Any new data are checked for validity and plausibility. Provided no errors have occurred, the data are copied to the output structure where they remain pending until the "REC_REQ" request is canceled by the requesting block. This ends the read job and a new read job can then be executed.

A read job is aborted if the requesting block cancels the "REC_REQ" request during the job.

1.1.3 I/Os

Input parameters

The following table shows the input parameters of the "FB_UA_RM" block:

Parameter	Description	Type	Default
RUNUPCYC	Number of startup cycles	INT	10
ID_1	Parameter ID for BRCV_1	WORD	0
ID_2	Parameter ID for BRCV_2	WORD	0
R_ID_1	Parameter R_ID for BRCV_1	DWORD	0
R_ID_2	Parameter R_ID for BRCV_2	DWORD	0
ID	ID of the block	INT	0

Parameter	Description	Type	Default
REQ001_ST …. REQ128_ST	Request 1 …. Request 128	UDT_UA_R_REQ_ST	
MSG_EVID	ALARM_8P: EV_ID	DWORD	0

Output parameters

The following table shows the output parameters of the "FB_UA_RM" block:

Parameter	Description	Type	Default
PRI_R1 … PRI_R128	Priority index for request 1 - 128	INT	0
QRUN_UP	Status display startup active = 1	BOOL	FALSE
QNOT_OK	Block flag "OK" not set, Error = 1	BOOL	FALSE
QGR_ERR	Function block group error/fault	BOOL	FALSE
QPARAMF	1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	FALSE
QARCHERR	1 = Archive error, invalid data in receive buffer or script error in the OS	BOOL	FALSE
QERR	1 = Read error, requested data and read data do not match	BOOL	FALSE
QCOM_ERR	1 = Communication error, Error message from "BRCV" coupling blocks	BOOL	FALSE
QREQ_ERR	Error ID of OS, 0 = No errors	BOOL	FALSE
QDATA_EN	1 = Auxiliary data cell active	BOOL	FALSE
QUA_ST	Output structure	UDT_UA_R_ST	
QREQ_ID	ID of the requesting block	INT	0
QARCH_NA	Archive name	STRING[16]	''
QARCH_NO	Archive number	DINT	0
QREC_ID	Record number in the user archive	DINT	0
QNO_REC	Number of rows to be read	DINT	0
QCOLUMN	Column number at which reading commences; 0 = Entire row	DINT	0
QNO_COL	Number of columns to be read	DINT	0
QOFFSET	Offset, application-specific value	DINT	0

Parameter	Description	Type	Default
QCAL_LEN	Data length calculated in the OS script, application-specific value	DINT	0
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	FALSE
QMSG_ERR	ALARM_8P: ERROR	BOOL	FALSE
MSG_STAT	ALARM_8P: STATUS	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QBR1_ERR	BRCV 1 ERROR	BOOL	FALSE
QBR1_NDR	BRCV 1 NDR	BOOL	FALSE
QBR1_STA	BRCV 1 STATUS	WORD	0
QBR1_LEN	BRCV 1 LEN	WORD	0
QBR2_ERR	BRCV 2 ERROR	BOOL	FALSE
QBR2_NDR	BRCV 2 NDR	BOOL	FALSE
QBR2_STA	BRCV 2 STATUS	WORD	0
QBR2_LEN	BRCV 2 LEN	WORD	0
REQ_ID	ID of the requesting block	INT	0
ARCH_NA	Archive name	STRING[16]	''
ARCH_NO	Archive number	DINT	0
REC_ID	Record number in the user archive	DINT	0
NO_REC	Number of rows to be read	DINT	0
COLUMN	Column number at which reading commences; 0 = Entire row	DINT	0
NO_COL	Number of columns to be read	DINT	0
WM_ID	ID of the FB_UA_WM, used with parameter management to recalculate parameters;	INT	0
PARA_1	Associated value 1 for read job, application-specific value	DINT	0
PARA_2	Associated value 2 for read job, application-specific value	DINT	0
PARA_3	Associated value 3 for read job, application-specific value	DINT	0
PARA_4	Associated value 4 for read job, application-specific value	REAL	0.0
PARA_5	Associated value 5 for read job, application-specific value	REAL	0.0
PARA_6	Associated value 6 for read job, application-specific value	REAL	0.0
DATA_1 … DATA_50	Auxiliary data cell (1 to 50) for read job, application-specific data	DWORD	0
QREQ_TYP	Type ID of the requesting block	INT	0

Parameter	Description	Type	Default
QPRI_MIN	Lowest priority index allocated	INT	0
QPRI_MAX	Higher priority index allocated	INT	0
QPRI_ACT	Number of highest-priority request	INT	0
QPRI_IND	Next priority index to be allocated	INT	0
QGEN_OS	Target generation version	DWORD	DW#16#0
GEN_AS	Generation version	DWORD	DW#16#0

In/out parameters

The following table shows the in/out parameters of the "FB_UA_RM" block:

Parameter	Description	Type	Default
REC_REQ	1 = OS read job active	BOOL	FALSE
RESET	1 = Reset read interface	BOOL	FALSE
REQ_ERR	OS script error ID	DINT	0
OS_TRIG	OS trigger	INT	0

Archive designation (transferred from requesting block)

The archive designation comprises the archive name and the archive number in the following format:

"ARCH_NA"_"ARCH_NO"

Example:

ARCH_NA = "FB_MATERIAL" and ARCH_NO = "1"

This produces the archive designation "FB_MATERIAL_001" (the archive number is always 3-digit).

If the archive designation comprises only the archive name, a "0" is entered for the "ARCH_NO" parameter.

Column specification (COLUMN and NO_COL) (transferred from requesting block)

If the value of the "COLUMN" is equal to "0", the entire row is read. In this case, the value of the "NO_COL" parameter is irrelevant.

If the value of the "COLUMN" is greater than "0", data are read starting from the column specified. In this case, the value of the "NO_COL" parameter must be greater than "0".

Auxiliary data cell (transferred from requesting block)

The application-specific auxiliary data cell is activated with the "DATA_EN" parameter.

When "DATA_EN" = "1", the bytes configured with the "DATA" pointer (a max. of 200) are entered in the communication interface to the OS as 50 "DWORD". The communication interface to the OS consists of the AS outputs "DATA_1" to "DATA_50".

1.1.4 Error handling

Errors occurring in the block are signaled via the integrated "ALARM_8P" block, and are entered in the output structure or displayed at digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals. The request signal "REC_REQ" must be canceled by the requesting block if an error occurs. This confirms receipt. No additional read jobs can be processed until the "REC_REQ" request signal has been canceled.

The following error messages can be generated:

Output structure	Output	Message text	Meaning
QERR	QERR	QERR RCV UA "arch_na""arch_no" ID "req_id" REC_ID "rec_id" NO "no_rec" COL "column"/"no_col"	Requested data and read data do not match
QARCHERR	QARCHERR	QARCHERR RCV UA "arch_na""arch_no" ID "req_id" REC_ID "rec_id" NO "no_rec" COL "column"/"no_col"	Archive error, invalid data in the receive buffer or a script error in the OS
QCOM_ERR	QCOM_ERR	QCOM_ERR RCV UA communication error	Communication error, error message from "BRCV" coupling blocks
	QBR1_ERR	QBR1_ERR communication error	Error message from first "BRCV" coupling block
	QBR2_ERR	QBR2_ERR communication error	Error message from second "BRCV" coupling block
QPARAMF	QPARAMF	QPARAMF RCV UA ID/R_ID incorrect	Parameter assignment error, incorrect ID_1, R_ID_1 or ID_2, R_ID_2
		Multiple QPARAMF RCV UA "req_id" IDs	Parameter assignment error, duplicate REQ_ID

1.1.5 Messaging

The block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following process control messages can be generated:

Message	Message class	Event	Message text
SIG_1	AS process control Message - error	Parameter assignment error, incorrect ID_1, R_ID_1 or ID_2, R_ID_2	$$BlockComment$$ QPARAMF RCV UA ID/R_ID incorrect
SIG_2	AS process control Message - fault	Communication error, error message from "BRCV" coupling blocks	$$BlockComment$$ QCOM_ERR RCV UA communication error
SIG_3	AS process control Message - fault	Archive error, invalid data in the receive buffer or a script error in the OS	$$BlockComment$$ QERR RCV UA @4%s@@5%d@ ID @8%d@ REC_ID @9%d@ NO @10%d@ COL @6%d@/@7%d@
SIG_4	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QARCHERR RCV UA @4%s@@5%d@ ID @8%d@ REC_ID @9%d@ NO @10%d@ COL @6%d@/@7%d@

Message	Message class	Event	Message text
SIG_5	AS process control Message - error	Parameter assignment error, duplicate REQ_ID	$$BlockComment$$ QPARAMF RCV Multiple UA @8%d@ IDs
SIG_6	AS process control Message - fault	QBR1_ERR	$$BlockComment$$ QBR1_ERR communication error
SIG_7	AS process control Message - fault	QBR2_ERR	$$BlockComment$$ QBR2_ERR communication error
SIG_8	No message	-	-

The following keywords are used:

- $$BlockComment$$: This keyword inserts the instance-specific block comment before the message text.
- $$HID$$ source
- $$AREA$$: OS_AREA

The following associated values are used:

Associated value	Format	Value
SD_1	-	
SD_2	-	
SD_3	-	
SD_4	@4%s@	ARCH_NA
SD_5	@5%d@	ARCH_NO
SD_6	@6%d@	COLUMN
SD_7	@7%d@	NO_COL
SD_8	@8%d@	REQ_ID
SD_9	@9%d@	REC_ID
SD_10	@10%d@	NO_REC

1.2 FB_UA_WM

1.2.1 Description

Object properties

The following table shows the object properties of the "FB_UA_WM" block:

Name	Number	Family	Module
FB_UA_WM	FB1776	APF	Communication

Area of application

The block executes the following functions:

- Writing one or more data records from an AS data block to an OS user archive on request
- Monitoring communication (generation of alarms)

Startup characteristics

The "FB_UA_WM" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

Communication on the AS level between the requesting block and the FB_UA_WM block is implemented with the "UDT_UA_W_REQ_ST" (request) and "UDT_UA_W_ST" (feedback) structures.

The data is written through the "BSEND" block, which occurs twice in the "FB_UA_WM" block, enabling the establishment of a redundant connection.

OS tags are used to coordinate the data communication between the AS and OS. Appropriate scripts associated with the function block and its functionalities are triggered for operator control at the OS level.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

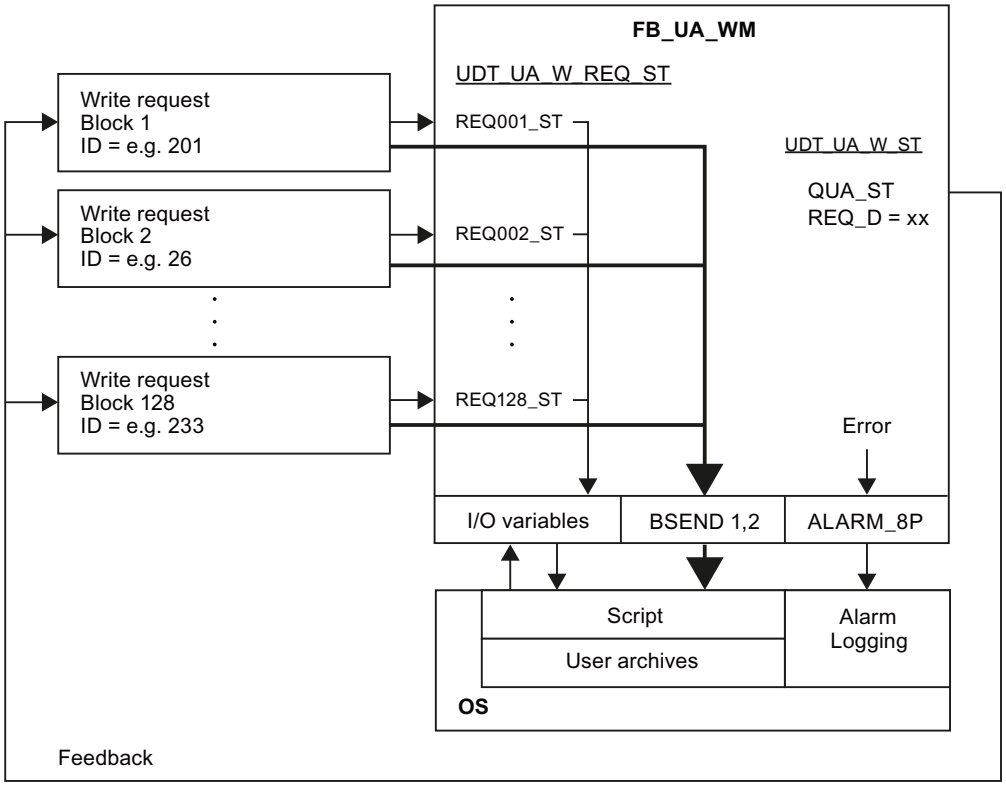
Note

For information on configuring the "FB_UA_WM" block, refer to the APF Engineering Manual.

1.2.2 Functions

Overview

The figure below provides an overview of how the "FB_UA_WM" block works:



How it works

Upon request, the "FB_UA_WM" function block writes data to an OS user archive via the "BSEND" communication block (SFB12).

The data to be written are copied from a requesting block. Communication between the requesting block and the "FB_UA_WM" function block occurs at the AS level using the "UDT_UA_W_REQ_ST" (request) and "UDT_UA_W_ST" (feedback) structures.

The "FB_UA_WM" function block can manage a maximum of 128 requests. Each request occupies a separate structure input (UDT_UA_W_REQ_ST) in the block. The requesting block transfers its uniquely identifying ID (REQ_ID) to the structure.

The feedback in response to a write request is stored in a structure output (UDT_UA_W_ST) and identified with the REQ_ID associated with the job. As a result, the requesting block is able to detect that the feedback data belong to its write request.

The block monitors the 128 input structures for valid write requests. A write request is valid if it fulfills the following conditions:

- "REC_REQ" = 1
- "MODE" <> 0
- "REQ_ID" <> 0
- "ARCH_NA" <>' '
- Either "COLUMN" <> 0 and "NO_COL" <> 0 or "NO_COL" = 0

Each valid request is allocated a priority in the order of its occurrence. Priority is allocated as consecutive numbers from 1 to 128; 1 is the highest priority and 128 the lowest.

If no write jobs are active and there is at least one valid request pending, the request data of the input structure with the highest priority are entered in the AS-OS coordination interface and the data to be sent, including the request data, are copied to the send mailbox (raw data cell). The transfer of raw data to the OS is triggered by activating the send job at the "BSEND" block (REQ = 1).

The feedback that the send job has been executed without errors by the "BSEND" block (DONE = 1) activates the trigger for the OS script.

Once the write job is completed on the OS, the OS enters the request data in the AS-OS coordination interface for confirmation (this overwrites the old request data there) and sets the "REQ_FIN" bit to "1". "REQ_FIN" = 1 compares these request data in the AS with the original request data. If and only if they are identical, are they copied to the output structure, where the "QWR_OK" is set. The data remain in the output structure until the write request at the "REC_REQ" is canceled by the requesting block. This ends the write job and a new write job can then be executed.

A write job is aborted if the requesting block cancels the "REC_REQ" request during the job.

The maximum size of the raw data cell is 16304 bytes:

Field	Format	Bytes	Address
ID	INT	2	0
REQ_ID	INT	2	2
MODE	WORD	2	4
ARCH_NA	STRING 16	18	6
ARCH_NO	DINT	4	24
REC_ID	DINT	4	28
NO_REC	DINT	4	32
COLUMN	DINT	4	36
NO_COL	DINT	4	40
GEN_AS	DWORD	4	44
PARA_1	DINT	4	48
PARA_2	DINT	4	52
PARA_3	DINT	4	56
PARA_4	REAL	4	60
PARA_5	REAL	4	64
PARA_6	REAL	4	68

Field	Format	Bytes	Address
LENGTH	WORD	2	72
DATA	ARRAY 16230 BYTES		74 to 16303

1.2.3 I/Os

Input parameters

The following table shows the input parameters of the "FB_UA_WM" block:

Parameter	Description	Type	Default
RUNUPCYC	Number of startup cycles	INT	10
ID_1	Parameter ID for BSEND_1	WORD	0
ID_2	Parameter ID for BSEND _2	WORD	0
R_ID_1	Parameter R_ID for BSEND _1	DWORD	0
R_ID_2	Parameter R_ID for BSEND _2	DWORD	0
ID	ID of the block	INT	0
REQ001_ST …. REQ128_ST	Write request 1 …. Write request 128	UDT_UA_W_REQ_ST	
MSG_EVID	ALARM_8P: EV_ID	DWORD	0

Output parameters

The following table shows the output parameters of the "FB_UA_WM" block:

Parameter	Description	Type	Default
PRI_R1 … PRI_R128	Priority index for write job 1 - 128	INT	0
QRUN_UP	Status display startup active = 1	BOOL	FALSE
QNOT_OK	Block flag "OK" not set, Error = 1	BOOL	FALSE
QGR_ERR	Function block group error/fault	BOOL	FALSE
QPARAMF	1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	FALSE
QARCHERR	1 = Archive error Job data in the WinCC communication interface reset by the OS or script error in the OS	BOOL	FALSE
QERR	1 = Write error Job data in WinCC communication interface do not correspond to original job data.	BOOL	FALSE

Parameter	Description	Type	Default
QCOM_ERR	1 = Communication error Error message from "BSEND" coupling blocks	BOOL	FALSE
QREQ_ERR	Error ID from OS 0 = No errors	BOOL	FALSE
QUA_ST	Output structure with job feedback	UDT_UA_W_ST	
QREQ_ID	ID of the requesting block	INT	0
QMODE	Operating mode	WORD	0
QARCH_NA	Archive name	STRING[16]	''
QARCH_NO	Archive number	DINT	0
QCOLUMN	Record number in the user archive	DINT	0
QREC_ID	Number of rows to be written	DINT	0
QNO_REC	Column number at which writing commences; 0 = Entire row	DINT	0
QNO_COL	Number of columns to be written	DINT	0
QPARA_1	Associated value 1 for write job, application-specific value	DINT	0
QPARA_2	Associated value 2 for write job, application-specific value	DINT	0
QPARA_3	Associated value 3 for write job, application-specific value	DINT	0
QPARA_4	Associated value 4 for write job, application-specific value	REAL	0.0
QPARA_5	Associated value 5 for write job, application-specific value	REAL	0.0
QPARA_6	Associated value 6 for write job, application-specific value	REAL	0.0
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	FALSE
QMSG_ERR	ALARM_8P: ERROR	BOOL	FALSE
MSG_STAT	ALARM_8P: STATUS	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QBS1_ERR	BSEND 1 ERROR	BOOL	FALSE
QBS1_NDR	BSEND 1 NDR	BOOL	FALSE
QBS1_STA	BSEND 1 STATUS	WORD	0
QBS1_LEN	BSEND 1 LEN	WORD	0
QBS2_ERR	BSEND 2 ERROR	BOOL	FALSE
QBS2_NDR	BSEND 2 NDR	BOOL	FALSE
QBS2_STA	BSEND 2 STATUS	WORD	0
QBS2_LEN	BSEND 2 LEN	WORD	0
REQ_ID	ID of the requesting block	INT	0
MODE	Operating mode	WORD	0
ARCH_NA	Archive name	STRING[16]	''
ARCH_NO	Archive number	DINT	0
REC_ID	Record number in the user archive	DINT	0

Parameter	Description	Type	Default
NO_REC	Number of rows to be written	DINT	0
COLUMN	Column number at which writing commences; 0 = Entire row	DINT	0
NO_COL	Number of columns to be written	DINT	0
PARA_1	Associated value 1 for write job, application-specific value	DINT	0
PARA_2	Associated value 2 for write job, application-specific value	DINT	0
PARA_3	Associated value 3 for write job, application-specific value	DINT	0
PARA_4	Associated value 4 for write job, application-specific value	REAL	0.0
PARA_5	Associated value 5 for write job, application-specific value	REAL	0.0
PARA_6	Associated value 6 for write job, application-specific value	REAL	0.0
QREQ_TYP	Type ID of the requesting block	INT	0
QPRI_MIN	Lowest priority index allocated	INT	0
QPRI_MAX	Higher priority index allocated	INT	0
QPRI_ACT	Number of highest-priority request	INT	0
QPRI_IND	Next priority index to be allocated	INT	0
QGEN_AS	Generation version	DWORD	0
GEN_OS	Target generation version	DWORD	0

In/out parameters

The following table shows the in/out parameters of the "FB_UA_WM" block:

Parameter	Description	Type	Default
REC_REQ	1 = OS write job active	BOOL	FALSE
REQ_FIN	1 = OS write job complete	BOOL	FALSE
RESET	1 = Reset job interface	BOOL	FALSE
REQ_ERR	OS script error ID	DINT	0
OS_TRIG	OS trigger	INT	0

Archive designation (transferred from requesting block)

The archive designation comprises the archive name and the archive number in the following format:

"ARCH_NA"_"ARCH_NO"

Example:

ARCH_NA = "FB_MATERIAL" and ARCH_NO = "1"

This produces the archive designation "FB_MATERIAL_001" (the archive number is always 3-digit).

If the archive designation comprises only the archive name, a "0" is entered for the "ARCH_NO" parameter.

Column specification (COLUMN and NO_COL) (transferred from requesting block)

If the value of the "COLUMN" is equal to "0", the entire row is written. In this case, the value of the "NO_COL" parameter is irrelevant.

If the value of the "COLUMN" is greater than "0", data are written starting from the column specified. In this case, the value of the "NO_COL" parameter must be > 0.

"MODE" parameter (transferred from requesting block)

The "MODE" parameter is evaluated by an OS script.

MODE	Meaning
1	Create new data record
2	Overwrite existing data record
3	Delete data record
10	FB_STOR
11	FB_SACT
12	FB_ST_AC: Delete booking with material ID
13	FB_ST_AC: Delete booking without material ID
14	FB_ST_AC: Delete material ID

1.2.4 Error handling

Errors occurring in the block are signaled via the integrated "ALARM_8P" block, and are entered in the output structure or displayed at digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals. The request signal "REC_REQ" must be canceled by the requesting block if an error occurs. This confirms receipt of the error information. No additional write jobs can be processed until the "REC_REQ" request signal has been canceled.

The following error messages can be generated:

Output structure	Output	Message text	Meaning
QERR	QERR	QERR SEND UA "arch_na""arch_no" ID "req_id" REC_ID "rec_id" NO "no_rec" COL "column"/"no_col"	Error, job data in WinCC communication interface do not correspond to original job data
QARCHERR	QARCHERR	QARCHERR SEND UA "arch_na""arch_no" ID "req_id" REC_ID "rec_id" NO "no_rec" COL "column"/"no_col"	Archive error; job data in the WinCC communication interface reset by the OS, or script error occurred in OS

Output structure	Output	Message text	Meaning
QCOM_ERR	QCOM_ERR	QCOM_ERR SEND UA communication error	Communication error, error message from "BSEND" coupling blocks
	QBS1_ERR	QBS1_ERR communication error	Error message from 1st "BSEND" coupling block
	QBS2_ERR	QBS2_ERR communication error	Error message from 2nd "BSEND" coupling block
QPARAMF	QPARAMF	QPARAMF SEND UA ID/R_ID incorrect	Parameter assignment error, incorrect ID_1, R_ID_1 or ID_2, R_ID_2
		Multiple QPARAMF SEND UA "req_id" IDs	Parameter assignment error, duplicate REQ_ID

1.2.5 Messaging

The block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following process control messages can be generated:

Message	Message class	Event	Message text
SIG_1	AS process control Message - error	Parameter assignment error, incorrect ID_1, R_ID_1 or ID_2, R_ID_2	$$BlockComment$$ QPARAMF SEND UA ID/R_ID incorrect
SIG_2	AS process control Message - fault	Communication error, error message from "BSEND" coupling blocks	$$BlockComment$$ QCOM_ERR SEND UA communication error
SIG_3	AS process control Message - fault	Error; job data in WinCC communication interface do not correspond to original job data	$$BlockComment$$ QERR SEND UA @4%s@@5%d@ ID @8%d@ REC_ID @9%d@ NO @10%d@ COL @6%d@/ @7%d@
SIG_4	AS process control Message - fault	Archive error; job data in the WinCC communication interface reset by the OS, or script error occurred in OS	$$BlockComment$$ QARCHERR SEND UA @4%s@@5%d@ ID @8%d@ REC_ID @9%d@ NO @10%d@ COL @6%d@/ @7%d@
SIG_5	AS process control Message - error	Parameter assignment error, duplicate REQ_ID	$$BlockComment$$ multiple QPARAMF SEND UA @8%d@ IDs
SIG_6	AS process control Message - fault	QBS1_ERR	$$BlockComment$$ QBS1_ERR communication error
SIG_7	AS process control Message - fault	QBS2_ERR	$$BlockComment$$ QBS2_ERR communication error
SIG_8	No message		

The following keywords are used:

- $$BlockComment$$: This keyword inserts the instance-specific block comment before the message text.
- $$HID$$ source
- $$AREA$$: OS_AREA

The following associated values are used:

Associated value	Format	Value
SD_1	-	
SD_2	-	
SD_3	-	
SD_4	@4%s@	ARCH_NA
SD_5	@5%d@	ARCH_NO
SD_6	@6%d@	COLUMN
SD_7	@7%d@	NO_COL
SD_8	@8%d@	REQ_ID
SD_9	@9%d@	REC_ID
SD_10	@10%d@	NO_REC

1.3 User archives

The user archives of the corresponding APF modules are used to access the data records.

1.4 Operator control and monitoring

No picture objects are assigned to the "FB_UA_RM" or "FB_UA_WM" blocks. The blocks are operated on the OS using the configured APF modules.

Material management

2

2.1 FB_XXXMA

2.1.1 Description

Object properties

The following table shows the object properties of FB_XXXMA:

Name	Number	Family	Module
FB_XXXMA	FB822	APF	Material management

Area of application

The block is used to request and provide material master data.

Startup characteristics

The "FB_XXXMA" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

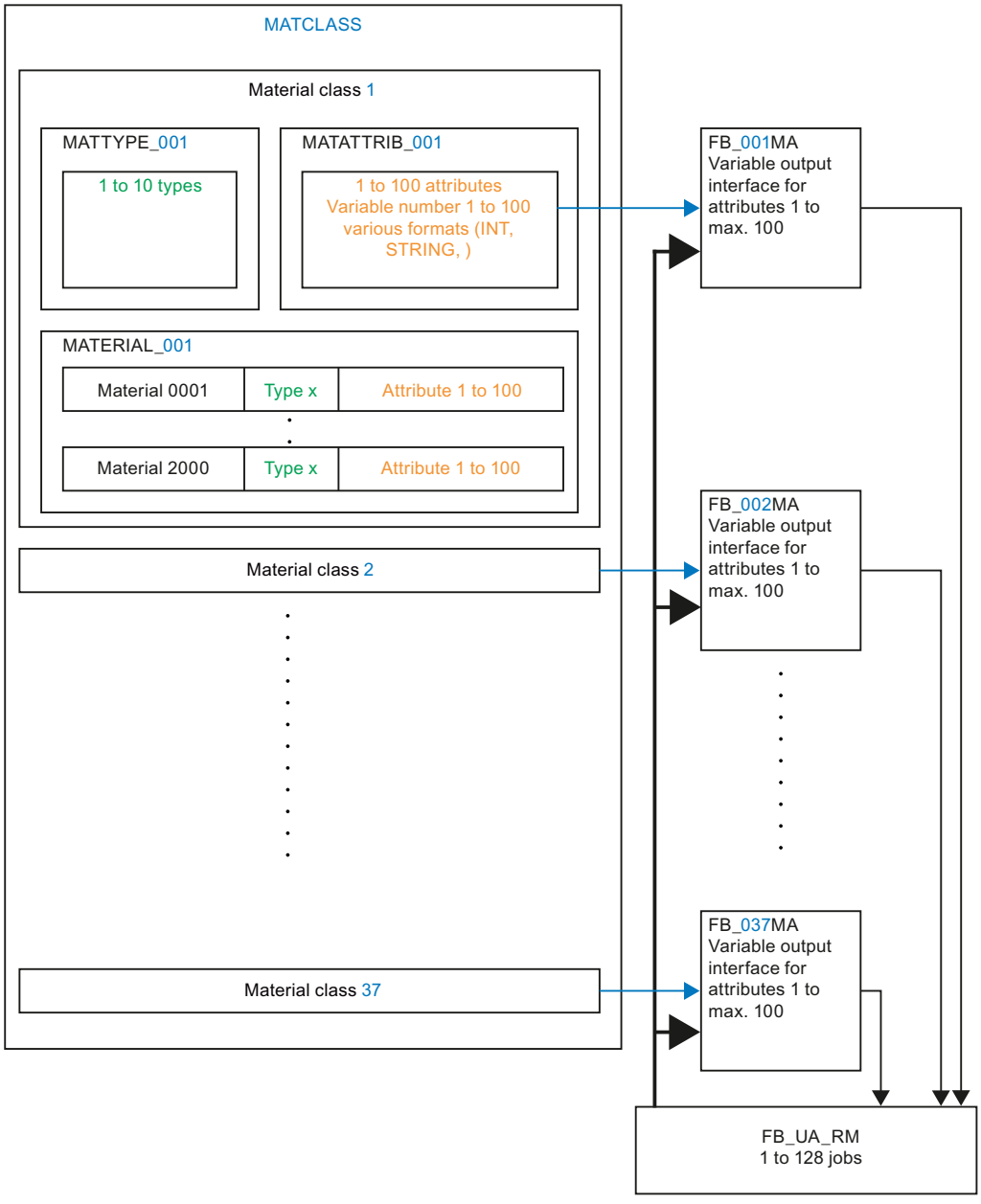
Corresponding scripts for the function block and its functionalities on the AS level are available for handling on the OS level. The "ALARM_8P" structure is used to implement data communication with the OS message system.

2.1.2 Generation

For each class of material created using the engineering tool in the "FB_MATCLASS" user archive, a block of the type "FB_XXXMA" is generated and saved in the master data library. The generated blocks are numbered consecutively.

The block outputs depend on the structure of the "FB_MATATTRIB_XXX" user archive, in which the attributes of a material class are defined. The corresponding block needs to be generated again if the attributes are changed.

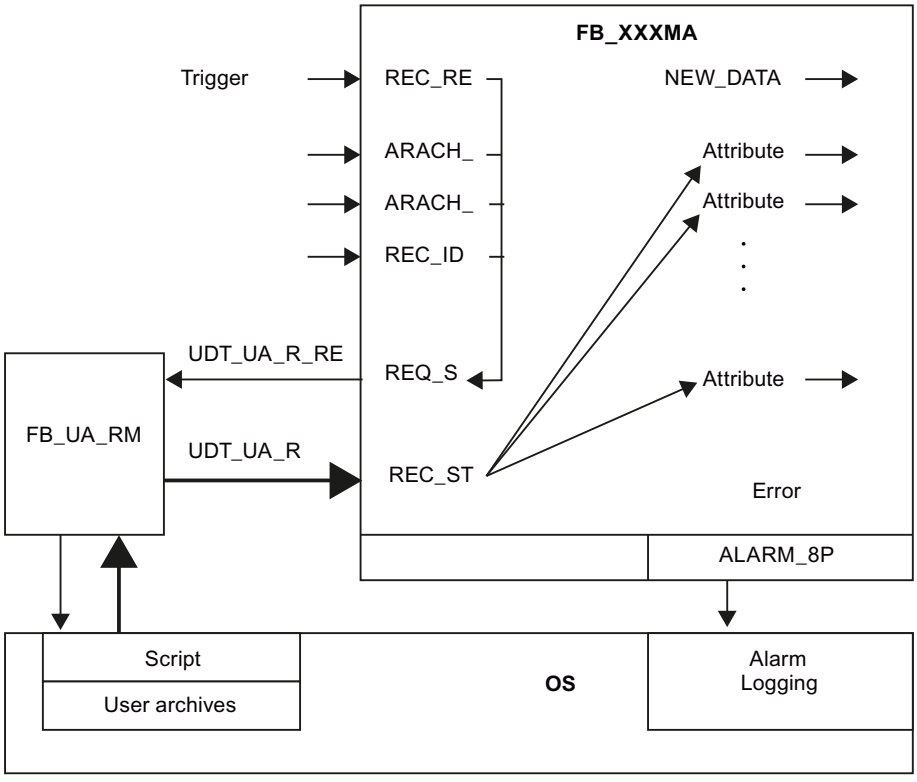
The figure below provides an overview of the structure of "FB_XXXMA" blocks:



2.1.3 Functions

Overview

The figure below provides an overview of how the "FB_XXXMA" block works:



How it works

Following a request by a positive edge at input "REC_REQ", the request data (REC_ID (Material ID), ARCH_NA /ARCH_NO etc.) are written to the output structure "UDT_UA_R_REQ_ST". The read job is activated in the structure by setting the "REC_REQ" bit.

Once the read job has been executed, the communication block "FB_UA_RM" writes the request data (including REQ_ID) and the pointer to the material data record to the "UDT_UA_R_ST" input structure of the block, where it also sets the "NEW_DATA" input bit.

Note

For detailed information on the "UDT_UA_R_REQ_ST" and "UDT_UA_R_ST" structures, see section Data structures and data formats (Page 239).

The input data are only assigned to the block's read job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" in the input structure.

Once the input structure data have been assigned to the read job, the data are checked for validity and plausibility. The system evaluates the following:

- Whether the request data in the input and output structure match
- Whether the data length calculated corresponds to the data length of the outputs at the block

If the data are plausible and valid, the "NEW_DATA" output is set and the data are written from the "FB_UA_RM" communication block to the outputs of the block, where they are available for further processing.

The "NEW_DATA" output is reset by resetting the "REC_REQ" request input.

The "FB_UA_RM" communication block reports any errors it detects during execution of the read job to the "FB_XXXMA" block. The "FB_XXXMA" block deactivates the read job if an error occurs ("REC_REQ" = 0). This also happens if the "FB_XXXMA" block itself detects an error.

The "COUNT" input value defines how often the read job is to be repeated if an error occurs (COUNT = 0 -> no repetition). If the number of read job repetitions is not reached, the job is reactivated by the "FB_XXXMA" block after a wait cycle.

An error message is output after the last repetition.

A new read job can be activated by setting the request signal at the "REC_REQ" input again.

A positive edge at the "RESET" input resets the material data at the block output, the "NEW_DATA" output, and the error messages.

2.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_XXXMA" block:

Parameter	Description	Type	Default
SAMPLE_T	Sampling time [s] The sampling time is set by the CFC editor	REAL	1
RUNUPCYC	Number of cycles following CPU startup during which the block is not yet working and messages are suppressed.	INT	10
ID	ID of the block The ID must be unique	INT	0
ARCH_NA	Archive name	STRING[16]	''
ARCH_NO	Archive number	DINT	0
REC_ID	Data record number in user archive	DINT	0
BA_NA	Batch name	STRING[32]	''
BA_ID	Batch ID	DWORD	0
COUNT	Number of repetitions if an error occurs	INT	1
REC_REQ	Trigger read job	BOOL	0
RESET	Reset read interface	BOOL	0

Parameter	Description	Type	Default
TIME_MON	Monitoring time for read job	REAL	30.0
REC_ST	Feedback structure from FB_UA_RM	UDT_UA_R_ST	
MSG_EVID	ALARM_8P option: EV_ID	DWORD	0

Output parameters

The following table shows the output parameters of the "FB_XXXMA" block:

Parameter	Description	Type	Default
QRUN_UP	Status display, startup active = 1	BOOL	0
QNOT_OK	Block flag "OK" not set, Error = 1	BOOL	0
QGR_ERR	Group error/fault	BOOL	0
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0
QRELEASED	1= Function block released for processing	BOOL	0
QERR	1 = Read error, Header: requested data and read data do not match	BOOL	0
QERR_VER	1 = Version check error	BOOL	0
QMON_ERR	1 = Read job monitoring timeout	BOOL	0
QLEN_ERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0
QRM_ERR	FB_UA_RM: 1 = Read error, requested data and read data do not match	BOOL	0
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0
QREQ_ACT	Read job active	BOOL	0
NEW_DATA	Data have been read	BOOL	0
REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST	
QID	Read data: ID	INT	0
QARCH_NA	Read data: Archive name	STRING[16]	''
QARCH_NO	Read data: Archive number	DINT	0
QREC_ID	Read data: Data record number in user archive	DINT	0
QNO_REC	Read data: Number of rows to be read	DINT	0
QCOLUMN	Read data: Column number at which reading commences	DINT	0

Parameter	Description	Type	Default
QNO_COL	Read data: Number of columns to be read	DINT	0
QOFFSET	Read data: Offset, application-specific value	DINT	0
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0
QRESERVE	Read data: Reserve	DINT	0
QNAME	Read data: Material name	STRING[64]	''
QDESCR	Read data: Description	STRING[128]	''
QUOM	Read data: Unit	STRING[16]	''
QVERSION	Read data: Version	REAL	0.0
QCH_USER	Read data: Login name of operator who performed the last change	STRING[32]	''
QCH_DAT	Read data: Time of change part 1	DWORD	16#08123123
QCH_DATB	Read data: Time of change part 2	DWORD	16#59590000
QTYPE	Read data: Material type	DINT	0
QSHLIFE	Read data: first read attribute, minimum shelf life	DINT	0
INT1	Read data: Example, material process value 1	DINT	0
TEXT	Read data: Example, material process value 2	STRING[8]	''
REAL1	Read data: Example, material process value 3	REAL	0.0
REAL2	Read data: Example, material process value 4	REAL	0.0
BOOL1	Read data: Example, material process value 5	BOOL	0
INVISIBLE	Read data: Example, material process value 6	BOOL	0
REAL3	Read data: Example, material process value 7	REAL	0.0

Parameter	Description	Type	Default
DATE1	Read data: Example, material process value 8	DWORD	0
DATE1B	Read data: Example, material process value 9	DWORD	0
TIME1	Read data: Example, material process value 10	DWORD	0
TIME1B	Read data: Example, material process value 11	DWORD	0
DT1	Read data: Example, material process value 12	DWORD	0
DT1B	Read data: Example, material process value 13	DWORD	0
ENUM1	Read data: Example, material process value 14	DINT	0
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST	-
MSG_STAT	ALARM_8P: STATUS	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	0
QMSG_ERR	ALARM_8P: ERROR	BOOL	0

In/out parameters

The following table shows the in/out parameters of the "FB_XXXMA" block:

Parameter	Description	Type	Default
APF_AS	APF function block version in AS	WORD	
APF_OS	APF function block version in OS	WORD	
GEN_AS	Generation information in AS	DWORD	
GEN_OS	Generation information in OS	DWORD	

2.1.5 Error handling

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. The error bits are available at the block as an output signal. In the event of an error, the request signal "REC_REQ" must be reset to confirm that the error information has been processed.

The following error messages can be generated:

Output	Message text	Meaning
QERR	QERR data error read job ID "id" REC_ID "rec_id"	Requested data and read data do not match
QMON_ERR	QMON_ERR runtime error read job ID "id" REC_ID "rec_id"	Read job monitoring timeout
QLEN_ERR	QLEN_ERR data block length of receive data is not correct ID "id" REC_ID "rec_id"	Length of the read data does not match the length of the block outputs
QPARAMF	QPARAMF parameter assignment error ID "id" REC_ID "rec_id"	Parameter assignment error ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0
QERR_VER	QERR_VER Version check error	Error in the version information

2.1.6 Messaging

The "FB_XXXMA" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The version error will additionally be reported through the ALARM_8P (SIG_5).

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QLEN_ERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@
SIG_2	AS process control Message - fault	Read job monitoring timeout	$$BlockComment$$ QMON_ERR runtime error read job ID @4%d@ REC_ID @5%d@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QERR data error read job ID @4%d@ REC_ID @5%d@
SIG_4	AS process control Message - fault	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@ REC_ID @5%d@
SIG_5	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER error in version check
SIG_6	No message		
SIG_7	No message		
SIG_8	No message		

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	-	
SD_7	-	
SD_8	-	
SD_9	-	
SD_10	-	

2.2 User archives

2.2.1 FB_MATCLASS

The material classes are defined in the "FB_MATCLASS" user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of the class	INTEGER 32
NAME	Name of the class	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

2.2.2 FB_MATTYPE_XXX, FB_MATTYPE

The material types of a material class are defined in the "FB_MATTYPE_XXX" user archives. An archive is assigned to each material class and numbered accordingly. There is also a "collective archive" without _XXX that contains one instance of each material type. The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID of the type	INTEGER 32
NAME	Name of the type	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

2.2.3 FB_MATATTRIB_XXX

Up to 100 material properties (attributes) of a material class are defined in the "FB_MATATTRIB_XXX" user archives. An archive is assigned to each material class and numbered accordingly.

The archive has the following structure:

Element	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Name of the attribute	STRING 32
UOM	Unit of measure for the attribute	STRING 16
FORMAT	Format of the attribute	INTEGER 32
ENUM_MAT	Name/coding of an enumeration (Name of the PCS7 enumeration)	STRING 16
HL	High limit of operator control for numerical values	DOUBLE
LL	Low limit of operator control for numerical values	DOUBLE
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the output in the corresponding function block	STRING 20
VISIBLE	Display of the attribute in the process picture	INTEGER 32

Format coding

The following table shows the coding of the attribute in "FORMAT":

Integer value	Format
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places

Integer value	Format
5	REAL with 4 decimal places
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
20xx	DATE
21xx	TIME
22xx	DATE AND TIME
30	BOOL
40	ENUMERATION

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

2.2.4 FB_MATERIAL_XXX

The materials of a material class and their attribute values are saved in the "FB_MATERIAL_XXX" user archives.

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID of material	INTEGER 32*
NAME	Material name	STRING 64
DESCR	Description of material	STRING 128
UOM	Unit of measure for the material	STRING 16

Element	Description	Format
VERSION	Current version	REAL
CHANGEUSER	Name of the operator who made the last change	STRING 32
CHANGEDATE	Date of the last change	DATE
TYPE	Assignment to a material type	-
ATTRIB_QSHLIFE ATTRIB_<ioname> ….. ATTRIB_<ioname>	Value of property ….. ….. Value of property	Represents the formats of the attributes from the corresponding "FB_MATATTRIB_XXX" user archive

Note

*The manual input in the editor dialog is limited to 2,000,000,000.

The engineering tool creates the first attribute as the shelf life (INTEGER). The shelf life is the time, for example, in hours, days or months, for which a base material will keep. It can be used in the material lot to calculate the best before date.

The remaining attributes are allocated to the column name "ATTRIB_" with the "FCT_BLOCK_IO_NAME" of the FB_MATATTRIB_XXX table.

Note

Forming the attribute column name (a maximum of 20 characters) from "ATTRIB_" and "<ioname>" limits the actual useful length of the IO name to 13 characters.

2.3 Operator control and monitoring

2.3.1 Block icons

There is no block icon for the "FB_XXXMA" block. This block is always controlled indirectly from other blocks.

2.3.2 Picture objects

The interface for material management operator control and monitoring consists of the following picture objects:

- FB_MAT_ED.PDL
  You can use the picture "FB_MAT_ED.pdl" to create, change, delete and copy material master data.
- @FB_MAT_ED_WINDOW.PDL
  You can use the picture "FB_MAT_ED_WINDOW.pd" to edit the attribute parameters of the materials. This picture is an integrated window in the "FB_MAT_ED.pdl" picture and cannot be used separately.

Note

The process pictures are pre-configured and do not need to be edited.

For detailed information on interface operator control and monitoring in process mode, see the APF Operator Manual.

2.3.3 Messages

The following messages can be output:

Message class	Event	Message text
Operator message	Save button	@7%s@: @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ material: @6%s@
Operator message	Save button	@7%s@: @2%s@: new = @5%s@ @8%s@ old = @4%s@ @8%s@ material: @6%s@
Operator message	Save button	@7%s@: material @6%s@ changed
Operator message	Save button	@7%s@: material @6%s@ created
Operator message	Delete button	@7%s@: material @6%s@ deleted
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
OS process control error	Cyclic backup action: The export file could not be created.	@7%s@: Backup file (@2%s@) could not be exported
OS process control error	Cyclic backup action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the backup file
OS process control error	Cyclic backup action: Directory 1 could not be created	@7%s@: Backup directory 1 does not exist / could not be created
OS process control error	Cyclic backup action: Directory 2 could not be created	@7%s@: Backup directory 2 does not exist / could not be created
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 1 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 2 (@2%s@) could not be copied

Message class	Event	Message text
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 1 is corrupt
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 2 is corrupt
OS process control error	Cyclic backup action: Call the external backup program failed	@7%s@: Initialization of the backup program failed

Message associated values

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%d@	Not used
Message associated value 2	@2%s@	Material attribute / backup file
Message associated value 3	@3%d@	Not used
Message associated value 4	@4%s/d@	Old value (string or number)
Message associated value 5	@5%s/d@	New value (string or number)
Message associated value 6	@6%s@	Material + ID, here: material name (ID)
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	Unit / UOM
Message associated value 9	@9%s@	Not used
Message associated value 10	@10%s@	Source, here: material management

For additional information, refer to the PCS 7 online help.

Material lot management

3

3.1 FB_MLOT

3.1.1 Description

Object properties

The following table shows the object properties of FB_MLOT:

Name	Number	Family	Module
FB_MLOT	FB1778	APF	Material lot management

Area of application

The "FB_MLOT" block requests and provides material lot data in the AS.

Startup characteristics

The "FB_MLOT" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

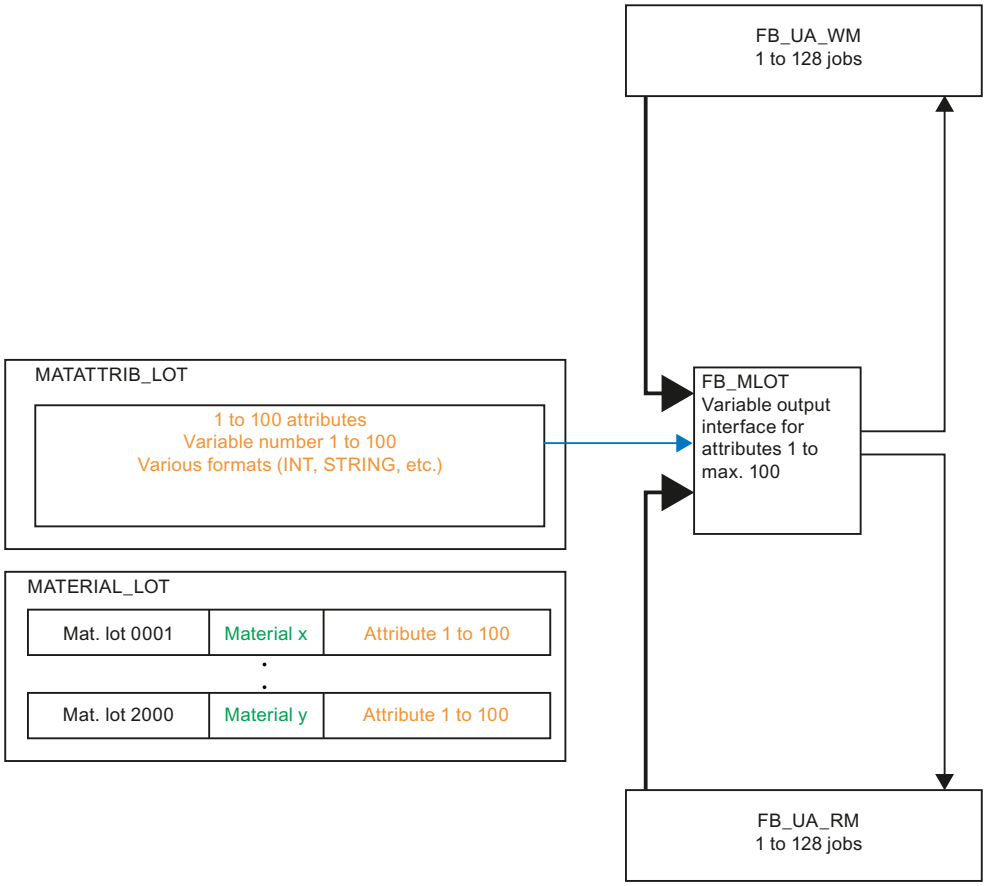
Data exchange with the "FB_UA_WM" communication block is performed via the "UDT_UA_W_REQ_ST" (request) and "UDT_UA_W_ST" (feedback) structures.

Corresponding scripts for the function block and its functionalities on the AS level are available for handling on the OS level. The "ALARM_8P" structure is used to implement data communication with the OS message system.

3.1.2 Generation

An "FB_MLOT" block is generated and saved in the master data library for the attributes of the material lots (100 max.) created using the engineering tool in the "FB_MATATTRIB_LOT" user archive. An output is created for each attribute in the "FB_MLOT" block.

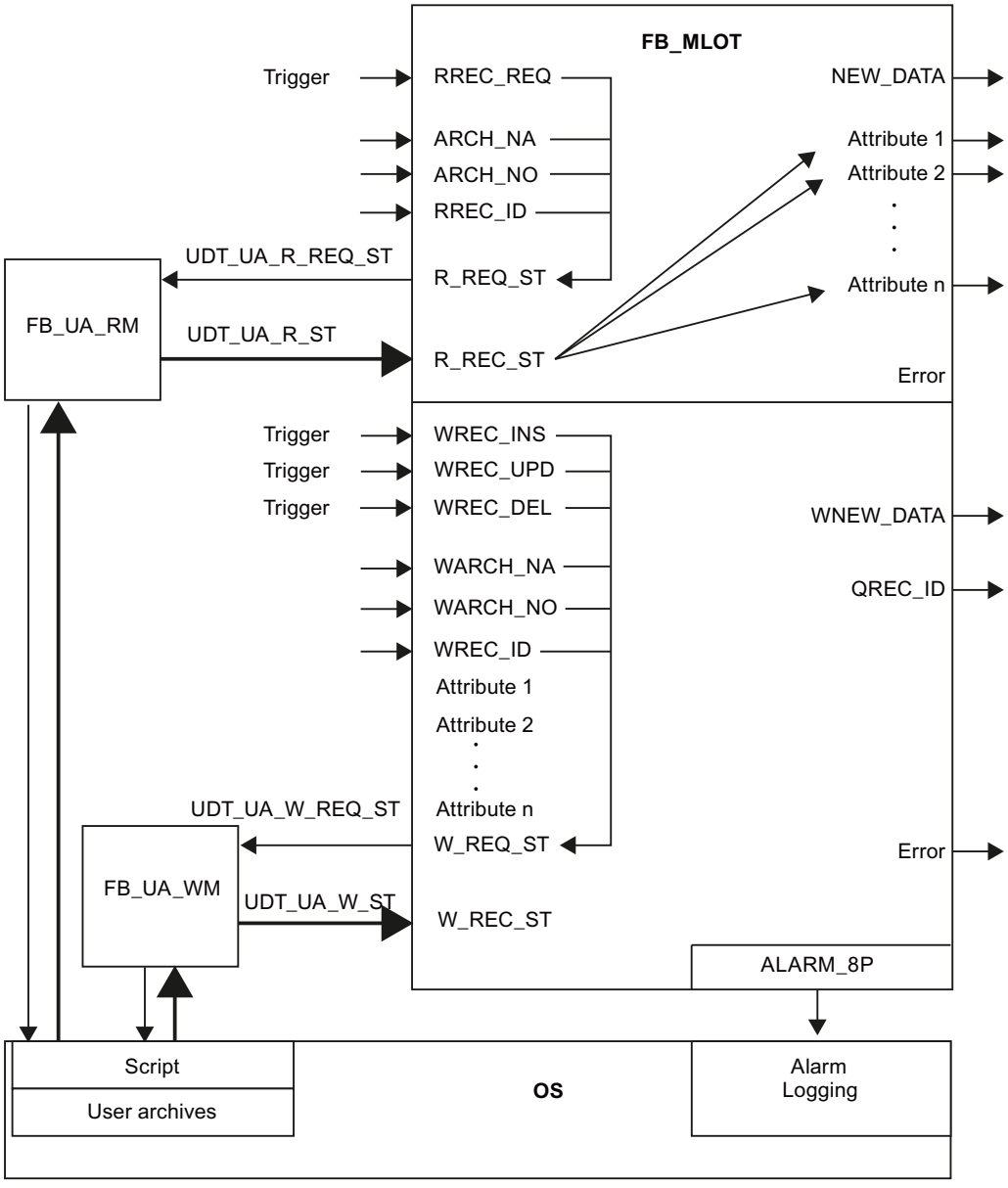
The figure below provides an overview of the "FB_MLOT" block structure:



3.1.3 Functions

Overview

The figure below provides an overview of how the "FB_MLOT" block works:



Reading data

Following a request by a positive edge at request input "RREC_REQ", the request data (RREC_ID (material lot ID), ARCH_NA /ARCH_NO, etc.) are written to the output structure "UDT_UA_R_REQ_ST". The read job is activated in the structure by setting the "REC_REQ" bit.

Once the read job has been executed, the communication block "FB_UA_RM" writes the request data (including RREQ_ID) and the pointer to the material data record to the "UDT_UA_R_ST" input structure of the block, where it also sets the "NEW_DATA" input bit.

Note

For detailed information on the "UDT_UA_R_REQ_ST" and "UDT_UA_R_ST" structures, see section Data structures and data formats (Page 239).

The input data are only assigned to the block's read job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" in the input structure.

Once the input structure data have been assigned to the read job, the data are checked for validity and plausibility. The system evaluates the following:

- Whether the request data in the input and output structure match
- Whether the data length calculated corresponds to the data length of the outputs at the block

If the data are plausible and valid, the "NEW_DATA" output is set in the FB_MLOT and the data are written from the "FB_UA_RM" communication block to the outputs of the block, where they are available for further processing.

The "NEW_DATA" output is reset by resetting the "RREC_REQ" request input.

The "FB_UA_RM" communication block reports any errors it detects during execution of the read job to the "FB_MLOT" block. The "FB_MLOT" block deactivates the read job if an error occurs ("RREC_REQ" = 0). This also happens if the "FB_MLOT" block itself detects an error.

A positive edge at the "RRESET" input resets the material data at the block output, the "NEW_DATA" output and the error messages.

Writing data

This block allows you to create new material lots and modify or delete existing lots. All the attributes required are available as block inputs.

When a new material lot is created, the system checks that the material lot name entered at the block and the material lot ID are unique. When a material lot is modified or deleted, only the material lot ID set at the block must be unique.

Following a request by a positive edge at the request input ("WREQ_INS", "WREQ_UPD" or "WREQ_DEL"), the request data (REC_ID, ARCH_NA / ARCH_NO, etc.) and the attribute input data are written to the output structure "UDT_UA_W_REQ_ST". The write job is activated in the structure by setting the "REC_REQ" bit.

Note

For detailed information on the "UDT_UA_W_REQ_ST" and "UDT_UA_W_ST" structures, see section Data structures and data formats (Page 239).

The following inputs are required when you insert ("WREQ_INS") or update ("WREQ_UPD") a data record:

- INAME
- DESCR
- UOM
- IVERSION
- MATID
- LOSSTAT / LOTSTAT
- MHD / BBD
- MHDB / BBDB
- INT1 (attribute as an example)
- INT2 (attribute as an example)

These data are not required for deleting ("WREQ_DEL").

The "INAME" input is not analyzed upon change or deletion. The data record to be changed or deleted is selected on the basis of the value at the "WREC_ID" input.

Once the write job has been processed, the "FB_UA_WM" block writes the confirmation data (including REQ_ID) to the "UDT_UA_W_ST" input structure of the block and sets the "QWR_OK" bit.

The input data are only assigned to the block's write job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" of the input structure.

Once the input structure data have been assigned to the write job and the "QWR_OK" bit has been set, the data are checked for validity and plausibility. The system checks whether the request data in the input and output structure match.

If the data are plausible and valid, the "WR_OK" output is set.

The "WR_OK" output is reset by resetting the "WREC_REQ" request input.

The "FB_UA_WM" communication block reports any errors it detects during execution of the write job to the "FB_MLOT" block. The "FB_MLOT" block deactivates the write job if an error occurs ("REC_REQ" = 0). This also happens if the "FB_MLOT" block itself detects an error.

A positive edge at the "WRESET" input resets the "WR_OK" output and the error messages relating to the write job.

The material lot ID (WREC_ID) can also be assigned by the OS during insertion (WREQ_INS). The setting is made with the "OS_REC_ID" input. Assignment is made by the OS when "OS_REC_ID" = "1". In this case, a value does not need to be entered at the "WREC_ID" input for inserting. The ID assigned by the OS is available at the "QWREC_ID" output for further processing.

Once a material lot has been created, you cannot change the corresponding material ID or the name of the material lot with an update.

Repetition of jobs

The "COUNT" input value defines how often a job is to be repeated if an error occurs (COUNT = 0 -> no repetition). Until the specified number of job repetitions has been reached, the job is reactivated by FB_MLOT after a wait cycle.

An error message is output after the last repetition.

Exception: The job is not repeated after inserting (WREQ_INS).

A new job is only possible if the request signal at the request inputs "RREC_REQ", "WREQ_INS", "WREC_UPD" or "WREQ_DEL" is set again.

Cyclic automatic deletion of material lots (FB_MatLotDelTrigger.pas)

Using the engineering tool, you can create a cyclic trigger for automatically deleting material lots which are no longer used. For a material lot to be deleted, the following criteria must be met when this automatic delete function is executed:

- The material lot is no longer booked in any storage location (depending on the settings, one material lot may be booked at up to 10 positions per storage location).
- The user-defined date plus the user-defined number of hours is older than the current time stamp.

A corresponding operator message is generated with the user name "PLC" for each material lot deleted.

3.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_MLOT" block:

Parameter	Description	Type	Default
SAMPLE_T	Sampling time [s] The sampling time is set by the CFC editor	REAL	1
RUNUPCYC	Number of cycles following CPU startup during which the block is not yet working and the messages are suppressed.	INT	10
ID	ID of the block The ID must be unique.	INT	0
ARCH_NA	Archive name	STRING[16]	''
ARCH_NO	Archive number	DINT	0
RREC_ID	Data record number in user archive (read jobs)	DINT	0
RREC_REQ	Trigger read job	BOOL	0
WREC_ID	Data record number in user archive (write jobs)	DINT	0
WREC_INS	Trigger insert material lot	BOOL	FALSE
WREC_DEL	Trigger delete material lot	BOOL	FALSE
WREC_UPD	Trigger change material lot	BOOL	FALSE

Parameter	Description	Type	Default
OS_REC_ID	1= REC_ID is assigned upon insertion by OS	BOOL	FALSE
COUNT	Number of repetitions if an error occurs	INT	1
RRESET	1 = Reset read interface	BOOL	0
WRESET	1 = Reset write interface	BOOL	FALSE
TIME_MON	Read and write job monitoring time	REAL	FALSE
R_REC_ST	Feedback structure from FB_UA_RM	UDT_UA_R_ST	30.0
W_REC_ST	Feedback structure from FB_UA_WM	UDT_UA_W_ST	
BA_NA	Batch name	STRING[32]	''
BA_ID	Batch ID	DWORD	0
INAME	Data to be written: Material lot name	STRING[64]	
DESCR	Data to be written: Material lot description	STRING[128]	''
UOM	Data to be written: Material lot unit	STRING[16]	''
IVERSION	Data to be written: Material lot version	REAL	''
MATID	Data to be written: Material ID	DINT	0.0
LOSSTAT / LOTSTAT	Status of material lot	DINT	0
MHD / BBD	Best before date part 1	DWORD	0
MHDB / BBDB	Best before date part 2	DWORD	0
INT1	Data to be written: Example, material lot process value 1	DINT	0
INT2	Data to be written: Example, material lot process value 2	DINT	0
MSG_EVID	ALARM_8P option: EV_ID	DWORD	0

Output parameters

The following table shows the output parameters of the "FB_MLOT" block:

Parameter	Description	Type	Default
QRUN_UP	Status display startup active = 1	BOOL	0
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0
QRELEASED	1 = Function block released for processing	BOOL	1
QGR_ERR	Function block group error/fault	BOOL	0
QERR_VER	1 = Version check error	BOOL	0
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0
QRERR	1 = Read error, Header: requested data and read data do not match	BOOL	0
QRMONERR	1 = Read job monitoring timeout	BOOL	0

Parameter	Description	Type	Default
QRLENERR	1 = Read error, Header: requested data and read data do not match	BOOL	0
QRM_ERR	FB_UA_RM: 1 = Read error, requested data and read data do not match	BOOL	0
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0
QWERR	1 = Write error, Header: requested data and written data do not match	BOOL	0
QWMONERR	1 = Write job monitoring timeout	BOOL	0
QWM_ERR	FB_UA_WM: 1 = Write error, requested data and written data do not match	BOOL	0
QWM_AERR	FB_UA_RM: 1 = Archive error, invalid data in send buffer	BOOL	0
QWM_CERR	FB_UA_RM: 1 = Communication error, error message from "BSEND" coupling blocks	BOOL	0
QWM_PERR	FB_UA_WM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0
QREQ_ERR	Several write jobs requested in same cycle	BOOL	0
QIN_PROC	Another block write job is already active	BOOL	0
QREQ_RM	Read job active	BOOL	0
QREQ_WM	Write job active	BOOL	0
QDEL_ERR	Error deleting a material lot	BOOL	0
QINS_ERR	Error inserting a material lot	BOOL	0
QUPD_ERR	Error changing a material lot	BOOL	0
QMAT_ERR	Error inserting a material lot	BOOL	0
NEW_DATA	Data have been read	BOOL	0
WR_OK	Data have been written	BOOL	0
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST	
W_REQ_ST	Request structure to FB_UA_WM	UDT_UA_W_REQ_ST	
QID	Read data: ID	INT	0
QARCH_NA	Read data: Archive name	STRING[16]	''
QARCH_NO	Read data: Archive number	DINT	0
QREC_ID	Read data: Data record number in user archive	DINT	0

Parameter	Description	Type	Default
QNO_REC	Read data: Number of rows to be read	DINT	0
QCOLUMN	Read data: Column number at which reading commences	DINT	0
QNO_COL	Read data: Number of columns to be read	DINT	0
QOFFSET	Read data: Offset, application-specific value	DINT	0
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0
QRESERVE	Read data: Reserve	DINT	0
QNAME	Read data: Material lot name	STRING[64]	''
QDESCR	Read data: Description	STRING[128]	''
QUOM	Read data: Unit	STRING[16]	''
QVERSION	Read data: Version	REAL	0.0
QCH_USER	Read data: Login name of operator who performed the last change	STRING[32]	''
QCH_DAT	Read data: Time of change part 1	DWORD	16#08123123
QCH_DATB	Read data: Time of change part 2	DWORD	16#59590000
QMATID	Read data: Material ID	DINT	0
QLOSSTAT / QLOTSTAT	Status of material lot	DINT	0
QMHD / QBBD	Best before date part 1	DWORD	0
QMHDB / QBBDB	Best before date part 2	DWORD	0
QINT1	Read data: Example, material process value 1	DINT	0
QINT2	Read data: Example, material process value 2	DINT	0
QWREC_ID	Feedback REC_ID from Write Manager	DINT	0
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST	
MSG_STAT	ALARM_8P: STATUS	WORD	0

Parameter	Description	Type	Default
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	0
QMSG_ERR	ALARM_8P: ERROR	BOOL	0

In/out parameters

The following table shows the in/out parameters of the "FB_MLOT" block:

Parameter	Description	Type	Default
APF_AS	APF function block version in AS	WORD	
APF_OS	APF function block version in OS	WORD	
GEN_AS	Generation information in AS	DWORD	
GEN_OS	Generation information in OS	DWORD	

3.1.5 Error handling

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. The error bits are available at the block as an output signal. In the event of an error, the request signal "REC_REQ" must be reset to confirm that the error information has been processed.

The following error messages can be generated:

Output structure	Output	Message text	Meaning
	QRERR	QRERR data error job ID "id" RREC_ID "rec_id" "material name"	Requested data and read data do not match
QMON_ERR	QRMONERR	QRMONERR runtime error job ID "id" RREC_ID "rec_id" "material name"	Read job monitoring timeout
	QRLENERR	QRLENERR data block length of the receive data is not correct ID "id" RREC_ID "rec_id" "material name"	Length of the read data does not match the length of the block outputs
	QWERR	Q"ERR data error job ID "id" REC_ID "rec_id"	Requested data and read data do not match
QMON_ERR	QWMONERR	QWMONERR runtime error read job ID "id" WREC_ID "rec_id" "material name"	Write job monitoring timeout
	QPARAMF	QPARAMF parameter assignment error ID "id" WREC_ID "rec_id"	Parameter assignment error, • ID=0 or • REC_ID=0 or • ARCH_NA=' ' or • TIME_MON=0 or • MATID = 0 (with WREC_INS) or • NO_COL <= 0 (with COLUMN > 0)

Output structure	Output	Message text	Meaning
	QREQ_ERR	$$BlockComment$$ QREQ_ERR, more than one trigger of function active ID "id"	More than one action (insert/update/delete) triggered in same cycle
		$$BlockComment$$ QIN_PROC, a write job is already active ID "id"	Action not possible because an action is already active (insert/update/delete)
	QERR_VER	QERR_VER Version check error	Error in the version information

3.1.6 Messaging

"ALARM_8P_1" block messages

The "FB_MLOT" block uses the "ALARM_8P_1" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QRLENERR data block length of the receive data is not correct ID @4%d@ RREC_ID @5%d@
SIG_2	AS process control Message - fault	Read job monitoring timeout	$$BlockComment$$ QRMONERR runtime error read job ID @4%d@ RREC_ID @5%d@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QRERR data error read job ID @4%d@ RREC_ID @5%d@
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0 or MATID = 0 (with WREC_INS) or NO_COL <= 0 (with COLUMN > 0)	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@
SIG_5	AS process control Message - fault	Write job monitoring timeout	$$BlockComment$$ QWMONERR runtime error write job ID @4%d@ WREC_ID @6%d@ @7%s@@8%s@
SIG_6	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QWERR data error read job ID @4%d@ WREC_ID @6%d@ @7%s@@8%s@

Message	Message class	Event	Message text
SIG_7	AS process control Message - fault	More than one action (insert/update/delete) triggered in same cycle	$$BlockComment$$ QREQ_ERR, more than one trigger of function active ID @4%d@
SIG_8	AS process control Message - fault	Action not possible because an action is already active (insert/update/delete)	$$BlockComment$$ QIN_PROC, a write job is already active ID @4%d@

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	READ REC_ID
SD_6	@6%d@	WRITE REC ID
SD_7	@7%s@	NAME part 1
SD_8	@8%s@	NAME part 2
SD_9	-	
SD_10	-	

"ALARM_8P_2" block messages

The "FB_MLOT" block uses the "ALARM_8P_2" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER Error in version check
SIG_2		None	$$BlockComment$$ A8P_2 SIG2
SIG_3		None	$$BlockComment$$ A8P_2 SIG3
SIG_4		None	$$BlockComment$$ A8P_2 SIG4
SIG_5		None	$$BlockComment$$ A8P_2 SIG5
SIG_6		None	$$BlockComment$$ A8P_2 SIG6
SIG_7		None	$$BlockComment$$ A8P_2 SIG7
SIG_8		None	$$BlockComment$$ A8P_2 SIG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	READ REC_ID
SD_6	@6%d@	WRITE REC ID
SD_7	@7%s@	NAME part 1
SD_8	@8%s@	NAME part 2
SD_9	-	
SD_10	-	

3.2 User archives

3.2.1 FB_MATERIAL_LOT

Material lots and their values for the attributes are saved in the "FB_MATERIAL_LOT" user archive.

The archive has the following structure:

Column	Description	Format
REC_ID	Internal ID of material lot	INTEGER 32 *
NAME	Name of material lot	STRING 64
DESCR	Description of material lot	STRING 128
UOM	Unit of material lot	STRING 16
VERSION	Current version	REAL
CHANGEUSER	Name of the operator who made the last change	STRING 32
CHANGEDATE	Date of the last change	DATE
MATID	Assignment to base material	-
ATTRIB_LOTSTAT ATTRIB_BBD ….. ATTRIB_<ioname>	Value of property ….. ….. Value of property	Represents the formats of the attributes from the corresponding user archive "FB_MATATTRIB_LOT"

Note

*The manual input in the editor dialog is limited to 2,000,000,000.

The engineering tool creates the first attribute as the material state.

The engineering tool creates the second attribute as the best before date.

The remaining attributes are assigned the column name "ATTRIB_" with the "FCT_BLOCK_IO_NAME" of the FB_MATATTRIB_LOT table.

Note

Forming the attribute column name (a maximum of 20 characters) from "ATTRIB_" and "<ioname>" limits the actual useful length of the IO name to 13 characters.

3.2.2 FB_MATATTRIB_LOT

Up to 100 material lot attributes (properties) are defined in the FB_MATATTRIB_LOT user archive.

The archive has the following structure:

Column	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Name of the attribute	STRING 32
UOM	Unit of measure for the attribute	STRING 16
FORMAT	Format of the attribute	INTEGER 32
ENUM_MAT	Name of enumeration / coding of material (name of the PCS7 enumeration when FORMAT=40)	STRING 16
HL	Low limit of operator control for numerical values	DOUBLE
LL	High limit of operator control for numerical values	DOUBLE
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the output in the corresponding function block	STRING 20
VISIBLE	Display of the attribute in the process picture	INTEGER 32
CLASS_VISIBLE_1 …. CLASS_VISIBLE_4	Not currently used	INTEGER 32

Format coding

The following table shows the coding of the attribute in "FORMAT":

Integer value	Format
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places
5	REAL with 4 decimal places
-	-
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
-	-
20xx	DATE
21xx	TIME
22xx	DATE AND TIME
-	-
30	BOOL
40	ENUMERATION

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
-	-
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
-	-
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
-	-
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

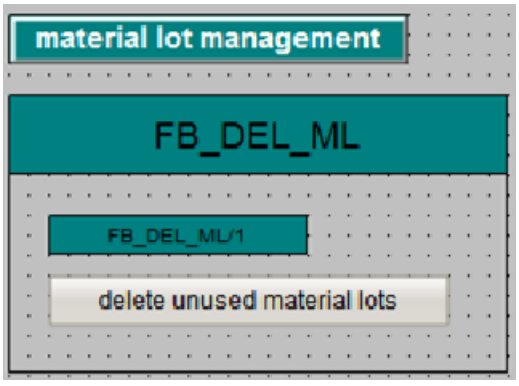
3.3 Operator control and monitoring

3.3.1 Block icons

No block icon is assigned to the "FB_MLOT" block. This block is always controlled indirectly from other blocks.

@Template_APF.pdl

The "@Template_APF.pdl" template picture contains a template for configuring the deletion of material lots which are not in use:



3.3.2 Picture objects

The interface for material lot management operator control and monitoring consists of the following picture objects:

- FB_MAT_LOT_ED.PDL
  The picture allows you to create, change, delete and copy material lot master data.
- @FB_MAT_LOT_ED_WINDOW.PDL
  You can use the picture to edit the attribute parameters of the material lots. This picture is an integrated window in the "FB_MAT_LOT_ED.pdl" picture and cannot be used separately.

In addition to the editor picture objects you can use the faceplate:

- @PG_FB_DEL_ML.pdl
  You can delete unused material lots (not assigned to a storage location) with the faceplate.

Note

The process pictures are pre-configured and do not need to be edited.

For detailed information on interface operator control and monitoring in process mode, see the APF Operator Manual.

3.3.3 Messages

The following messages can be output:

Message class	Event	Message text
Operator message	Save button	@7%s@: @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ material lot: @6%s@
Operator message	Save button	@7%s@: @2%s@: new = @5%s@ @8%s@ old = @4%s@ @8%s@ material lot: @6%s@
Operator message	Save button	@7%s@: material lot @6%s@ changed
Operator message	Save button	@7%s@: material lot @6%s@ created
Operator message	Delete button	@7%s@: material lot @6%s@ deleted
Operator message	New material lot successfully generated with FB_MLOT	@7%s@: material lot @6%s@ created (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
Operator message	Existing material lot successfully deleted with FB_MLOT	@7%s@: material lot @6%s@ deleted (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
Operator message	Existing material lot successfully changed with FB_MLOT	@7%s@: material lot @6%s@ changed (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
OS process control message - fault	Attempt to generate existing material lot again with FB_MLOT	@7%s@: Material lot @6%s@ already exists - cannot be created again! (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
OS process control message - fault	Attempt to delete a non-existent material lot with FB_MLOT	@7%s@: Material lot @6%s@ does not exist - cannot be deleted! (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
OS process control message - fault	Attempt to modify a non-existent material lot with FB_MLOT	@7%s@: Material lot @6%s@ does not exist - cannot be changed! (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
OS process control message - fault	Attempt to create a material lot with a non-existent material ID with FB_MLOT	@7%s@: Material does not exist - material lot cannot be created! (FB_UA_WM.ID: @4%d@ FB_MLOT.ID: @5%d@)
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
OS process control error	Cyclic backup action: The export file could not be created.	@7%s@: Backup file (@2%s@) could not be exported
OS process control error	Cyclic backup action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the backup file
OS process control error	Cyclic backup action: Directory 1 could not be created	@7%s@: Backup directory 1 does not exist / could not be created
OS process control error	Cyclic backup action: Directory 2 could not be created	@7%s@: Backup directory 2 does not exist / could not be created
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 1 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 2 (@2%s@) could not be copied

Message class	Event	Message text
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 1 is corrupt
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 2 is corrupt
OS process control error	Cyclic backup action: Call the external backup program failed	@7%s@: Initialization of the backup program failed

Message associated values

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%s@	Batch name
Message associated value 2	@2%s@	Material lot attribute / backup file
Message associated value 3	@3%d@	Batch ID OR partial record ID
Message associated value 4	@4%s/d@	Old value (string or number) OR ID of FB_UA_WM block
Message associated value 5	@5%s/d@	New value (string or number) OR ID of FB_MLOT block
Message associated value 6	@6%s@	Material lot name + ID, here: Material lot name (ID)
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	Unit / UOM
Message associated value 9	@9%s@	Not used
Message associated value 10	@10%s@	Source, here: Material lot management

For additional information, refer to the PCS 7 online help.

Storage location management

4

4.1 FB_STOR (FB_STORI, FB_STORR)

4.1.1 Description

Object properties

The following table shows the object properties of FB_STOR:

Name	Number	Family	Module
FB_STOR	FB1781	APF	Storage location management

Area of application

The "FB_STOR" block provides the storage location parameters (attributes) available in the user archive in the AS and transfers the actual values in the AS to the user archive.

On the basis of the "FB_STOR" block, the APF Engineering Tool generates the "FB_STORI" and "FB_STORR" blocks for displaying format-related numerical values.

Startup characteristics

The "FB_STOR" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures. Data exchange with the "FB_UA_WM" communication block is performed via the "UDT_UA_W_REQ_ST" (request) and "UDT_UA_W_ST" (feedback) structures.

A read or write request can be sent to the block via the "FB_STOR_ED.pdl" process picture. The process picture writes the "OP_REC_R" input parameters when a storage location is saved

and "OP_REC_W" when a storage location is updated. The actual value content (CONT_A input) is also available as a WinCC tag.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

4.1.2 Generation

Using the APF Engineering Tool, the following blocks are generated on the basis of the "FB_STOR" (FB1781) block:

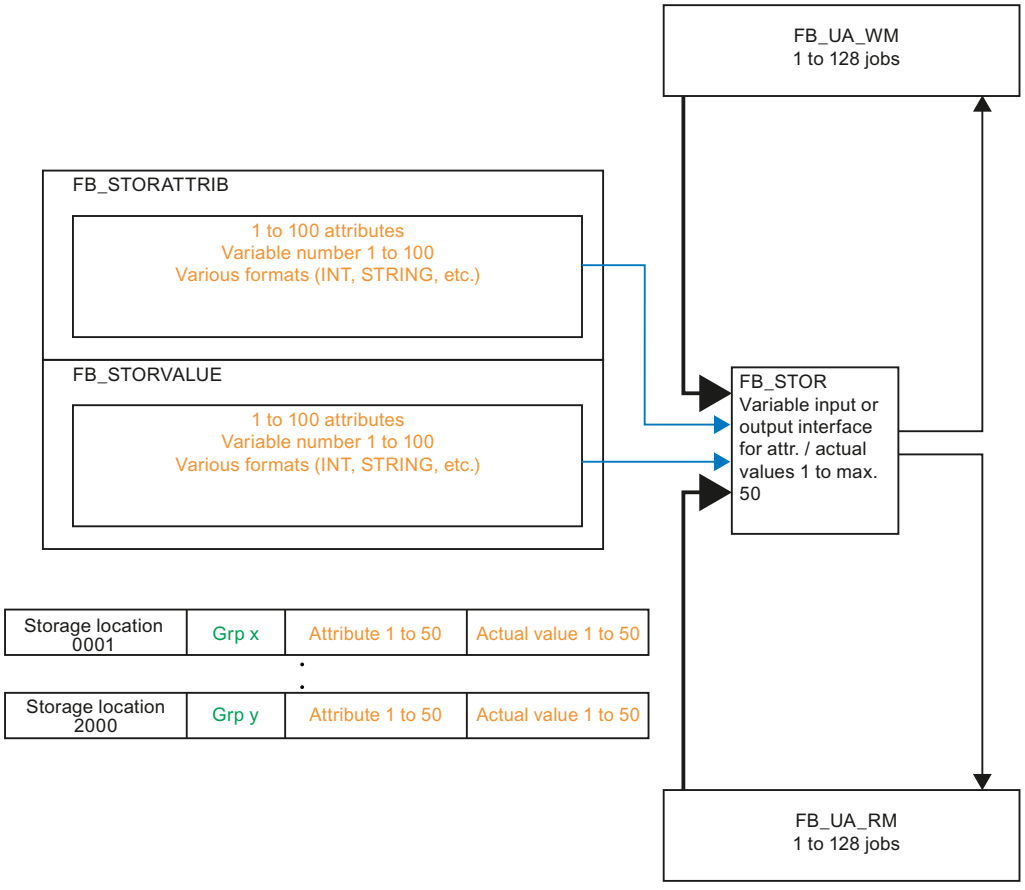
Function block (FB)	Number	Function
FB_STORI	Specified by the "FB Number (I)" field of the APF engineering tool	Provides the storage location parameters (attributes) available in the "FB_STORATTRIB" user archive in the AS and transfers the actual values in the AS to the "FB_STORVALUE" user archive Format of format-related numerical values: INTEGER (32)
FB_STORR	Specified by the "FB Number (R)" field of the APF engineering tool	Provides the storage location parameters (attributes) available in the "FB_STORATTRIB" user archive in the AS and transfers the actual values in the AS to the "FB_STORVALUE" user archive Format of format-related numerical values: REAL

The "FB_STORI" and "FB_STORR" blocks differ in their display of format-related numerical values for FORMAT= 53, 54 and 70. These values are displayed in the blocks as INTEGER (32-bit) in FB_STORI and as REAL in FB_STORR. The two blocks write to the same archive (as DOUBLE 64-bit).

The input and output parameters of the function blocks depend on the structure of the "FB_STORATTRIB" and "FB_STORVALUE" user archives.

One FB_STORI or FB_STORR is to be used for each storage location in the CFC.

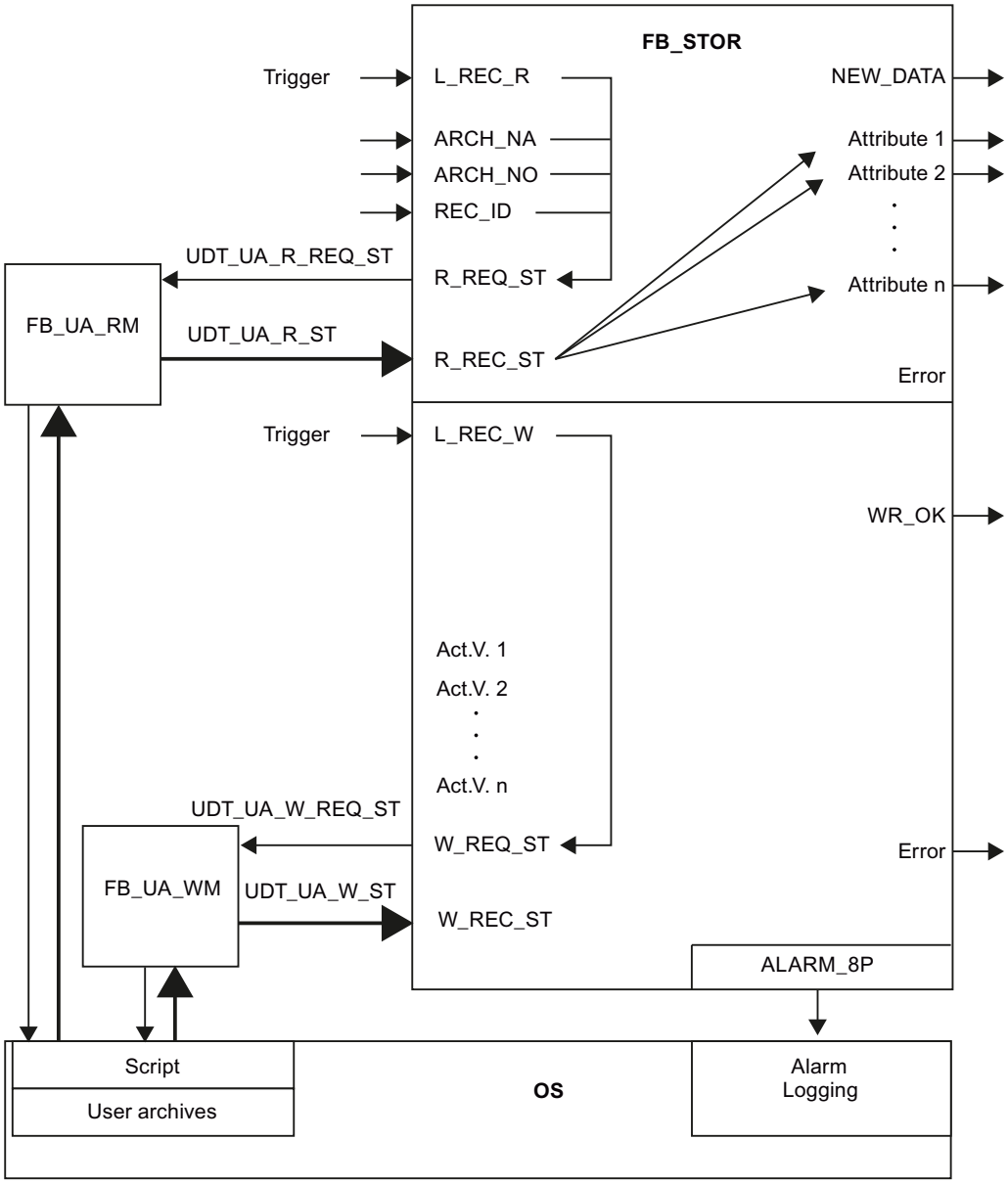
The figure below provides an overview of the "FB_STOR" block structure:


FB_UA_WM
1 to 128 jobs
FB_STORATTRIB
1 to 100 attributes
Variable number 1 to 100
Various formats (INT, STRING, etc.)
FB_STORVALUE
1 to 100 attributes
Variable number 1 to 100
Various formats (INT, STRING, etc.)
FB_STOR
Variable input or output interface for attr. / actual values 1 to max. 50
Storage location 0001
Grp x
Attribute 1 to 50
Actual value 1 to 50
Storage location 2000
Grp y
Attribute 1 to 50
Actual value 1 to 50
FB_UA_RM
1 to 128 jobs

4.1.3 Functions

Overview

The figure below provides an overview of how the "FB_STOR" block works:



Reading data

Following a request by a positive edge at request input "L_REC_R", the request data (REC_ID (storage location ID), ARCH_NA /ARCH_NO, etc.) are written to the output structure "UDT_UA_R_REQ_ST". The read job is activated in the structure by setting the "REC_REQ" bit.

Once the read job has been executed, the communication block "FB_UA_RM" writes the request data (including REQ_ID) and the pointer to the storage location data record to the "UDT_UA_R_ST" input structure of the block, where it also sets the "NEW_DATA" input bit.

Note

For detailed information on the "UDT_UA_R_REQ_ST" and "UDT_UA_R_ST" structures, see section Data structures and data formats (Page 239).

The input data are only assigned to the block's read job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" in the input structure.

Once the input structure data have been assigned to the read job, the data are checked for validity and plausibility. The system evaluates the following:

- Whether the request data in the input and output structure match
- Whether the data length calculated corresponds to the data length of the outputs at the block

If the data are plausible and valid, the "NEW_DATA" output is set and the data are written from the "FB_UA_RM" communication block to the outputs of the block, where they are available for further processing.

The "NEW_DATA" output is reset by resetting the "REC_REQ" request input.

The "FB_UA_RM" communication block reports any errors it detects during execution of the read job to the "FB_STOR" block. The "FB_STOR" block deactivates the read job if an error occurs ("REC_REQ" = 0). This also happens if the "FB_STOR" block itself detects an error.

A positive edge at the "RRESET" input resets the material data at the block output, the "NEW_DATA" output and the error messages.

Writing data (updating actual values)

Following a request by a positive edge at request input "L_REC_W", the request data (REC_ID (storage location ID), ARCH_NA /ARCH_NO, etc.) and the actual value input data are written to the output structure "UDT_UA_W_REQ_ST". The write job is activated in the structure by setting the "REC_REQ" bit.

Note

For detailed information on the "UDT_UA_W_REQ_ST" and "UDT_UA_W_ST" structures, see section Data structures and data formats (Page 239).

Once the write job has been processed, the "FB_UA_WM" block writes the confirmation data (including REQ_ID) to the "UDT_UA_W_ST" input structure of the block and sets the "QWR_OK" bit.

The input data are only assigned to the block's write job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" of the input structure.

Once the input structure data have been assigned to the write job and the "QWR_OK" bit has been set, the data are checked for validity and plausibility. The system checks whether the request data in the input and output structure match.

The "WR_OK" output of the "FB_STOR" block is only set if the data are plausible and valid.

The "WR_OK" output is reset by resetting the "L_REC_W" request input.

The "FB_UA_WM" communication block reports any errors it detects during execution of the write job to the "FB_STOR" block. The "FB_STOR" block deactivates the write job if an error occurs ("REC_REQ" = 0). This also happens if the "FB_STOR" block itself detects an error.

A positive edge at the "WRESET" input resets the "WR_OK" output and the error messages relating to the write job.

Automatic transfer of values from the user archive

The "FB_STOR" block can automatically trigger reading of storage location data from the user archive upon a CPU restart. For this function to operate, bit 15 in the "TRIGMODE" input word must be set.

If an error occurs during reading, the block launches another attempt to read the storage location data after the time set at the "INIREP_T" input. This process is not interrupted and is always performed again whenever an error occurs.

Automatic transfer of actual values to the user archive

Because the DB number of the "FB_STOR" block must always be current in the corresponding data record of the archive, upon initial insertion the block triggers a write request to transfer its own DB number to the user archive.

If the DB number changes during runtime, for example, following full compilation of the AS, the block registers this and again automatically triggers a write job.

If an error occurs during writing, the block launches another attempt to write its DB number to the user archive after the time set at the "INIREP_T" input. This process is not interrupted and is always performed again whenever an error occurs.

To update actual values in the user archive automatically after certain process value changes, you can define specific changes at the "TRIGMODE" input after which the block is to automatically write its process values to the user archive. In accordance with the bits set, the block monitors the specified inputs and triggers a write job to the user archive if one of these values is modified.

Bit	Monitored input	Meaning
0	Not used	-
1	Not used	-
2	STOR_STATUS	Storage location status
3	MAT_STATUS	Material status
4	REL1STAT	Static procedure enable 1
5	REL2STAT	Static procedure enable 2
6	REL3STAT	Static procedure enable 3
7	REL4STAT	Static procedure enable 4
8	REL1DYN	Dynamic procedure enable 1
9	REL2DYN	Dynamic procedure enable 2
10	REL3DYN	Dynamic procedure enable 3

Bit	Monitored input	Meaning
11	REL4DYN	Dynamic procedure enable 4
12	Not used	-
13	Not used	-
14	Not used	-
15	*None (restart)*	Read data upon CPU restart

Automatic repetition of jobs

The "COUNT" input value defines how often a job is to be repeated if an error occurs (COUNT = 0 -> no repetition). Until the specified number of job repetitions has been reached, the job is reactivated by FB_STOR after a wait cycle.

An error message is output after the last repetition.

A new job is only possible if the request signal at the request inputs "L_REC_R" and "L_REC_W" is set again.

4.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_STOR" block:

Parameter	Description	Type	Default	WinCC tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
ID	ID of the block	INT	0	-
ARCH_NA	Archive name	STRING[16]	''	-
REC_ID	Data record number in user archive	DINT	0	-
FORMAT	Data format content 0 = DINT, >1 = REAL	DINT		-
BA_NA	Batch name	STRING[32]	''	-
BA_ID	Batch ID	DWORD	0	-
COUNT	Number of repetitions of the send or read job if an error occurs	INT	1	-
INIREP_T	Delay time for repetition of initialization if an error occurs	REAL	600.0	-
TIME_MON	Read/write job monitoring time	REAL	30.0	-
TRIGMODE	Trigger mode for automatically sending current process values	WORD	0	-
R_REC_ST	Input structure with job feedback from FB_UA_RM	UDT_UA_R_ST		-
W_REC_ST	Input structure with job feedback from FB_UA_WM	UDT_UA_W_ST		-
L_REC_R	Interconnectable input: 1 = Read data request	BOOL	0	-

Parameter	Description	Type	De-fault	WinCC tag
L_REC_W	Interconnectable input: 1 = Write data request	BOOL	0	-
RRESET	Reset data from read job	BOOL	0	-
WRESET	Reset data from write job	BOOL	0	-
DBNO	Number of the instance data block	DINT	0	-
CONT_A	Current process value 1	DINT	0	X
CONT_P	Current process value 2	REAL	0.0	-
STO_STAT	Current process value 3	DINT	0	-
MAT_STAT	Current process value 4	DINT	0	-
REL1STAT	Current process value 5	DINT	0	-
REL2STAT	Current process value 6	DINT	0	-
REL3STAT	Current process value 7	DINT	0	-
REL4STAT	Current process value 8	DINT	0	-
REL1DYN	Current process value 9	DINT	0	-
REL2DYN	Current process value 10	DINT	0	-
REL3DYN	Current process value 11	DINT	0	-
REL4DYN	Current process value 12	DINT	0	-
VALUE_13	Current process value 13	DINT	0	-
VALUE_14	Current process value 14	DINT	0	-
MSG_EVID	ALARM_8P: EV_ID	DWORD	0	-
"x": WinCC tag available; "-": WinCC tag not available				

Output parameters

The following table shows the output parameters of the "FB_STOR" block:

Parameter	Description	Type	Default	WinCC tag
QREC_R	Status of read data request	INT	0	X
QREC_W	Status of write data request	INT	0	X
QRUN_UP	Status display startup active = 1	BOOL	0	-
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	X
QGR_ERR	Function block group error/fault	BOOL	0	-
QERR_VER	1 = Version check error	BOOL	0	
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0	-
QRERR	1 = Read error, Header: requested data and read data do not match	BOOL	0	-
QRMONERR	1 = Read job monitoring timeout	BOOL	0	-
QRLENERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0	-
QRM_ERR	FB_UA_RM: 1 = Read error, Requested data and read data do not match	BOOL	0	-

Parameter	Description	Type	Default	WinCC tag
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0	-
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0	-
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QWERR	1 = Write error, data header: requested data and read data do not match	BOOL	0	-
QWMONERR	1 = Write job monitoring timeout	BOOL	0	-
QWM_ERR	FB_UA_WM: 1 = Write error, requested data and written data do not match	BOOL	0	-
QWM_AERR	FB_UA_WM: 1 = Archive error, invalid data in receive buffer	BOOL	0	-
QWM_CERR	FB_UA_WM: 1 = Communication error, error message from "BSEND" coupling blocks	BOOL	0	-
QWM_PERR	FB_UA_WM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QREQ_RM	1 = Read job active	BOOL	0	-
QREQ_WM	1 = Write job active	BOOL	0	-
NEW_DATA	1= data have been read	BOOL	0	-
WR_OK	1= data have been written	BOOL	0	-
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST		-
W_REQ_ST	Request structure to FB_UA_WM	UDT_UA_W_REQ_ST		-
QID	Read data: ID	INT	0	-
QARCH_NA	Read data: Archive name	STRING[16]	''	-
QARCH_NO	Read data: Archive number	DINT	0	-
QREC_ID	Read data: Data record number in user archive	DINT	0	-
QNO_REC	Read data: Number of rows to be read	DINT	0	-
QCOLUMN	Read data: Column number at which reading commences	DINT	0	-
QNO_COL	Read data: Number of columns to be read	DINT	0	-
QOFFSET	Read data: Offset, application-specific value	DINT	0	-
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0	-
QRESERVE	Read data: Reserve	DINT	0	-
QNAME	Read data: Name	STRING[64]	''	-
QDESCR	Read data: Description	STRING[128]	''	-
QLIC	Read data: Location identifier	STRING[16]	''	-
QSTORGR	Read data: Storage group	DINT	0	-
QVERSION	Read data: Version	REAL	0.0	-
QCH_USER	Read data: Login name of the operator who made the last change	STRING[32]	''	-

Parameter	Description	Type	Default	WinCC tag
QCH_DAT	Read data: Time of change part 1	DWORD	16#08123123	-
QCH_DATB	Read data: Time of change part 2	DWORD	16#59590000	-
QFORMAT	Read data: Data format content	DINT	-1	-
QACCMODE	Read data: Booking mode	DINT	0	-
QMATID	Read data: Status value 1	DINT	0	-
QMLOTID1	Read data: Status value 2	DINT	0	-
QMLOT_CONT1	Read data: Status value 3	DINT	0	-
QMLOTID2	Read data: Status value 4	DINT	0	-
QMLOT_CONT2	Read data: Status value 5	DINT	0	-
QMLOTID3	Read data: Status value 6	DINT	0	-
QMLOT_CONT3	Read data: Status value 7	DINT	0	-
QCONT_MAX	Read data: Status value 8	DINT	0	-
QCONT_MIN	Read data: Status value 9	DINT	0	-
QSTO_CON	Read data: Status value 10	DINT	0	-
QSTO_USE_IN	Read data: Status value 11	DINT	0	-
QSTO_USE_OUT	Read data: Status value 12	DINT	0	-
QUOM	Read data: Status value 13	STRING[16]	''	-
QMACLRE1	Read data: Status value 14	DINT	0	-
QMACLRE2	Read data: Status value 15	DINT	0	-
QMACLRE3	Read data: Status value 16	DINT	0	-
QMACLRE4	Read data: Status value 17	DINT	0	-
QMATYRE1	Read data: Status value 18	DINT	0	-
QMATYRE2	Read data: Status value 19	DINT	0	-
QMATYRE3	Read data: Status value 20	DINT	0	-
QMATYRE4	Read data: Status value 21	DINT	0	-
QATTR16	Read data: Status value 22	DINT	0	-
QATTR17	Read data: Status value 23	DINT	0	-
QATTR18	Read data: Status value 24	DINT	0	-
ALARM_ST	ALARM structure for the connection of an external message block	UDT_ALARM_ST		-
MSG_STAT	ALARM_8P: STATUS	WORD	0	-
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0	-
QMSG_SUP	ALARM_8P: 1 = Message suppression active	BOOL	0	-
QMSG_ERR	ALARM_8P: ERROR	BOOL	0	-
"x": WinCC tag available; "-": WinCC tag not available				

Input and output parameters

The following table shows the input and output parameters of the "FB_STOR" block:

Parameter	Description	Type	Default	WinCC tag
OP_REC_R	Operator: 1 = Read data	BOOL	0	X
OP_REC_W	Operator: 1 = Write data	BOOL	0	X

Parameter	Description	Type	Default	WinCC tag
APF_AS	APF function block version in AS	WORD	0	X
APF_OS	APF function block version in OS	WORD	0	X
GEN_AS	Generation information in AS	DWORD	0	X
GEN_OS	Generation information in OS	DWORD	0	X
"x": WinCC tag available;				

4.1.5 Error handling

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. The error bits are available at the block as an output signal. In the event of an error, the request signal (e.g., "L_REC_R") must be reset to confirm that the error information has been processed.

The following error messages can be generated:

Output structure	Output	Message text	Meaning
	QRLENERR	Length of the read data does not match the length of the block outputs	QRLENERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
MON_ERR	QRMONERR	Read job monitoring timeout	QRMONERR runtime error read job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
	QRERR	Requested data and read data do not match	QRERR data error read job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
	QPARAMF	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	QPARAMF parameter assignment error ID @4%d@
MON_ERR	QWMONERR	Write job monitoring timeout	QWMONERR runtime error send job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
	QWERR	Requested data and written data do not match	QWERR data error send job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
	QRM_ERR	General FB_UA_RM error	
	QRM_AERR	FB_UA_RM archive error	
	QRM_CERR	FB_UA_RM communication error	
	QRM_PERR	FB_UA_RM parameter assignment error	
	QWM_ERR	General FB_UA_WM error	
	QWM_AERR	FB_UA_WM archive error	
	QWM_CERR	FB_UA_WM communication error	

Output structure	Output	Message text	Meaning
	QWM_PERR	Parameter assignment error of the FB_UA_WM	
	QERR_VER	QERR_VER Version check error	Error in the version information

4.1.6 Messaging

The "FB_STOR" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The version error will additionally be reported through the ALARM_8P (SIG_7).

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QRLEN_ERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
SIG_2	AS process control Message - fault	Read job monitoring timeout	$$BlockComment$$ QRMON_ERR runtime error read job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QRERR data error read job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@
SIG_5	AS process control Message - fault	Write job monitoring timeout	$$BlockComment$$ QWMON_ERR runtime error send job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
SIG_6	AS process control Message - fault	requested data and written data do not match	$$BlockComment$$ QWERR data error send job ID @4%d@ REC_ID @5%d@ @6%s@@7%s@
SIG_7	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER error in version check
SIG_8	No message	-	-

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	@6%s@	NAME part 1
SD_7	@7%s@	NAME part 2
SD_8	-	
SD_9	-	
SD_10	-	

4.2 FB_SACT

4.2.1 Description

Object properties

The following table shows the object properties of FB_SACT:

Name	Number	Family	Module
FB_SACT	FB1783	APF	Storage location management

Area of application

The "FB_SACT" (storage process values) block performs rapid updates of up to 10 actual values for up to 25 "FB_STOR" blocks. It is adapted to the "FB_STOR.pdl" process picture and can update all 25 storage locations displayed in the picture, along with their parameters, in one single operation.

The block has the following functions:

- Collecting actual values of storage locations (max. of 25) requested by the OS
- Processing data and transferring data to the OS in a job with the "FB_UA_WM" block
- Monitoring data communication (generation of alarms)

Startup characteristics

The "FB_SACT" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

Data exchange with the "FB_UA_WM" communication block on the AS level is performed via the "UDT_UA_W_REQ_S" (request) and "UDT_UA_W_ST" (feedback) structures.

Operator control and monitoring tags are used to coordinate the data communication between the AS and OS.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

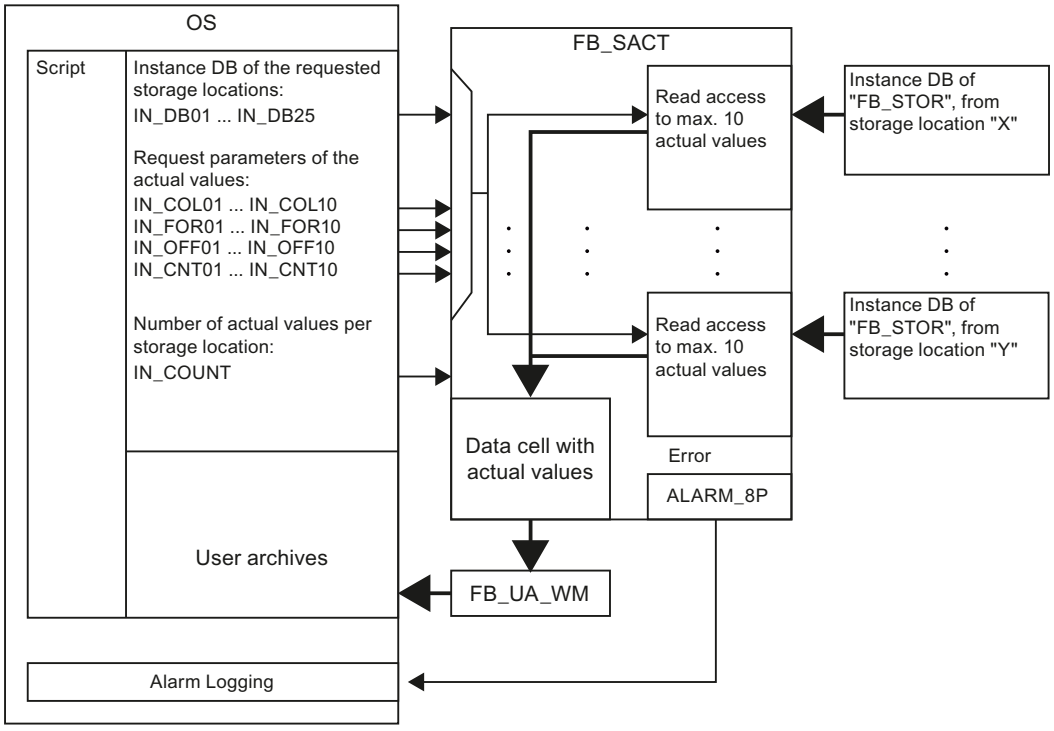
4.2.2 Generation

The "FB_SACT" block is not generated by the APF Engineering Tool. It is part of the library supplied with APF.

4.2.3 Functions

Overview

The figure below provides an overview of the "FB_SACT" block function:



How it works

At a request from the OS, the "FB_SACT" block reads the current actual values of the storage locations from the AS and sends these as a data cell to the requesting OS using the "FB_UA_WM" block.

At the input parameters "IN_DB01" to "IN_DB25", the OS specifies the "FB_STOR" instance data blocks of the storage locations from which actual values are to be read. If the actual values of fewer than 25 storage locations are requested, the OS writes "0" to the "IN_DB…" inputs that are not required.

The OS defines the actual values to be read with specifications at the input parameters "IN_FOR01" to "IN_FOR10" and "IN_OFF01" to "IN_OFF10". The "IN_OFF..." parameter specifies the position of the required actual value in the status area of the instance DB of the corresponding "FB_STOR" block. The "IN_FOR.." parameter is used to calculate the volume of data to be read.

You can also query the status of individual binary values in a data word. The "IN_OFF..." parameter specifies the position of the data word in the instance DB, and the "IN_CNT..." parameter (binary value position) defines the bit to be queried (= 0 to 15).

The number of actual values to be read is specified by the OS with the "IN_COUNT" parameter. Up to 10 actual values can be queried from each of a maximum of 25 storage locations.

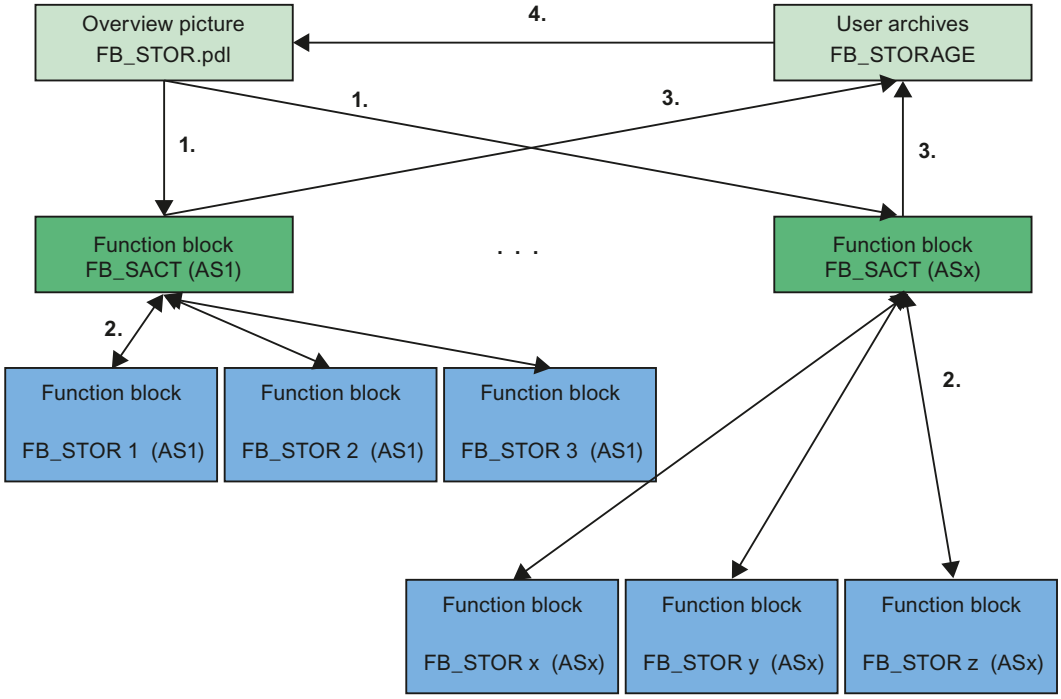
When the function is triggered by the OS and "OP_REC_W" = 1, the actual values are read from the instance data blocks of the "FB_STOR" block and entered in full in a data cell. The number of "IN_COUNT" actual values and the "IN_COL01" / "IN_FOR01" to "IN_COL10" / "IN_FOR10" parameters are written in the header of the data cell. The "FB_SACT" block sends the data cell to the OS with the "FB_UA_WM" block. If data transfer is completed without errors, the "FB_SACT" block sets the output "WR_OK".

The request in progress is aborted if the OS "OP_REC_W" request is canceled (OP_REC_W = 0).

One "FB_SACT" block must be configured in the CFC for each AS. The individual "FB_SACT" blocks to be triggered are identified by the AS ID in the user archive ("ASNO" in the FB_STORAGE archive).

The AS ID must be entered in the "FB_STOR_ED.pdl" process picture. This ID must correspond to the ID specified in the APF Engineering Tool as it is assigned to a tag name for the corresponding "FB_SACT" block.

The following schematic diagram shows how the "FB_SACT" block works:



Data cell

The maximum size of the data cell for transferring the data is 16230 bytes. The index aa, bb, etc. can have the value 01, 02, 03 up to a maximum of 25 (IN_DB with input value 0 is skipped), and index y the value 1 to a maximum of "IN_COUNT", in ascending order.

CELL	FORMAT	Bytes	Address
IN_COUNT	INT	2	0
IN_COL1	INT	2	2
IN_FOR1	INT	2	4

CELL	FORMAT	Bytes	Address
IN_COL2	INT	2	6
IN_FOR2	INT	2	8
IN_COL3	INT	2	10
IN_FOR3	INT	2	12
IN_COL4	INT	2	14
IN_FOR4	INT	2	16
IN_COL5	INT	2	18
IN_FOR5	INT	2	20
IN_COL6	INT	2	22
IN_FOR6	INT	2	24
IN_COL7	INT	2	26
IN_FOR7	INT	2	28
IN_COL8	INT	2	30
IN_FOR8	INT	2	32
IN_COL9	INT	2	34
IN_FOR9	INT	2	36
IN_COL10	INT	2	38
IN_FOR10	INT	2	40
IN_DBaa, REC_ID	DINT	4	42
IN_DBaa, FORMAT	DINT	4	46
IN_DBaa, requested value 1	Requested format	Depends on format	50
IN_DBaa, requested value 2	Requested format	Depends on format	…
…	...	…	...
IN_DBaa, requested value y	Requested format	Depends on format	…
IN_DBbb, REC_ID	DINT	4	…
IN_DBbb, FORMAT	DINT	4	…
IN_DBbb, requested value 1	Requested format	Depends on format	…
IN_DBbb, requested value 2	Requested format	Depends on format	…
…	...	…	...
IN_DBbb, requested value y	Requested format	Depends on format	…
IN_DB…, REC_ID	DINT	4	…
IN_DB…, FORMAT	DINT	4	…
IN_DB…, requested value 1	Requested format	Depends on format	…
IN_DB…, requested value 2	Requested format	Depends on format	…

CELL	FORMAT	Bytes	Address
…	...	…	...
IN_DB…, requested value y	Requested format	Depends on format	… Maximum of 16229

4.2.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_SACT" block:

Parameter	Description	Type	Default
SAMPLE_T	Sampling time [s]	REAL	1
RUNUPCYC	Number of startup cycles	INT	10
ID	ID of the block	INT	0
ARCH_NA	Archive name	STRING[16]	''
IN_DB01 to IN_DB25	Number of FB_STOR instance data block for storage locations 1 to 25	DINT	0
IN_COL01 to IN_COL10	Column in user archive for values 1 to 10	DINT	0
IN_FOR01 to IN_FOR10	Format ID for values 1 to 10	DINT	0
IN_OFF01 to IN_OFF10	Offset for values 1 to 10	DINT	0
IN_CNT01 to IN_CNT10	Position of binary values 1 to 10	WORD	0
IN_COUNT	Number of values to be transferred per storage location	DINT	0
COUNT	Number of send trigger repetitions if an error occurs	INT	1
TIME_MON	Monitoring time	REAL	30.0
W_REC_ST	Input structure with job feedback from FB_UA_WM	UDT_UA_W_ST	
RESET	Reset data	BOOL	0
MSG_EVID	ALARM_8P: EV_ID	DWORD	0

Output parameters

The following table shows the output parameters of the "FB_SACT" block:

Parameter	Description	Type	Default
QREC_W	Status request data write	INT	0
QINDBERR	1 = Error, no instance DB or incorrect length	BOOL	0
QRUN_UP	Status display startup active = 1	BOOL	0
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0
QRELEASED	1= Function block released for processing	BOOL	1
QGR_ERR	Function block group error/fault	BOOL	0
QERR_VER	1 = Version check error	BOOL	0
QPARAMF	1 = parameter assignment error	BOOL	0
QLEN_ERR	1 = Error, data exceeds data cell size	BOOL	0
QCNT_ERR	1 = Error, input "IN_COUNT" > 10 or <= 0	BOOL	0
QBOOLERR	1 = Error, bit selection "IN_CNT…" > 15 or < 0	BOOL	0
QERR	1 = Write job error	BOOL	0
QMONERR	1 = Write job monitoring timeout	BOOL	0
QWM_ERR	FB_UA_WM: 1 = Write error Job data in WinCC communication interface do not correspond to original job data.	BOOL	0
QWM_AERR	FB_UA_WM: 1 = Archive error Job data in WinCC communication interface reset by the OS or script error in the OS	BOOL	0
QWM_CERR	FB_UA_WM: 1 = Communication error Error message from "BSEND" coupling blocks	BOOL	0
QWM_PERR	FB_UA_WM: 1 = parameter assignment error duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0
QREQ_WM	1 = Write job at FB_UA_WM active	BOOL	0
WR_OK	1 = Data request/write job completed without errors	BOOL	0
W_REQ_ST	Output structure with send trigger for FB_UA_WM	UDT_UA_W_REQ_ST	
ALARM_ST	Interface with an external ALARM_8P	UDT_ALARM_ST	
MSG_STAT	ALARM_8P: STATUS	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	0
QMSG_ERR	ALARM_8P: ERROR	BOOL	0
OP_REC_W	Operator: 1 = Data request	BOOL	0

In/out parameters

The following table shows the input and output parameters of the "FB_SACT" block:

Parameter	Description	Type	Default
OP_REC_W	Operator: 1 = Data request	BOOL	0
APF_AS	APF function block version in AS	WORD	0
APF_OS	APF function block version in OS	WORD	0

4.2.5 Error handling

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals.

The following error messages can be generated:

Request structure	Output	Meaning
	QCNT_ERR	Error, invalid IN_COUNT request parameter, IN_COUNT <= 0 or > 10
	QBOOLERR	Error, invalid "IN_CNTxx" request parameter, IN_CNT… < 0 or > 15
	QINDBERR	Instance DB error; no instance data block or insufficient data length
	QLEN_ERR	Error Maximum data cell size exceeded, data cell > 16230 bytes
	QPARAMF	Parameter assignment error; ID or TIME_MON <= 0 or ARCH_NA not specified
	QERR	Data error REQ_ID in request and receive structure not the same
MON_ERR	QMONERR	Runtime error; monitoring timeout
	QWM_ERR	Error from FB_UA_W: Data error, job data in WinCC communication interface do not correspond to original job data
	QWM_AERR	Error from FB_UA_W: Archive error; job data in the WinCC communication interface reset by the OS, or script error in the OS
	QWM_CERR	Error from FB_UA_W: Communication error; Error message from "BSEND" coupling blocks
	QWM_PERR	Error from FB_UA_W: Parameter assignment error; incorrect ID_1, R_ID_1 or ID_2, R_ID_2 or duplicate REQ_ID

4.2.6 Messaging

The "FB_SACT" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - error	IN_COUNT <= 0 or > 10	$$BlockComment$$ QCNT_ERR parameter error IN_COUNT <=0 or >10 D @4%d@, IN_COUNT @6%d@
SIG_2	AS process control Message - fault	Communication error, error message from "BSEND" coupling blocks	$$BlockComment$$ QMONERR monitoring timeout D @4%d@, IN_COUNT @6%d@
SIG_3	AS process control Message - fault	REQ_ID in request and receive structure not the same	$$BlockComment$$ QERR data error ID @4%d@, IN_COUNT @6%d@
SIG_4	AS process control Message - error	ID or TIME_MON <= 0 or ARCH_NA not specified	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@
SIG_5	AS process control Message - error	No instance data block or insufficient data length	$$BlockComment$$ QINDBERR parameter error instance DB @5%d@ ID @4%d@
SIG_6	AS process control Message - fault	Maximum data cell size exceeded, data cell > 16230 bytes	$$BlockComment$$ QLEN_ERR length error writing data record ID @4%d@
SIG_7	AS process control Message - error	IN_CNT… < 0 or > 15	$$BlockComment$$ QBOOLERR parameter error IN_CNT <0 or >15 ID @4%d@, instance DB @5%d@, IN_COUNT @6%d@
SIG_8	No message	-	-

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA

The following associated values are used:

Associated value	Format	Value
SD_1	-	
SD_2	-	
SD_3	-	
SD_4	@4%d@	ID
SD_5	@5%d@	IN_DB number
SD_6	@6%d@	IN_COUNT
SD_7	-	
SD_8	-	
SD_9	-	
SD_10	-	

4.3 FB_ST_AC

4.3.1 Description

Object properties

The following table shows the object properties of FB_ST_AC:

Name	Number	Family	Module
FB_ST_AC	FB1782	APF	Storage location management

Area of application

The "FB_ST_AC" block (storage account) handles the booking and clearing of materials/ material lots from the AS.

Startup characteristics

The "FB_ST_AC" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

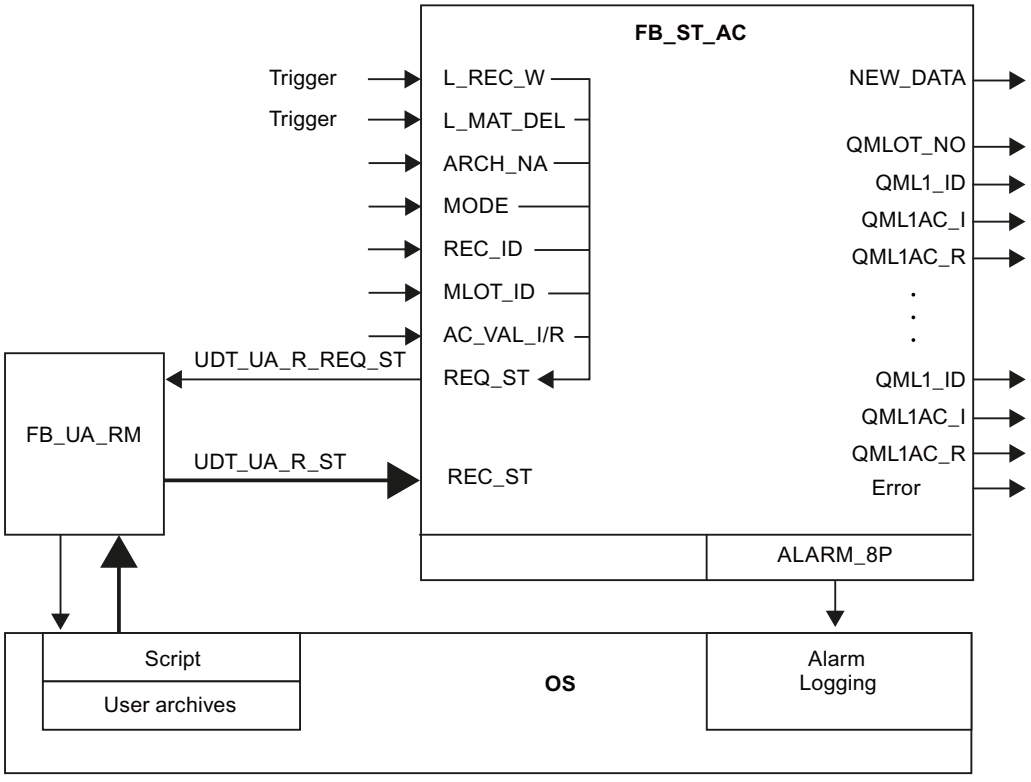
4.3.2 Generation

The "FB_ST_AC" block is not generated by the APF Engineering Tool. It is available in the APF library.

4.3.3 Functions

Overview

The figure below provides an overview of how the "FB_ST_AC" block works:



Booking / clearing

Booking is done at the block using the "FB_UA_RM" block. The following data are transferred to the "FB_UA_RM" block upon a positive edge at the "L_REC_W" input:

- Selected storage location ID ("REC_ID" input)
- Booking quantity (either "AC_VAL_I" input or "AC_VAL_R" input depending on value of "FORMAT" input).
  A positive booking quantity means "entering in the book"; a negative booking quantity means "clearing from the book".
- Material lot ID ("MLOT_ID" input)

These values are evaluated by the OS script and, in accordance with the accounting mode, either booked to or cleared from the specified storage location following a positive error check.

If errors have occurred, a least one error output is set and a corresponding message triggered.

Clearing: once the booking process is complete, part lots which have been cleared and their partial quantities are available for further processing at the 30 block outputs "QMLx_ID", "QMLxAC_I" and "QMLxAC_R" (x = 1..10).

Deleting material

If the storage location is empty (STOR_CONT column <= 0), you can use the "L_MAT_DEL" input to delete the material in the storage location. The storage location is then set to 0 in the user archive.

If the storage location is not empty (STOR_CONT column > 0), the job is not executed and an archive error is triggered.

Converting material

A material lot in a storage location and its base material can be converted to another material lot with new base material if the process allows.

Material conversion mode is activated with the block input "CH_MAT = 1".

Material is converted at the block using the "FB_UA_RM" block. The following data are transferred to the "FB_UA_RM" block upon a positive edge at the "L_REC_W" input:

- Selected storage location ID ("REC_ID" input),
- Booking quantity (either "AC_VAL_I" input or "AC_VAL_R" input depending on value of "FORMAT" input)
- New material lot ID ("MLOT_ID" input).

These values are evaluated by the OS script and, in accordance with the accounting mode, either booked or rebooked in the specified storage location following a positive error check.

If errors have occurred, a least one error output is set and a corresponding message triggered.

Specify "0" as the booking quantity to simply convert the material lot. If the booking quantity specified is greater than "0", this quantity is also booked. After booking, the storage location has only one material lot with the total quantity consisting of the quantity available at the storage location before booking and the specified booking quantity.

Rebooking material

There are two steps to "Rebooking material": "clearing" and "booking". Rebooking does not change the number of material lots required or their partial quantities. This means that if three material lots are cleared from the source storage location, these three material lots and their partial quantities are all booked to the destination storage location.

Material rebook mode is activated with the block input "AC_INOUT" = 1.

A material lot is rebooked at the block using the "FB_UA_RM" block. The following data are transferred to the "FB_UA_RM" block upon a positive edge at the "L_REC_W" input:

- Selected source storage location ID ("REC_ID" input)
- Selected destination storage location ID ("DST_ID" input) and
- Booking quantity (either "AC_VAL_I" input or "AC_VAL_R" input depending on value of "FORMAT" input)

The material lot ID ("MLOT_ID" input) does not need to be entered in this mode. These values are evaluated by the OS script and, in accordance with the accounting mode, either booked to or cleared from the specified storage location following a positive error check.

If errors have occurred, a least one error output is set and a corresponding message triggered.

Once the booking process is complete, part lots which have been rebooked and their partial quantities are available for further processing at the 30 block outputs "QMLx_ID", "QMLxAC_I" and "QMLxAC_R" (x = 1..10).

Converting units

You can convert units. This is possible if a convertible unit with the correct notation has been configured at the storage location in the OS editor screen and at the UOM block input. You can also define your own units, notations and conversion formulas in the C function FB_ConvertUOM.fct.

Unit	Pre-defined notations	Unit you can convert to
Liter	l, dm³, cdm, dm^3	Hectoliters (hl), cubic meters (m³, cm, m^3)
Hectoliter	hl	Liters (l, dm³, cdm, dm^3), cubic meters (m³, cm, m^3)
Cubic meter	m³, cm, m^3	Liters (l, dm³, cdm, dm^3), hectoliters (hl)
Kilogram	kg	Tons (t)
Tons	t	Kilograms (kg)

4.3.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_ST_AC" block:

Parameter	Description	Type	Default
SAMPLE_T	Sampling time [s]	REAL	1
RUNUPCYC	Number of startup cycles	INT	10
ID	ID of the block	INT	0
ARCH_NA	Archive name	STRING[16]	''
REC_ID	Data record number in user archive	DINT	0
DST_ID	Destination storage location ID in user archive	DINT	0
FORMAT	Data format content 0 = DINT, >1 = REAL	DINT	0
UOM	Unit	STRING[16]	''
BA_NA	Batch name	STRING[32]	''
BA_ID	Batch ID	DWORD	0
COUNT	Number of repetitions if an error occurs	INT	1
TIME_MON	Read job monitoring time	REAL	30.0
R_REC_ST	Input structure with job feedback from FB_UA_RM	UDT_UA_R_ST	
AC_INOUT	1 = "Material lot rebook" mode	BOOL	0
CH_MAT	1 = "Material conversion" mode	BOOL	0
L_REC_W	Interconnectable input: 1 = Write data request	BOOL	0
L_MAT_DEL	Interconnectable input: 1 = Delete material	BOOL	0

Parameter	Description	Type	Default
RESET	Reset data from write job	BOOL	0
MODE	Mode	WORD	0
MLOT_ID	Material lot ID	DINT	0
AC_VAL_I	Booking value in DINT format	DINT	0
AC_VAL_R	Booking value in REAL format	REAL	0
MSG_EVID	ALARM_8P: EV_ID	DWORD	0

"MODE" parameter

The "MODE" input parameter has the following functions:

Bit	Meaning
0	0: Clearing always executed without deleting the material 1: Material is automatically deleted as well if the storage location is empty after clearing
1..3	Not used
4	1 = Suppression of group errors with QCONTERR
5	1 = Suppression of group errors with QOV_DINT
6	1 = Suppression of "Alarm_8P" message with QCONTERR
7	1 = Suppression of "Alarm_8P" message with QOV_DINT
8 - 15	Not used

Output parameters

The following table shows the output parameters of the "FB_ST_AC" block:

Parameter	Description	Type	Default
QREC_R	Status of read data request	INT	0
QRUN_UP	Status display startup active = 1	BOOL	0
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0
QRELEASED	1= Function block released for processing	BOOL	1
QGR_ERR	Function block group error/fault	BOOL	0
QERR_VER	1 = Version check error	BOOL	0
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0
QERR	1 = Read error, data header: requested data and read data do not match	BOOL	0
QMONERR	1 = Read job monitoring timeout	BOOL	0
QMLIDERR	1 = Error "MLOT_ID = 0"	BOOL	0
QAVALERR	1 = Error "Booking value = 0"	BOOL	0
QRM_ERR	FB_UA_RM: 1 = Error, requested data and read data do not match	BOOL	0
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BSEND" coupling blocks	BOOL	0

Parameter	Description	Type	Default
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0
QMLOTMAX	1 = Error, max. number of material lots reached	BOOL	0
QREL_ERR	1 = Release error	BOOL	0
QMLOTERR	1 = Error, "MLOT" not available	BOOL	0
QCONTERR	1 = Error, store content <0 or >MAX = 1	BOOL	0
QFORMERR	1 = Error, booking value format	BOOL	0
QOV_DINT	1 = Error, "DINT" store content value overflow	BOOL	0
QMAT_ERR	1 = Error, material ID of material lot not the same as material ID of storage location	BOOL	0
QUOM_ERR	1 = Unit cannot be converted to unit in storage location	BOOL	0
QDST_ERR	1 = Destination storage location ID in "Material rebooking" mode is 0	BOOL	0
QREQ_RM	1 = Job active	BOOL	0
NEW_DATA	1= data have been updated	BOOL	0
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST	
QID	Read data: ID	INT	0
QARCH_NA	Read data: Archive name	STRING[16]	''
QARCH_NO	Read data: Archive number	DINT	0
QREC_ID	Read data: Data record number in user archive	DINT	0
QNO_REC	Read data: Number of rows to be read	DINT	0
QCOLUMN	Read data: Column number at which reading commences	DINT	0
QNO_COL	Read data: Number of columns to be read	DINT	0
QOFFSET	Read data: Offset, application-specific value	DINT	0
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0
QRESERVE	Read data: Reserve	DINT	0
QMLOT_NO	Number of possible material lots	DINT	0
QML1_ID	Part material lot ID 1	DINT	0
QML1AC_I	Partial quantity in part material lot 1 (integer)	DINT	0
QML1AC_R	Partial quantity in part material lot 1 (Real)	DINT	0
QML2_ID	Part material lot ID 2	DINT	0
QML2AC_I	Partial quantity in part material lot 2 (integer)	DINT	0
QML2AC_R	Partial quantity in part material lot 2 (Real)	DINT	0
QML3_ID	Part material lot ID 3	DINT	0
QML3AC_I	Partial quantity in part material lot 3 (integer)	DINT	0
QML3AC_R	Partial quantity in part material lot 3 (Real)	DINT	0
QML4_ID	Part material lot ID 4	DINT	0
QML4AC_I	Partial quantity in part material lot 4 (integer)	DINT	0

Parameter	Description	Type	Default
QML4AC_R	Partial quantity in part material lot 4 (Real)	DINT	0
QML5_ID	Part material lot ID 5	DINT	0
QML5AC_I	Partial quantity in part material lot 5 (integer)	DINT	0
QML5AC_R	Partial quantity in part material lot 5 (Real)	DINT	0
QML6_ID	Part material lot ID 6	DINT	0
QML6AC_I	Partial quantity in part material lot 6 (integer)	DINT	0
QML6AC_R	Partial quantity in part material lot 6 (Real)	DINT	0
QML7_ID	Part material lot ID 7	DINT	0
QML7AC_I	Partial quantity in part material lot 7 (integer)	DINT	0
QML7AC_R	Partial quantity in part material lot 7 (Real)	DINT	0
QML8_ID	Part material lot ID 8	DINT	0
QML8AC_I	Partial quantity in part material lot 8 (integer)	DINT	0
QML8AC_R	Partial quantity in part material lot 8 (Real)	DINT	0
QM9_ID	Part material lot ID 9	DINT	0
QM9AC_I	Partial quantity in part material lot 9 (integer)	DINT	0
QM9AC_R	Partial quantity in part material lot 9 (Real)	DINT	0
QML10_ID	Part material lot ID 10	DINT	0
QML10AC_I	Partial quantity in part material lot 10 (integer)	DINT	0
QML10AC_R	Partial quantity in part material lot 10 (Real)	DINT	0
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST	
MSG_STAT	ALARM_8P: STATUS	WORD	0
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0
QMSG_SUP	ALARM_8P: 1 = Message suppression active	BOOL	0
QMSG_ERR	ALARM_8P: ERROR	BOOL	0

In/out parameters

The following table shows the input and output parameters of the "FB_ST_AC" block:

Parameter	Description	Type	Default
APF_AS	APF function block version in AS	WORD	0
APF_OS	APF function block version in OS	WORD	0

4.3.5 Error handling

Block error

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals. In this case, the "NEW_DATA" output is not set.

The following error messages can be generated:

Request structure	Output	Meaning
	QPARAMF	Parameter assignment error; ID or TIME_MON <= 0
	QERR	Data error REQ_ID in request and receive structure not the same
MON_ERR	QMONERR	Runtime error; Monitoring timeout
	QRM_ERR	Error from FB_UA_RM: Data error, job data in WinCC communication interface do not correspond to original job data
	QRM_AERR	Error from FB_UA_RM: Archive error; job data in the WinCC communication interface reset by the OS, or script error in the OS
	QRM_CERR	Error from FB_UA_RM: Communication error; Error message from "BSEND" coupling blocks
	QRM_PERR	Error from FB_UA_RM: Parameter assignment error; incorrect ID_1, R_ID_1 or ID_2, R_ID_2 or duplicate REQ_ID

Booking errors

In contrast to normal troubleshooting, a special error output and the setting of the "NEW_DATA" output indicate those errors occurring during operation which are not related to communication between FB_ST_AC / FB_UA_RM / OS script:

Output	Meaning
QMLOTMAX	All storage locations full, booking not possible
QREL_ERR	Release error: Violation of a criterion for accounting mode valid at storage location
QMLOTERR	Material lot not available, booking not possible
QCONTERR	More booked/cleared than the number defined at the storage location (CONT_MAX or CONT_MIN column) When the target total quantity > CONT_MAX, the target total quantity is written to the user archive (STOR_CONT column). When the target total quantity < CONT_MIN, CONT_MIN is written to the user archive.
QFORMERR	Formats of the booking and the storage location do not match (e.g. Booking of REAL values in an INTEGER storage location)
QOV_DINT	DINT overflow: Total booked to storage location exceeds maximum DINT value limit
QMAT_ERR	Incorrect material in the storage location, material lot cannot be booked
QUOM_ERR	Unit cannot be converted to unit in storage location
QMLIDERR	Specified material lot = 0, booking job is not started
QAVALERR	Specified quantity = 0, booking job is not started
QDST_ERR	Specified destination storage location ID = 0 (only relevant in "Material lot rebook" mode)

4.3.6 Messaging

The "FB_ST_AC" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	QMLOTMAX or QREL_ERR or QMLOTERR or QCONTERR or QFORMERR or QOV_DINT or QMAT_ERR or QUOM_ERR	$$BlockComment$$ error booking data @7%d@ ID @4%d@ REC_ID @5%d@ MLOT_ID @6%d@
SIG_2	AS process control Message - fault	Write job monitoring timeout (QMONERR)	$$BlockComment$$ QMON_ERR runtime error write job ID @4%d@ REC_ID @5%d@ MLOT_ID @6%d@
SIG_3	AS process control Message - fault	requested data and read data do not match (QERR)	$$BlockComment$$ QERR data error write job ID @4%d@ REC_ID @5%d@ MLOT_ID @6%d@
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0 or (DST_ID = 0 and ACC_INOUT = 1)	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@ REC_ID @5%d@ MLOT_ID @6%d@
SIG_5	AS process control Message - error	Specified material lot ID <= 0 (QMLIDERR)	$$BlockComment$$ QMLIDERR error MLOT_ID <= 0 ID @4%d@ REC_ID @5%d@
SIG_6	AS process control Message - error	Booking value = 0 (QAVALERR), not relevant in material conversion mode	$$BlockComment$$ QAVALERR error booking value = 0 ID @4%d@ REC_ID @5%d@
SIG_7	AS process control Message - fault	Length of the read data does not match the length of the block outputs (QLEN_ERR)	$$BlockComment$$ QLEN_ERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@ MLOT_ID @6%d@
SIG_8	No message	-	-

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID

Associated value	Format	Value
SD_5	@5%d@	REC_ID
SD_6	@6%d@	MLOT_ID
SD_7	@7%d@	REQ_ERR
SD_8	-	
SD_9	-	
SD_10	-	

4.4 FB_STLOC

4.4.1 Description

Object properties

The following table shows the object properties of FB_STLOC:

Name	Number	Family	Module
FB_STLOC	FB1779	APF	Storage location management

Area of application

The AS function block "FB_STLOC" (storage location) is used to search for a material in storage locations from the AS.

The block offers the following functions:

- Sending search requests for a storage location to the OS
  - Freely definable search filter
  - Freely definable results sorting
- Searching the user archive on the OS for the specified storage location, listing storage locations found and reporting them to the AS
- Automatic or manual selection of a storage location from the selection listed and reserving the setpoint quantity to be stored at the storage location
- Canceling the reserved setpoint quantity when storage is complete.
- Monitoring data communication (generation of alarms)

Note

The physical storage is made via application-specific features, not by this block.

Startup characteristics

The "FB_STLOC" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

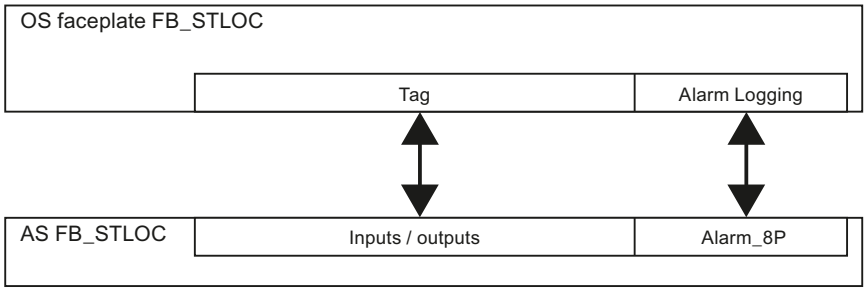
Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

For operator control and monitoring, signal exchange takes place between the AS and the OS via the inputs/outputs of the "FB_STLOC" block.

Operator control and monitoring tags are used for data communication between the AS and OS.

Corresponding scripts for the function block and its functionalities on the AS level are available for handling on the OS level.

The "ALARM_8P" structure is used to implement data communication with the OS message system.



4.4.2 Generation

The "FB_STLOC" block is not generated by the APF Engineering Tool. It is available in the APF library.

4.4.3 Functions

Overview

The AS function block "FB_STLOC" is used to search storage locations for a material from the AS. You specify the storage location using a search filter defined in the AS. The "FB_UA_RM" block sends the search query to the OS. The OS returns up to 10 storage locations matching the search. In automatic mode, the first storage location is selected. In manual mode, the operator selects the storage location using the corresponding faceplate. The data for the selected storage location are available in the AS at the outputs of the "FB_STLOC" block. When storage is started, the system checks the selected storage location against the filter criteria again and reserves the specified quantity of storage location content in the user archive. This quantity reservation is canceled again once storage is complete.

"Search" function

Once search criteria have been specified, the search is started with a positive edge at the "L_SEARCH" input of the block. The OS returns up to 10 storage locations matching the search. The number of storage locations found is output in the AS at the "QHITS" output of the block.

The following feedback is output when the "Search" function is completed:

- The search was successful: QSEL_POS = 1, QRUN_ERR = 0
- The search was not successful: QSEL_POS = 0, QRUN_ERR = 1

Selecting the storage location

There are two modes, "Manual" and "Auto":

- "Manual" mode
  The operator selects the required storage location by clicking on the corresponding line in the faceplate. The search can be started again with the "Search" button. Provided storage has not been started, you can change the storage location by starting a new search ("Search" button or edge change "L_SEARCH")
- "Auto" mode
  The first storage location is automatically selected. Provided storage has not been started, you can change the storage location by switching to manual mode and starting a new search ("Search" button or "L_SEARCH" edge change).

Once the storage location is selected, feedback is output with QSEL_OK=1; the data of the selected storage location are available for further processing at the outputs in the AS.

"START" function

Following a positive edge at the "L_START" output of the block, the system checks the selected storage location against the filter criteria and, if they match, reserves the setpoint quantity at the storage location. This quantity is reserved by adding the quantity to be stored to the "STOR_USE_IN" cell in the user archive. The data of the active storage location are available at the block outputs in the AS for further processing.

The following feedback is output when the "START" function is completed:

- The execution of the "START" function was successful:
  QIDLE = 0, QACTIVE = 1, QRUN_ERR = 0
- The execution of the "START" function was not successful:
  QIDLE = 1, QACTIVE = 0, QRUN_ERR = 1

If the check of the selected storage location is negative in automatic mode, the next storage location found in the list is automatically checked. This process continues until a suitable storage location is found in the list (of a maximum of 10). If no suitable storage location is found in the list, the search must be triggered again.

"END" function

Following a positive edge at the "L_END" input of the block, the quantity reservation at the storage location is canceled. The reservation is canceled by subtracting the quantity to be stored from the "STOR_USE_IN" cell in the user archive.

The following feedback is output when the "END" function is completed:

- The execution of the "END" function was successful:
  QIDLE = 0, QACTIVE = 1, QEND_ERR = 0
- The execution of the "END" function was not successful:
  QIDLE = 1, QACTIVE = 0, QEND_ERR = 1

User-defined filter/sorting

The filter ID specified at the "LFILTER" input parameter is activated at the "EN_FILT = 1" input.

Converting units

See the FB_ST_AC block (Page 83) for a description.

4.4.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_STLOC" block. Search criteria can be set via the LSTORGR to LVALUE6B parameters.

Parameter	Description	Type	Default	OS tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
NO_OF_PO	Number of storage locations for selection (1 to 10)	INT	10	X
ID	ID of the block	INT	0	-
ARCH_NA	Archive name	STRING[16]	''	-

Parameter	Description	Type	Default	OS tag
FORMAT	Data format, quantity	DINT	0	X
BA_NA	Batch name	STRING[32]	''	-
BA_ID	Batch ID	DWORD	0	-
COUNT	Number of repetitions if an error occurs	INT	1	-
TIME_MON	Monitoring time for read job	REAL	30.0	-
EN_FILT	Activates user-defined filter/sorting	BOOL	0	-
SEL_MODE	Activates 2-step mode	BOOL	0	x
R_REC_ST	Input structure with job feedback from FB_UA_RM	UDT_UA_R_ST		-
L_SEARCH	Interconnectable input: 1 = Search storage location for material	BOOL	0	-
L_START	Interconnectable input: 1 = Start storage	BOOL	0	-
L_END	Interconnectable input: 1 = Storage complete	BOOL	0	-
RESET	1 = Reset data	BOOL	0	-
LSTORGR	Interconnectable input: Storage group	DINT	0	-
LMACLREL	Interconnectable input: Release material class	DINT	0	-
LMATYREL	Interconnectable input: Release material type	DINT	0	-
LSTOSTA1	Interconnectable input: Storage location status 1	DINT	0	-
LSTOSTA2	Interconnectable input: Storage location status 2	DINT	0	-
LSTOSTA3	Interconnectable input: Storage location status 3	DINT	0	-
LSTOSTA4	Interconnectable input: Storage location status 4	DINT	0	-
LMATSTA1	Interconnectable input: Material status 1	DINT	0	-
LMATSTA2	Interconnectable input: Material status 2	DINT	0	-
LMATSTA3	Interconnectable input: Material status 3	DINT	0	-
LMATSTA4	Interconnectable input: Material status 4	DINT	0	-
LPROREL1	Interconnectable input: Procedure enable 1	DINT	0	-
LPROREL2	Interconnectable input: Procedure enable 2	DINT	0	-
LPROREL3	Interconnectable input: Procedure enable 3	DINT	0	-
LPROREL4	Interconnectable input: Procedure enable 4	DINT	0	-
LAND_OR	Interconnectable input: AND / OR	BOOL	0	-
LCON_0	Interconnectable input: Content <= 0	BOOL	0	-
LAMVAL_I	Interconnectable input: Quantity in DINT format	DINT	0	-
LAMVAL_R	Interconnectable input: Quantity in REAL format	REAL	0.0	-
LAM_UOM	Interconnectable input: Unit of measurement for quantity	STRING[16]	''	-
LFILTER	Interconnectable input: Filter ID	DINT	0	-
LVALUE1	Interconnectable input: Value 1 (integer)	DINT	0	-
LVALUE2	Interconnectable input: Value 2 (integer)	DINT	0	-
LVALUE3	Interconnectable input: Value 3 (real / float)	REAL	0.0	-
LVALUE4	Interconnectable input: Value 4 (real / float)	REAL	0.0	-
LVALUE5	Interconnectable input: Value 5 (date part 1)	DWORD	16#90010100	-

Parameter	Description	Type	Default	OS tag
LVALUE5B	Interconnectable input: Value 5 (date part 2)	DWORD	16#0	-
LVALUE6	Interconnectable input: Value 6 (date part 1)	DWORD	16#90010100	-
LVALUE6B	Interconnectable input: Value 6 (date part 2)	DWORD	16#0	-
OP_EN	Enabled operations	WORD	16#FFFF	-
HIDE_BUT	Hide buttons	WORD	0	-
MODE	Mode	WORD	0	-
MSG_EVID	ALARM_8P: EV_ID	DWORD	0	-

"MODE" parameter

You can define the mode of the block using the input word "MODE":

Bit	Name	Meaning
0	EDGE_START_NEC	0 = Positive edge at the "L_START" input required. 1 = No positive edge at the "L_START" input required.
1	QSEL_ERR_MSG	0 = OS QSEL_ERR message deactivated 1 = OS QSEL_ERR message is active
2	QRUN_ERR_MSG	0 = OS QRUN_ERR message deactivated 1 = OS QRUN_ERR message is active

"OP_EN" parameter

You can enable button operation in the faceplate with the input word "OP_EN". This enable is only effective if the mode allows button operation.

Bit	Name	Meaning
0	AUT	0 = "Auto" operation disabled 1 = "Auto" key, external enable for operation
1	MAN	0 = "Manual" operation disabled 1 = "Manual" key, external enable for operation
2		not used
3	POSITION	0 = "Storage location selection" operation disabled 1 = "Storage location selection", external enable for operation
4	SEARCH	0 = "Search" operation disabled 1 = "Search" key, external enable for operation

"HIDE_BUT" parameter

You can hide buttons in the faceplate with the input word "HIDE_BUT".

Bit	Name	Meaning
0	MANAUT	0 = "Auto" and "Manual" are shown 1 = "Auto" and "Manual" are hidden
1		not used
2	SEARCH	0 = "Search" is shown 1 = "Search" is hidden

Output parameters

The following table shows the output parameters of the "FB_STLOC" block:

Parameter	Description	Type	Default	OS tag
QREC_R	Status of read data request	INT	0	X
QRUN_UP	Status display startup active = 1	BOOL	0	-
QNOT_OK	Block flag "OK" not set, Error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	X
QGR_ERR	Function block group error/fault	BOOL	0	-
QERR_VER	1 = Version check error	BOOL	0	X
QPARAMF	1 = Parameter assignment error, ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0	-
QRERR	1 = Read error, Header: requested data and read data do not match	BOOL	0	-
QRMONERR	1 = Read job monitoring timeout	BOOL	0	-
QRLENERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0	-
QRM_ERR	FB_UA_RM: 1 = Read error, requested data and read data do not match	BOOL	0	-
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0	-
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0	-
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QREQ_RM	1 = Read job active	BOOL	0	-
QSEARCH	1 = Storage location search active	BOOL	0	x
QSTART	1 = Start storage / check storage location active	BOOL	0	x
QEND	1 = Storage complete active	BOOL	0	x
QMAN_AUT	Status mode 0 = Manual, 1 = Auto	BOOL	0	x
QEN_FILT	1 = User-defined filter/sorting activated	BOOL	0	x

Parameter	Description	Type	Default	OS tag
QHITPLUS	1 = More than 10 storage locations found	BOOL	0	x
QSEL_POS	1 = Storage location(s) found, selection of storage location possible	BOOL	0	x
QSEL_OK	1 = Storage location selected	BOOL	0	x
QSEL_ERR	1 = Error in storage location search	BOOL	0	x
QRUN_ERR	1 = Error starting storage	BOOL	0	x
QEND_ERR	1 = Error completing storage	BOOL	0	x
QIDLE	1 = Free - storage not active	BOOL	0	x
QACTIVE	1 = Allocated - storage active	BOOL	0	x
NEW_DATA	1= data have been read	BOOL	0	-
QROW_SEL	Selected rows (storage locations)	WORD	16#0	-
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST		-
STORGR	Storage group	DINT	0	x
MACL_REL	Release material class	DINT	0	x
MATY_REL	Release material type	DINT	0	x
STO_STA1	Storage location status 1	DINT	0	x
STO_STA2	Storage location status 2	DINT	0	x
STO_STA3	Storage location status 3	DINT	0	x
STO_STA4	Storage location status 4	DINT	0	x
MAT_STA1	Material status 1	DINT	0	x
MAT_STA2	Material status 2	DINT	0	x
MAT_STA3	Material status 3	DINT	0	x
MAT_STA4	Material status 4	DINT	0	x
PRO_REL1	Procedure enable 1	DINT	0	x
PRO_REL2	Procedure enable 2	DINT	0	x
PRO_REL3	Procedure enable 3	DINT	0	x
PRO_REL4	Procedure enable 4	DINT	0	x
AND_OR	AND / OR	BOOL	0	x
CON_0	Content <= 0	BOOL	0	x
AM_VAL_I	Quantity in DINT format	DINT	0	x
AM_VAL_R	Quantity in REAL format	REAL	0.0	x
AM_UOM	Unit of measurement for quantity	STRING[16]	''	x
FILTER	Filter ID	DINT	0	x
VALUE1	Value 1 (integer)	DINT	0	x
VALUE2	Value 2 (integer)	DINT	0	x
VALUE3	Value 3 (real / float)	REAL	0.0	x
VALUE4	Value 4 (real / float)	REAL	0.0	x
VALUE5	Value 5 (date part 1)	DWORD	16#90010100	x
VALUE5B	Value 5 (date part 2)	DWORD	16#0	x
VALUE6	Value 6 (date part 1)	DWORD	16#90010100	x
VALUE6B	Value 6 (date part 2)	DWORD	16#0	x

Parameter	Description	Type	Default	OS tag
QID	Read data: ID	INT	0	-
QARCH_NA	Read data: Archive name	STRING[16]	''	-
QARCH_NO	Read data: Archive number	DINT	0	-
QREC_ID	Read data: Data record number in user archive	DINT	0	-
QNO_REC	Read data: Number of rows to be read	DINT	0	-
QCOLUMN	Read data: Column number at which reading commences	DINT	0	-
QNO_COL	Read data: Number of columns to be read	DINT	0	-
QOFFSET	Read data: Offset, application-specific value	DINT	0	-
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0	-
QRESERVE	Read data: Reserve	DINT	0	-
S_ID_SEL	Selected storage location: ID	DINT	0	x
S_NA_SEL	Selected storage location: Name	STRING[64]	''	x
M_ID_SEL	Selected storage location: Material ID	DINT	0	x
TAGN_SEL	Selected storage location: TAG name	STRING[128]	''	x
UOM_SEL	Selected storage location: Unit of the measured value	STRING[16]	''	x
FORM_SEL	Selected storage location: Data format	DINT	0	x
IAM_SEL	Selected storage location: Quantity in DINT format	DINT	0	x
ICON_SEL	Selected storage location: Content in DINT format	DINT	0	x
IMAX_SEL	Selected storage location: Maximum fill level in DINT format	DINT	0	x
RAM_SEL	Selected storage location: Quantity in REAL format	REAL	0.0	x
RCON_SEL	Selected storage location: Content in REAL format	REAL	0.0	x
RMAX_SEL	Selected storage location: Maximum fill level in REAL format	REAL	0.0	x
QS_ID	Active storage location: ID	DINT	0	x
QS_NA	Active storage location: Name	STRING[64]	''	x
QM_ID	Active storage location: Material ID	DINT	0	x
QTAG	Active storage location: TAG name	STRING[128]	''	x
QUOM	Active storage location: Unit of the measured value	STRING[16]	''	x
QFORM	Active storage location: Data format	DINT	0	x
QIAM	Active storage location: Quantity in DINT format	DINT	0	x
QICON	Active storage location: Content in DINT format	DINT	0	x
QIMAX	Active storage location: Maximum fill level in DINT format	DINT	0	x
QRAM	Active storage location: Quantity in REAL format	REAL	0.0	x
QRCON	Active storage location: Content in REAL format	REAL	0.0	x
QRMAX	Active storage location: Maximum fill level in REAL format	REAL	0.0	x

Parameter	Description	Type	Default	OS tag
QHITS	0 to 10 - number of matching storage locations for display / selection	DINT	0	x
QHIT_MAX	0 to 11 - number of matching storage locations 11 = There are more than 10 matching storage locations	DINT	0	x
S_ID_P01	Storage location found P01: ID	DINT	0	x
S_NA_P01	Storage location found P01: Name	STRING[64]	''	x
M_ID_P01	Storage location found P01: Material ID	DINT	0	x
TAGN_P01	Storage location found P01: TAG name	STRING[128]	''	x
UOM_P01	Storage location found P01: Unit of the measured value	STRING[16]	''	x
FORM_P01	Storage location found P01: Data format	DINT	0	x
IAM_P01	Storage location found P01: Quantity in DINT format	DINT	0	x
ICON_P01	Storage location found P01: Content in DINT format	DINT	0	x
IMAX_P01	Storage location found P01: Maximum fill level in DINT format	DINT	0	x
RAM_P01	Storage location found P01: Quantity in REAL format	REAL	0.0	x
RCON_P01	Storage location found P01: Content in REAL format	REAL	0.0	x
RMAX_P01	Storage location found P01: Maximum fill level in REAL format	REAL	0.0	x
S_ID_Pn	Storage location found Pn (n = 02 to 10): ID	DINT	0	x
S_NA_Pn	Storage location found Pn (n = 02 to 10): Name	STRING[64]	''	x
M_ID_Pn	Storage location found Pn (n = 02 to 10): Material ID	DINT	0	x
TAGN_Pn	Storage location found Pn (n = 02 to 10): TAG name	STRING[128]	''	x
UOM_Pn	Storage location found Pn (n = 02 to 10): Unit of the measured value	STRING[16]	''	x
FORM_Pn	Storage location found Pn (n = 02 to 10): Data format	DINT	0	x
IAM_Pn	Storage location found Pn (n = 02 to 10): Quantity in DINT format	DINT	0	x
ICON_Pn	Storage location found Pn (n = 02 to 10): Content in DINT format	DINT	0	x
IMAX_Pn	Storage location found Pn (n = 02 to 10): Maximum fill level in DINT format	DINT	0	x
RAM_Pn	Storage location found Pn (n = 02 to 10): Quantity in REAL format	REAL	0.0	x

Parameter	Description	Type	Default	OS tag
RCON_Pn	Storage location found Pn (n = 02 to 10): Content in REAL format	REAL	0.0	x
RMAX_Pn	Storage location found Pn (n = 02 to 10): Maximum fill level in REAL format	REAL	0.0	x
EN_AUT	Operator control enable: Auto	BOOL	0	x
EN_MAN	Operator control enable: Manual	BOOL	0	x
EN_SEAR	Operator control enable: Search	BOOL	0	x
EN_POS	Operator control enable: Selected row number	BOOL	0	x
DIS_MAAU	1 = Display of manual / auto buttons in the face-plate	BOOL	0	x
DIS_SEAR	1 = Display of search button in the faceplate	BOOL	0	x
QPOS	Selected row number	INT	0	x
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST		-
MSG_STAT	ALARM_8P: STATUS	WORD	0	-
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0	-
QMSG_SUP	ALARM_8P: 1 = Message suppression active	BOOL	0	-
QMSG_ERR	ALARM_8P: ERROR	BOOL	0	-

In/out parameters

The following table shows the in/out parameters of the "FB_STLOC" block:

Parameter	Description	Type	Default	OS tag
OPMANAUT	Operator: 0 = manual, 1 = auto	BOOL	0	x
OPSEARCH	Operator: 1 = Search	BOOL	0	x
OP_POS	Operator: Row number selection	INT	0	x
ROW_SEL	Selected rows (storage locations)	WORD	16#0	x
APF_AS	APF function block version in AS	WORD	0	X
APF_OS	APF function block version in OS	WORD	0	X

4.4.5 Error handling

Block error

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals.

The following error messages can be generated:

Request structure	Output	Meaning
	QPARAMF	Parameter assignment error; ID or ARCH_NA=' ' or TIME_MON <= 0
	QRLENERR	Data block length of the receive data is not correct
	QRERR	Data error Header data in request and receive structure not the same
MON_ERR	QMONERR	Runtime error; Monitoring timeout
	QRM_ERR	Error from FB_UA_RM: Data error, job data in WinCC communication interface do not correspond to original job data
	QRM_AERR	Error from FB_UA_RM: Archive error; job data in the WinCC communication interface reset by the OS, or script error in OS
	QRM_CERR	Error from FB_UA_RM: Communication error; Error message from "BRCV" coupling blocks
	QRM_PERR	Error from FB_UA_RM: Parameter assignment error; incorrect ID_1, R_ID_1 or ID_2, R_ID_2 or duplicate REQ_ID or, with activated filter (EN_FILT=1), filter ID < 0

4.4.6 Messaging

The "FB_STLOC" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - error	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QLEN_ERR data block length of the receive data is not correct ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_2	AS process control Message - error	Read job monitoring timeout	$$BlockComment$$ QMONERR runtime error read job ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_3	AS process control Message - error	requested data and read data do not match	$$BlockComment$$ QERR data error read job ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or ARCH_NA=' ' or TIME_MON=0 or with activated filter (EN_FILT=1), filter ID < 0	$$BlockComment$$ QPARAMF parameter assignment error ID @4

Message	Message class	Event	Message text
SIG_5	AS process control Message - error	QRUN_ERR: Error starting storage	$$BlockComment$$ QRUN_ERR ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_6	AS process control Message - error	QEND_ERR: Error completing storage	$$BlockComment$$ QEND_ERR ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_7	AS process control Message - error	QSEL_ERR: Error during search	$$BlockComment$$ QSEL_ERR ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7
SIG_8	Operating message	QACTIVE: Storage active	$$BlockComment$$ QACTIVE ID @4 REC_ID "@5 STORGR @6 quantity @8 / @9 @7

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	@6%d@	Storage group
SD_7	@7%s@	UOM
SD_8	@8%d@	Quantity in DINT format
SD_9	@9%f@	Quantity in REAL format
SD_10	-	

4.5 FB_STMAT

4.5.1 Description

Object properties

The following table shows the object properties of FB_STMAT:

Name	Number	Family	Module
FB_STMAT	FB1780	APF	Storage location management

Area of application

The AS function block "FB_STMAT" (storage material) is used to search materials in storage locations from the AS.

The block offers the following functions:

- Sending search requests for a material to the OS
  - Freely definable search filter
  - Freely definable results sorting
- Searching the user archive on the OS for the specified material, listing storage locations found and reporting them to the AS
- Automatic or manual selection of a storage location from the selection listed and reserving the setpoint quantity to be exported at the storage location
- Canceling the reserved setpoint quantity when export is complete.
- Monitoring data communication (generation of alarms)

Note

The actual export is made via application-specific mechanisms, not by this block.

Startup characteristics

The "FB_STMAT" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

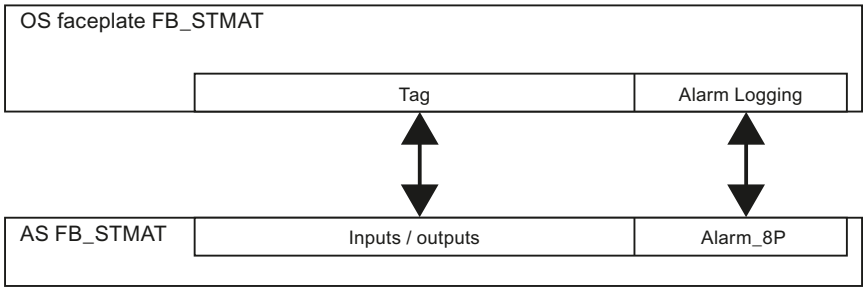
Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

For operator control and monitoring, signal exchange takes place between the AS and the OS via the inputs/outputs of the "FB_STMAT" block.

Operator control and monitoring tags are used for data communication between the AS and OS.

Corresponding scripts for the function block and its functionalities on the AS level are available for handling on the OS level.

The "ALARM_8P" structure is used to implement data communication with the OS message system.



4.5.2 Generation

The "FB_STMAT" block is not generated by the APF Engineering Tool. It is available in the APF library.

4.5.3 Functions

Overview

The "FB_STMAT" block is used to search for materials in storage locations from the AS. You specify the required material using a search filter defined in the AS. The "FB_UA_RM" block sends the search query to the OS. The OS returns up to 10 storage locations matching the search. In automatic mode, the storage location with the lowest material lot ID is selected. In manual mode, the operator selects the storage location using the corresponding faceplate. The data for the selected storage location are available in the AS at the outputs of the "FB_STMAT" block. When export is started, the system checks the selected storage location against the filter criteria and reserves the specified quantity in storage location content in the user archive. This quantity reservation is canceled again once export is complete.

User-defined filter/sorting

User-defined filter / sorting can be activated for the search (EN_FILT = 1). The filter ID is specified at the "LFILTER" input.

Use of remainder

Use of remainder is activated at the "EN_REM = 1" input. The maximum number of storage locations to be included with remainders is specified at the "SPLIT_PT" input parameter. When the function is started, the selected storage locations are reserved with the specified partial quantities. The end of partial quantity export is signaled by the "L_NEXT = 1" input, and this cancels the quantity reservation. The next storage location is then available at the outputs of the "FB_STMAT" block.

Triggering the "Search" function

Once search criteria have been specified, the search is started with a positive edge at the "L_SEARCH" input of the block. The OS returns up to 10 storage locations matching the search. The number of storage locations found is output in the AS at the "QHITS" output of the block.

When use of remainder is activated (EN_REM = 1) and in accordance with the input parameter "SPLIT_PT", a search is started for storage locations with remainders (number = SPLIT_PT), and storage locations which cover the quantity still required (= setpoint quantity minus all remainders).

User-defined filters / sorting can be activated for the search (EN_FILT = 1). The filter ID at the "LFILTER" input is entered in the search for storage locations with remainders, and the filter at the "LFILTER2" input for the search for storage locations covering the remaining quantity required.

The following feedback is output when the "Search" function is completed:

- The search was successful: QSEL_POS = 1, QRUN_ERR = 0
- The search was not successful: QSEL_POS = 0, QRUN_ERR = 1

Selecting the storage location

There are two modes, "Manual" and "Auto":

- "Manual" mode
  The operator selects the required storage location by clicking on the corresponding line in the faceplate. The search can be started again with the "Search" key. Provided the export has not been started, you can change the storage location by starting a new search ("Search" or edge change "L_SEARCH")
- "Auto" mode
  The storage location with the lowest material lot ID is automatically selected. Provided export has not yet been started, you can change the storage location by switching to manual mode and starting a new search ("Search" or "L_SEARCH" edge change).
  When use of remainder is activated (EN_REM = 1), the selection in "Auto" depends on the filters/sorting.

Once the storage location is selected, feedback is output with QSEL_OK=1; the data of the selected storage location are available for further processing at the outputs in the AS.

Triggering the "START" function

Following a positive edge at the "L_START" input of the block, the selected storage location is checked against the filter criteria. If they match, the setpoint quantity is reserved at the storage location. This quantity is reserved by adding the quantity to be exported to the "STOR_USE_OUT" cell in the user archive. The data of the active storage location are available at the block outputs in the AS for further processing.

If the check of the selected storage location is negative in automatic mode, the next storage location found in the list is automatically checked. This process continues until a suitable storage location is found in the list (of a maximum of 10). If no suitable storage location is found in the list, the search must be triggered again.

If use of remainder is activated (EN_REM = 1), the specified partial quantities and the content of the selected storage locations are reserved upon start. Automatic repetition is not available.

The following feedback is output when the "START" function is completed:

- The execution of the "START" function was successful:
  QIDLE = 0, QACTIVE = 1, QRUN_ERR = 0
- The execution of the "START" function was not successful:
  QIDLE = 1, QACTIVE = 0, QRUN_ERR = 1

Triggering the "NEXT" function

L_NEXT 0 -> 1 (only with use of remainder EN_REM = 1))

The end of partial quantity export is signaled by the "L_NEXT" input. Following a positive edge at the "L_NEXT" input, the quantity reservation at the active storage location is canceled for the partial quantity. The reservation is canceled by subtracting the quantity to be exported from the "STOR_USE_OUT" cell in the user archive. The next storage location is then available at the outputs of the "FB_STMAT" block.

Once the next storage location is made available at the outputs of the block, the counter value at the output "QROW_ACT" is incremented by 1. The output "QNEXTEND" is set to "1" for the last selected storage location. Only the "END" function can then be executed.

Triggering the "END" function

Following a positive edge at the "L_END" input of the block, the quantity reservation at the storage location is canceled. The reservation is canceled by subtracting the quantity to be exported from the "STOR_USE_OUT" cell in the user archive.

The following feedback is output when the "END" function is completed:

- The execution of the "END" function was successful:
  QIDLE = 0, QACTIVE = 1, QEND_ERR = 0
- The execution of the "END" function was not successful:
  QIDLE = 1, QACTIVE = 0, QEND_ERR = 1

Converting units

See the FB_ST_AC block (Page 83) for a description.

4.5.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_STMAT" block. Search criteria can be set via the LMAT to LFILTER2 parameters.

Parameter	Description	Type	Default	OS tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
NO_OF_PO	Number of storage locations for selection (1 to 10)	INT	10	x
ID	ID of the block	INT	0	-
ARCH_NA	Archive name	STRING[16]	''	-
FORMAT	Data format, quantity	DINT	0	x
BA_NA	Batch name	STRING[32]	''	-
BA_ID	Batch ID	DWORD	0	-
COUNT	Number of repetitions if an error occurs	INT	1	-
TIME_MON	Monitoring time for read job	REAL	30.0	-
EN_FILT	Activates user-defined filter/sorting	BOOL	0	-
SEL_MODE	Activates 2-step mode	BOOL	0	X
EN_REM	Activates use of remainder	BOOL	0	X
SPLIT_PT	Number of remainders to be included	DINT	0	X
R_REC_ST	Input structure with job feedback from FB_UA_RM	UDT_UA_R_ST		-
L_SEARCH	Interconnectable input: 1 = Search material	BOOL	0	-
L_START	Interconnectable input: 1 = Start export	BOOL	0	-
L_NEXT	Interconnectable input: 1 = Switch export to next storage location	BOOL	0	-
L_END	Interconnectable input: 1 = Export complete	BOOL	0	-
RESET	1 = Reset data	BOOL	0	-
LMAT	Interconnectable input: Material	DINT	0	-
LMAT_LOT	Interconnectable input: Material lot	DINT	0	-
LSTOSTA1	Interconnectable input: Storage location status 1	DINT	0	-
LSTOSTA2	Interconnectable input: Storage location status 2	DINT	0	-
LSTOSTA3	Interconnectable input: Storage location status 3	DINT	0	-
LSTOSTA4	Interconnectable input: Storage location status 4	DINT	0	-
LMATSTA1	Interconnectable input: Material status 1	DINT	0	-

Parameter	Description	Type	Default	OS tag
LMATSTA2	Interconnectable input: Material status 2	DINT	0	-
LMATSTA3	Interconnectable input: Material status 3	DINT	0	-
LMATSTA4	Interconnectable input: Material status 4	DINT	0	-
LPROREL1	Interconnectable input: Procedure enable 1	DINT	0	-
LPROREL2	Interconnectable input: Procedure enable 2	DINT	0	-
LPROREL3	Interconnectable input: Procedure enable 3	DINT	0	-
LPROREL4	Interconnectable input: Procedure enable 4	DINT	0	-
LAND_OR	Interconnectable input: AND / OR	BOOL	0	-
LAMVAL_I	Interconnectable input: Quantity in DINT format	DINT	0	-
LAMVAL_R	Interconnectable input: Quantity in REAL format	REAL	0.0	-
LAM_UOM	Interconnectable input: Unit of measurement for quantity	STRING[16]	''	-
LFILTER	Interconnectable input: Filter ID for material search without use of remainder or first filter ID for search for remaining amount with use of remainder	DINT	0	-
LVALUE1	Interconnectable input: Value 1 (integer)	DINT	0	-
LVALUE2	Interconnectable input: Value 2 (integer)	DINT	0	-
LVALUE3	Interconnectable input: Value 3 (real / float)	REAL	0.0	-
LVALUE4	Interconnectable input: Value 4 (real / float)	REAL	0.0	-
LVALUE5	Interconnectable input: Value 5 (date part 1)	DWORD	16#90010100	-
LVALUE5B	Interconnectable input: Value 5 (date part 2)	DWORD	16#0	-
LVALUE6	Interconnectable input: Value 6 (date part 1)	DWORD	16#90010100	-
LVALUE6B	Interconnectable input: Value 6 (date part 2)	DWORD	16#0	-
LFILTER2	Interconnectable input: 2. filter ID for search for full amount with use of remainder	DINT	0	-
OP_EN	Enabled operations	WORD	16#FFFF	-
HIDE_BUT	Hide buttons	WORD	0	-
MODE	Mode	WORD	0	-
MSG_EVID	ALARM_8P: EV_ID	DWORD	0	-

"MODE" parameter

You can define the mode of the block using the input word "MODE":

Bit	Name	Meaning
0	EDGE_START_NEC	0 = Positive edge at the "L_START" input required. 1 = No positive edge at the "L_START" input required.
1	QSEL_ERR_MSG	0 = OS QSEL_ERR message deactivated 1 = OS QSEL_ERR message is active
2	QRUN_ERR_MSG	0 = OS QRUN_ERR message deactivated 1 = OS QRUN_ERR message is active

"OP_EN" parameter

You can enable button operation in the faceplate with the input word "OP_EN". This enable is only effective if the mode allows button operation.

Bit	Name	Meaning
0	AUT	0 = "Auto" operation disabled 1 = "Auto" key, external enable for operation
1	MAN	0 = "Manual" operation disabled 1 = "Manual" key, external enable for operation
2		not used
3	POSITION	0 = "Storage location selection" operation disabled 1 = "Storage location selection", external enable for operation
4	SEARCH	0 = "Search" operation disabled 1 = "Search" key, external enable for operation

"HIDE_BUT" parameter

You can hide buttons in the faceplate with the input word "HIDE_BUT".

Bit	Name	Meaning
0	MANAUT	0 = "Auto" and "Manual" are shown 1 = "Auto" and "Manual" are hidden
1		not used
2	SEARCH	0 = "Search" is shown 1 = "Search" is hidden

Output parameters

The following table shows the output parameters of the "FB_STMAT" block:

Parameter	Description	Type	Default	OS tag
QREC_R	Status of read data request	INT	0	x
QRUN_UP	Status display startup active = 1	BOOL	0	-

Parameter	Description	Type	Default	OS tag
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	x
QGR_ERR	Function block group error/fault	BOOL	0	-
QERR_VER	1 = Version check error	BOOL	0	x
QPARAMF	1 = Parameter assignment error, ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0	-
QRERR	1 = Read error, Header: requested data and read data do not match	BOOL	0	-
QRMONERR	1 = Read job monitoring timeout	BOOL	0	-
QRLENERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0	-
QRM_ERR	FB_UA_RM: 1 = Read error, requested data and read data do not match	BOOL	0	-
QRM_AERR	FB_UA_RM: 1 = Archive error, Invalid data in receive buffer	BOOL	0	-
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0	-
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QREQ_RM	1 = Read job active	BOOL	0	-
QSEARCH	1 = Material search active	BOOL	0	x
QSTART	1 = Start export / check storage location active	BOOL	0	x
QNEXTEND	1 = Last storage location for export active	BOOL	0	x
QEND	1 = Export complete active	BOOL	0	x
QMAN_AUT	Status mode 0 = Manual, 1 = Auto	BOOL	0	x
QEN_FILT	1 = User-defined filter/sorting activated	BOOL	0	x
EN_REM	1 = Use of remainder activated	BOOL	0	x
QHITPLUS	1 = More than 10 storage locations found	BOOL	0	x
QSEL_POS	1 = Storage location(s) found, selection of storage location possible	BOOL	0	x
QSEL_OK	1 = Storage location selected	BOOL	0	x
QSEL_ERR	1 = Error in material search	BOOL	0	x
QRUN_ERR	1 = Error starting export	BOOL	0	x
QEND_ERR	1 = Error completing export	BOOL	0	x
QIDLE	1 = Free - export not active	BOOL	0	x
QACTIVE	1 = Allocated - export active	BOOL	0	x
NEW_DATA	1= data have been read	BOOL	0	-
QROW_SEL	Selected rows (storage locations)	WORD	16#0	-
QROW_NO	Number of selected rows (storage locations)	INT	0	x
QROW_ACT	Active row (consecutive number)	INT	0	x
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST		

Parameter	Description	Type	Default	OS tag
MAT	Material	DINT	0	x
MAT_LOT	Material lot	DINT	0	x
STO_STA1	Storage location status 1	DINT	0	x
STO_STA2	Storage location status 2	DINT	0	x
STO_STA3	Storage location status 3	DINT	0	x
STO_STA4	Storage location status 4	DINT	0	x
MAT_STA1	Material status 1	DINT	0	x
MAT_STA2	Material status 2	DINT	0	x
MAT_STA3	Material status 3	DINT	0	x
MAT_STA4	Material status 4	DINT	0	x
PRO_REL1	Procedure enable 1	DINT	0	x
PRO_REL2	Procedure enable 2	DINT	0	x
PRO_REL3	Procedure enable 3	DINT	0	x
PRO_REL4	Procedure enable 4	DINT	0	x
AND_OR	AND / OR	BOOL	0	x
AM_VAL_I	Quantity in DINT format	DINT	0	x
AM_VAL_R	Quantity in REAL format	REAL	0.0	x
AM_UOM	Unit of measurement for quantity	STRING[16]	''	x
FILTER	ID for material search without use of remainder or 1. first filter ID for search for remaining amount with use of remainder	DINT	0	x
VALUE1	Value 1 (integer)	DINT	0	x
VALUE2	Value 2 (integer)	DINT	0	x
VALUE3	Value 3 (real / float)	REAL	0.0	x
VALUE4	Value 4 (real / float)	REAL	0.0	x
VALUE5	Value 5 (date part 1)	DWORD	16#90010100	x
VALUE5B	Value 5 (date part 2)	DWORD	16#0	x
VALUE6	Value 6 (date part 1)	DWORD	16#90010100	x
VALUE6B	Value 6 (date part 2)	DWORD	16#0	x
FILTER2	2. filter ID for search for full amount with use of remainder	DINT	0	x
QID	Read data: ID	INT	0	-
QARCH_NA	Read data: Archive name	STRING[16]	''	-
QARCH_NO	Read data: Archive number	DINT	0	-
QREC_ID	Read data: Data record number in user archive	DINT	0	-
QNO_REC	Read data: Number of rows to be read	DINT	0	-
QCOLUMN	Read data: Column number at which reading commences	DINT	0	-
QNO_COL	Read data: Number of columns to be read	DINT	0	-
QOFFSET	Read data: Offset, application-specific value	DINT	0	-
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0	-
QRESERVE	Read data: Reserve	DINT	0	-
S_ID_SEL	Selected first storage location: ID	DINT	0	x

Parameter	Description	Type	Default	OS tag
S_NA_SEL	Selected first storage location: Name	STRING[64]	''	x
M_ID_SEL	Selected storage location: Material ID	DINT	0	x
MLOT_SEL	Selected first storage location: Material lot ID	DINT	0	x
TAGN_SEL	Selected first storage location: TAG name	STRING[128]	''	x
UOM_SEL	Selected first storage location: Unit of the measured value	STRING[16]	''	x
FORM_SEL	Selected first storage location: Data format	DINT	0	x
IAM_SEL	Selected first storage location: Quantity in DINT format	DINT	0	x
ICON_SEL	Selected first storage location: Content in DINT format	DINT	0	x
IMIN_SEL	Selected first storage location: Minimum fill level in DINT format	DINT	0	x
IRAM_SEL	Selected first storage location: Remainder in DINT format	DINT	0	x
RAM_SEL	Selected first storage location: Quantity in REAL format	REAL	0.0	x
RCON_SEL	Selected first storage location: Content in REAL format	REAL	0.0	x
RMIN_SEL	Selected first storage location: Minimum fill level in REAL format	REAL	0.0	x
RRAM_SEL	Selected first storage location: Remainder in REAL format	REAL	0.0	x
S_IDSLn	Selected nth storage location (n = 02 to 10): ID	DINT	0	x
S_NASLn	Selected nth storage location (n = 02 to 10): Name	STRING[64]	''	x
M_IDSLn	Selected nth storage location (n = 02 to 10): Material ID	DINT	0	x
MLOTSLn	Selected nth storage location (n = 02 to 10): Material lot ID	DINT	0	x
TAGNSLn	Selected nth storage location (n = 02 to 10): TAG name	STRING[128]	''	x
UOMSLn	Selected nth storage location (n = 02 to 10): Unit of the measured value	STRING[16]	''	x
FORMSLn	Selected nth storage location (n = 02 to 10): Data format	DINT	0	x
IAMSLn	Selected nth storage location (n = 02 to 10): Quantity in DINT format	DINT	0	x
ICONSLn	Selected nth storage location (n = 02 to 10): Content in DINT format	DINT	0	x
IMINSLn	Selected nth storage location (n = 02 to 10): Minimum fill level in DINT format	DINT	0	x
IRAMSLn	Selected nth storage location (n = 02 to 10): Remainder in DINT format	DINT	0	x

Parameter	Description	Type	Default	OS tag
RAMSLn	Selected nth storage location (n = 02 to 10): Quantity in REAL format	REAL	0.0	x
RCONSLn	Selected nth storage location (n = 02 to 10): Content in REAL format	REAL	0.0	x
RMINSLn	Selected nth storage location (n = 02 to 10): Minimum fill level in REAL format	REAL	0.0	x
RRAMSLn	Selected nth storage location (n = 02 to 10): Remainder in REAL format	REAL	0.0	x
QS_ID	Active storage location: ID	DINT	0	x
QS_NA	Active storage location: Name	STRING[64]	''	x
QM_ID	Active storage location: Material ID	DINT	0	x
QMLOT_ID	Active storage location: Material lot ID	DINT	0	x
QTAG	Active storage location: TAG name	STRING[128]	''	x
QUOM	Active storage location: Unit of the measured value	STRING[16]	''	x
QFORM	Active storage location: Data format	DINT	0	x
QIAM	Active storage location: Quantity in DINT format	DINT	0	x
QICON	Active storage location: Content in DINT format	DINT	0	x
QIMIN	Active storage location: Minimum fill level in DINT format	DINT	0	x
QIRAM	Active storage location: Remainder in DINT format	DINT	0	x
QRAM	Active storage location: Quantity in REAL format	REAL	0.0	x
QRCON	Active storage location: Content in REAL format	REAL	0.0	x
QRRAM	Active storage location: Remainder in REAL format	REAL	0.0	x
QRMIN	Active storage location: Minimum fill level in REAL format	REAL	0.0	x
QHITS	0 to 10 - number of matching storage locations for display / selection	DINT	0	x
QHIT_MAX	0 to 11 - number of matching storage locations 11 = There are more than 10 matching storage locations	DINT	0	x
S_ID_P01	Storage location found P01: ID	DINT	0	x
S_NA_P01	Storage location found P01: Name	STRING[64]	''	x
M_ID_P01	Storage location found P01: Material ID	DINT	0	x
MLOT_P01	Storage location found P01: Material lot ID	DINT	0	x
TAGN_P01	Storage location found P01: TAG name	STRING[128]	''	x
UOM_P01	Storage location found P01: Unit of the measured value	STRING[16]	''	x
FORM_P01	Storage location found P01: Data format	DINT	0	x
IAM_P01	Storage location found P01: Quantity in DINT format	DINT	0	x
ICON_P01	Storage location found P01: Content in DINT format	DINT	0	x

Parameter	Description	Type	Default	OS tag
IMIN_P01	Storage location found P01: Minimum fill level in DINT format	DINT	0	x
IRAM_P01	Storage location found P01: Remainder in DINT format	DINT	0	x
RAM_P01	Storage location found P01: Quantity in REAL format	REAL	0.0	x
RCON_P01	Storage location found P01: Content in REAL format	REAL	0.0	x
RMIN_P01	Storage location found P01: Minimum fill level in REAL format	REAL	0.0	x
RRAM_P01	Storage location found P01: Remainder in REAL format	REAL	0.0	x
S_ID_Pn	Storage location found Pn (n = 02 to 10): ID	DINT	0	x
S_NA_Pn	Storage location found Pn (n = 02 to 10): Name	STRING[64]	''	x
M_ID_Pn	Storage location found Pn (n = 02 to 10): Material ID	DINT	0	x
MLOT_Pn	Storage location found Pn (n = 02 to 10): Material lot ID	DINT	0	x
TAGN_Pn	Storage location found Pn (n = 02 to 10): TAG name	STRING[128]	''	x
UOM_Pn	Storage location found Pn (n = 02 to 10): Unit of the measured value	STRING[16]	''	x
FORM_Pn	Storage location found Pn (n = 02 to 10): Data format	DINT	0	x
IAM_Pn	Storage location found Pn (n = 02 to 10): Quantity in DINT format	DINT	0	x
ICON_Pn	Storage location found Pn (n = 02 to 10): Content in DINT format	DINT	0	x
IMIN_Pn	Storage location found Pn (n = 02 to 10): Minimum fill level in DINT format	DINT	0	x
IRAM_Pn	Storage location found Pn (n = 02 to 10): Remainder in DINT format	DINT	0	x
RAM_Pn	Storage location found Pn (n = 02 to 10): Quantity in REAL format	REAL	0.0	x
RCON_Pn	Storage location found Pn (n = 02 to 10): Content in REAL format	REAL	0.0	x
RMIN_Pn	Storage location found Pn (n = 02 to 10): Minimum fill level in REAL format	REAL	0.0	x
RRAM_Pn	Storage location found Pn (n = 02 to 10): Remainder in REAL format	REAL	0.0	x
EN_AUT	Operator control enable: Auto	BOOL	0	x
EN_MAN	Operator control enable: Manual	BOOL	0	x

Parameter	Description	Type	Default	OS tag
EN_SEAR	Operator control enable: Search	BOOL	0	x
EN_POS	Operator control enable: Selected row number	BOOL	0	x
DIS_MAAU	1 = Display of manual / auto buttons in the face-plate	BOOL	0	x
DIS_SEAR	1 = Display of search button in the faceplate	BOOL	0	x
QPOS	Selected row number	INT	0	x
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST		-
MSG_STAT	ALARM_8P: STATUS	WORD	0	-
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0	-
QMSG_SUP	ALARM_8P: 1 = Message suppression active	BOOL	0	-
QMSG_ERR	ALARM_8P: ERROR	BOOL	0	-

In/out parameters

The following table shows the in/out parameters of the "FB_STMAT" block:

Parameter	Description	Type	Default	OS tag
OPMANAUT	Operator: 0 = manual, 1 = auto	BOOL	0	x
OPSEARCH	Operator: 1 = Search	BOOL	0	x
OP_POS	Operator: Row number selection	INT	0	x
ROW_SEL	Selected rows (storage locations)	WORD	16#0	x
APF_AS	APF function block version in AS	WORD	0	x
APF_OS	APF function block version in OS	WORD	0	x

4.5.5 Error handling

Block error

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals.

The following error messages can be generated:

Request structure	Output	Meaning
	QPARAMF	Parameter assignment error; ID=0 or ARCH_NA=' ' or TIME_MON=0 or with activated filter (EN_FILT = 1): Filter ID < 0 or when use of remainder is activated (EN_REM = 1): SPLIT_PT < 1 / >9 or LFILTER2 < 0 or SP_MODE = 0
	QRLENERR	Data block length of the receive data is not correct

Request structure	Output	Meaning
	QRERR	Data error Header data in request and receive structure not the same
MON_ERR	QMONERR	Runtime error; Monitoring timeout
	QRM_ERR	Error from FB_UA_RM: Data error, job data in WinCC communication interface do not correspond to original job data
	QRM_AERR	Error from FB_UA_RM: Archive error; job data in the WinCC communication interface reset by the OS, or script error occurred in OS
	QRM_CERR	Error from FB_UA_RM: Communication error; Error message from "BRCV" coupling blocks
	QRM_PERR	Error from FB_UA_RM: Parameter assignment error; incorrect ID_1, R_ID_1 or ID_2, R_ID_2 or duplicate REQ_ID

4.5.6 Messaging

The "FB_STMAT" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QLEN_ERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_2	AS process control Message - fault	Read job monitoring timeout	$$BlockComment$$ QMONERR runtime error read job ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QERR data error read job ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_4	AS process control Message - error	Configuration error: ID=0 or ARCH_NA=' ' or TIME_MON=0 or with activated filter (EN_FILT=1) filter ID < 0, or when use of remainder is activated (EN_REM = 1) SPLIT_PT < 1 / >9 or LFILTER2 < 0 or SEL_MODE = 0	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@

Message	Message class	Event	Message text
SIG_5	AS process control Message - fault	QRUN_ERR: Error starting export	$$BlockComment$$ QRUN_ERR ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_6	AS process control Message - fault	QEND_ERR: Error completing export	$$BlockComment$$ QEND_ERR ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_7	AS process control Message - fault	QSEL_ERR: Error during search	$$BlockComment$$ QSEL_ERR ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@
SIG_8	Operating message – without acknowledgment	QACTIVE: Export active	$$BlockComment$$ QACTIVE ID @4%d@ REC_ID @5%d@ MAT @6%d@ quantity @8%d@ / @9%f@ @7%s@

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	@6%d@	Material
SD_7	@7%s@	UOM
SD_8	@8%d@	Quantity in DINT format
SD_9	@9%f@	Quantity in REAL format
SD_10	-	

4.6 User archives

4.6.1 FB_STORGROUP

The storage location groups are defined in this user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of the group	INTEGER 32
NAME	Name of the group	STRING 64
DESCR	Description of the storage location group	STRING 128

Element	Description	Format
LIC	Location identifier of the storage location group	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

4.6.2 FB_STORATTRIB

In this user archive, up to 50 storage location attributes (properties) are defined. The archive has the following structure:

Element	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Name of the attribute	STRING 64
UOM	Unit of measure for the attribute	STRING 16
FORMAT	Format of the attribute	INTEGER 32
ENUM_MAT	Name/coding of an enumeration	STRING 16
HL	High limit of operator control for numerical values	DOUBLE
LL	Low limit of operator control for numerical values	DOUBLE
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the output in the corresponding function block	STRING 20
VISIBLE	Display of the attribute in the process picture	INTEGER 32
OVERVIEW	Column position in the overview picture	INTEGER 32

Format coding

The following table shows the format coding in "INTEGER":

Integer value	Format
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places
5	REAL with 4 decimal places
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters

Integer value	Format
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
20xx	DATE
21xx	TIME
22xx	DATE AND TIME
30	BOOL
40	ENUMERATION
50	MATERIAL
51	MATERIAL (release check) X
52	MATERIAL LOT (release check) X
53	MATERIAL LOT CONTENTS FORMAT X
54	CONTENTS FORMAT (format-related value, cannot be modified)
60	32BITS (material releases) X
61	32BITS (material type releases) X
70	CONTENTS FORMAT (format-related value, can be modified)

The values marked X are not free for use. They are intended for the fixed attributes.

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

Example:

The integer value "2201" results in the following display in the screen:

"DD.MM.YYYY hh:mm:ss"

4.6.3 FB_STORVALUE

In this user archive, up to 50 storage location process values (actual values) are defined. The archive has the following structure:

Element	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Name of the attribute	STRING 64
UOM	Unit of the actual value	STRING 16
FORMAT	Format of the actual value	INTEGER 32
ENUM_MAT	Name/coding of an enumeration	STRING 16
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the output in the corresponding function block	STRING 20
VISIBLE	Display of the attribute in the process picture	INTEGER 32
OVERVIEW	Column position in the overview picture	INTEGER 32

Format coding

The following table shows the format coding in "INTEGER":

Integer value	Format
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places
5	REAL with 4 decimal places
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
20xx	DATE
21xx	TIME
22xx	DATE AND TIME
30	BOOL
40	ENUMERATION

Integer value	Format
50	MATERIAL
62	32BITS (static procedure enables from the Dairy Library) X
63	32 BITS (dynamic procedure enables from the Dairy Library) X
70	CONTENTS FORMAT (format-related numerical value)
71	STORAGE STATE (storage location status) X
72	MATERIAL STATE (material status) X

The values marked X are not free for use. They are intended for the fixed attributes.

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

4.6.4 FB_STORAGE

This user archive is where the storage locations are saved with their values for attributes and actual values. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of storage location	INTEGER 32 *
NAME	Name of storage location	STRING 64
DESCR	Description of storage location	STRING 128
LIC	Location identifier of material	STRING 16
STORGROUP	Storage location group assignment	INTEGER 32
VERSION	Current version	REAL
CHANGEUSER	Name of the operator who made the last change	STRING 32
CHANGEDATE	Date of the last change	DATE
REQ_TAG	Path of block	STRING 128
ASNO	Number of AS	INTEGER 32
DBNO	Instance data block number	INTEGER 32
FORMAT	Data format of content values	INTEGER 32

Element	Description	Format
ACC_MODE	Booking mode of storage location	INTEGER 32
MATID	ID of base material	INTEGER 32
MLOTID1 …. MLOTID10	ID of material lot (1 to 10)	INTEGER 32
MLOT_CONT1 …. MLOT_CONT10	Quantity of material lot in storage location (1 to 10), each by MLOTIDn	DOUBLE
CONT_MAX	Maximum content of storage location	DOUBLE
CONT_MIN	Minimum content of storage location	DOUBLE
STOR_CONT	Content of storage location This values is calculated on the basis of the booking and clearing processes.	DOUBLE
STOR_USE_IN	Planned storage quantity at the storage location This value is calculated on the basis of one or more filling processes and is used by the "FB_STLOC" function block	DOUBLE
STOR_USE_OUT	Planned quantity to be removed from the storage location This value is calculated on the basis of one or more removal processes and is used by the "FB_STMAT" function block.	DOUBLE
UOM	Unit of storage location content	STRING 16
MATCLASS_REL1 …. MATCLAS_REL4	Material class release (1 to 4)	INTEGER 32
MATTYPE_REL1 …. MATTYPE_REL4	Material type release (1 to 4)	INTEGER 32
ATTRIB_<ioname> .... ATTRIB_<ioname>	Value of property (1 to 50) Maximum of 50 cells	Corresponds to formats of the attributes from the corresponding user archive
ACT_CONT_A	Content of storage location (actual value)	DOUBLE
ACT_CONT_P	Content of storage location % (actual value)	REAL
ACT_STOR_STATE	Status of storage location	INTEGER 32
ACT_MAT_STATE	Status of material	INTEGER 32
ACT_REL1STAT …. ACT_REL4STAT	Status of static procedure enables (1 to 4)	INTEGER 32

Element	Description	Format
ACT_REL1DYN …. ACT_REL4DYN	Status of dynamic procedure enables (1 to 4)	INTEGER 32
VALUE_<ioname> …. VALUE_<ioname>	Actual value (1 to 50) Maximum of 50 cells	Corresponds to formats of the actual values from the corresponding user archive

Note

*The manual input in the editor dialog is limited to 2,000,000,000.

The attributes are assigned the column name "ATTRIB_" with the "FCT_BLOCK_IO_NAME" of the FB_STORATTRIB table.

Note

Forming the attribute column name (a maximum of 20 characters) from "ATTRIB_" and "<ioname>" limits the actual useful length of the IO name to 13 characters.

The actual values are assigned the column name "VALUE_" with the "FCT_BLOCK_IO_NAME" of the FB_STORVALUE table.

Note

Forming the actual value column name (a maximum of 20 characters) from "VALUE_" and "<ioname>" limits the actual useful length of the IO name to 14 characters.

ACC_MODE

The ACC_MODE (accounting mode) column defines the accounting mode for booking material lots. This mode is required by the "FB_ST_AC" block and in the FB_MAT_IN.xxx and FB_MAT_ADJ.xxx faceplates.

The following table shows the mode coding in "INTEGER":

Integer value	Mode
1	A material lot can be booked if the storage location is empty (STOR_CONT <= 0). In this case, you may not mix multiple material lots.
2	A new material lot can be booked if all other material lots previously booked are identical. In this case, you may not mix different material lots.
3	Different material lots may be booked. In this case, you may mix different material lots.

FORMAT

The FORMAT column specifies the numerical format for the CONT_MAX, CONT_MIN, STOR_CONT, STOR_USE_IN, STOR_USE_OUT and ACT_CONT_A columns. This is required for display in the OS and for the corresponding AS block.

The following table shows the format coding in "INTEGER":

Integer value	Mode
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places
5	REAL with 4 decimal places

Procedure enables and material releases

The texts for the procedure enables can be found n the "FB_PRC_REL" archive (the name and content can be configured in the engineering tool).

A maximum of 128 classes and 128 types can be written with the material class and material type releases. This means that only the first 128 classes/types can ever be addressed with these releases. It is therefore important that you create the classes / types in the correct order.

Status texts

Texts for the status of the storage location and the material are managed using the FB_ST_STOR_STATUS and FB_ST_MAT_STATUS archives (the name and content can be configured in the engineering tool).

4.6.5 FB_STOR_ST_STATUS

A range of different storage states are saved in this user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of storage state	INTEGER 32
NAME	Name of storage state	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

4.6.6 FB_STOR_MAT_STATUS

A range of different material states are saved in this user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of material location status	INTEGER 32
NAME	Name of material state	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

4.6.7 FB_PROCEDURES

Up to 128 procedures are saved in this user archive. The archive has the following structure:

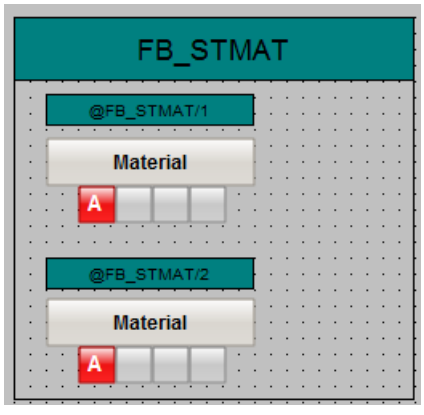
Element	Description	Format
REC_ID	Internal ID of procedure (corresponds to the bit number in the 4 double words / 32-bit integer values)	INTEGER 32
NAME	Name of procedure	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

4.7 Operator control and monitoring

4.7.1 Block icons

@TypicalsAPF.pdl

The "@TypicalsAPF.pdl" template picture contains templates for automatically generating block icons from the plant hierarchy:

Block	Templates
FB_STLOC	
FB_STMAT	

@Template_APF.pdl

The "@Template_APF.pdl" template picture contains templates for block icons and for calling additional functions for use in the process pictures.

4.7.2 Picture objects

The interface for storage location management operator control and monitoring consists of the following picture objects:

WinCC flow charts

- FB_STOR_ED.pdl
  The picture allows you to create, change, delete and copy storage location master data and to carry out updates with AS data.
- @FB_STOR_ED_WINDOW.pdl
  You can use the picture to edit the attribute and actual value parameters of the storage locations. This picture is implemented as a window in the "FB_STOR_ED" picture and cannot be used separately.
- FB_STOR.pdl
  The picture gives a freely configurable overview of all storage locations.

Faceplates with FB assignment

- @PG_FB_STLOC.PDL
  The faceplate enables you to search possible storage locations for material so that you can store it.
- @PG_FB_STMAT.PDL
  The faceplate enables you to search for material in storage locations so that you can export it.

Faceplates without FB assignment

- @PG_FB_MAT_ADJ.pdl
  This faceplate enables manual adjustment booking of material lots (including partial quantities) to storage locations.
- @PG_FB_MAT_IN.pdl
  This faceplate enables manual booking of materials and material lots (including partial quantities) to storage locations.
- @PG_FB_MAT_OUT.pdl
  This faceplate enables manual clearing of materials and material lots (including partial quantities) from storage locations.
- @PG_FB_MAT_IO.pdl
  This faceplate enables manual clearing and booking of materials and material lots (including partial quantities) from one storage location to another. The procedure is implemented in one step.
- @PG_FB_MAT_CONV.pdl
  This faceplate enables the manual conversion of materials and material lots in a storage location.

- @PG_FB_ST_SEARCH.pdl
  This faceplate enables you to search for storage locations corresponding to the selected search criteria.
- @PG_FB_MAT_SEARCH.pdl
  This faceplate enables you to search for materials / material lots in the storage locations that correspond to the selected search criteria.

Pictures for display in a picture window (cannot be used separately)

- @FB_Account.PDL
  Two-step operation for booking materials/material lots to storage locations.
- @FB_ShowMatLots.PDL
  Display of material lots during the material search.
- @FB_32BitsRel.PDL
  Display of procedure enables in FB_STOR.pdl.

Note

The process pictures are pre-configured and do not need to be edited.

For detailed information on interface operator control and monitoring in process mode, see the APF Operator Manual.

4.7.3 Messages

The following messages can be output:

Message class	Event	Message text
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ / storage location: @6%s@
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: @2%s@: new= @5%s@ @8%s@ old = @4%s@ @8%s@ / storage location: @6%s@
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: storage location @6%s@ changed
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: storage location @6%s@ created
Operator message	FB_STOR_ED.pdl: Delete button	@7%s@: storage location @6%s@ deleted
OS process control error	FB_STOR_ED.pdl: general OS scripts	Storage location name: @6%d@ error code: @5%d@
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: @2%s@: release @5%s@ set / storage location: @6%s@
Operator message	FB_STOR_ED.pdl: Save button	@7%s@: @2%s@: release @5%s@ removed / storage location: @6%s@
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ ADJ/IO.pdl	@7%s@: Booking not possible! Storage location is not empty!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ ADJ/IO.pdl	@7%s@: You cannot book different material lots!

Message class	Event	Message text
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: Maximum number of material lots is reached!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/ADJ/IO.pdl	@7%s@: Maximum content limit reached!
OS process control error	FB_ST_AC, @PG_FB_MAT_OUT/ADJ/IO.pdl	@7%s@: Minimum content limit reached!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: Material lot ID: @5%d@ does not exist!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: wrong material ID: @5%d@!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: Storage location ID: @5%d@ does not exist!
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: DINT limit reached!
Operator message	FB_ST_AC, @PG_FB_MAT_IN/IO.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) @4%.4f@ @8%s@ booked
Operator message	FB_ST_AC, @PG_FB_MAT_OUT/IO.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) @4%.4f@ @8%s@ cleared
OS process control error	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO/CONV.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) @4%.4f@ @8%s@ booking failed
Operator message	FB_ST_AC, @PG_FB_MAT_OUT/IO.pdl	@7%s@: material @9%s@ (ID: @5%d@) deleted from storage location
Operator message	@PG_FB_MAT_ADJ.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) correction booking successful
Operator message	FB_ST_AC, @PG_FB_MAT_ADJ.pdl	@7%s@: material lot @9%s@ quantity: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ / storage location: @6%s@
Operator message	FB_ST_AC, @PG_FB_MAT_IN/OUT/ADJ/IO.pdl	@7%s@: @3%d@. Material lot: new = @5%s@ alt = @4%s@ / storage location: @6%s@
Operator message	FB_ST_AC, @PG_FB_MAT_OUT/IO.pdl	@7%s@: Material ID: @5%d@ deleted!
Operator message	FB_ST_AC, @PG_FB_MAT_CONV.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) @4%.4f@ @8%s@ converted
Operator message	FB_ST_AC, @PG_FB_MAT_CONV.pdl	@7%s@: material lot @9%s@ (ID: @5%d@) @4%.4f@ @8%s@ cleared for conversion
OS process control error	FB_ST_AC, @PG_FB_MAT_CONV.pdl	@7%s@: quantity (@4%.4f@) could not be practically converted to unit (@8%s@)!
OS process control error	FB_ST_AC	@7%s@: Source storage location is empty!
Operator message	@PG_FB_STLOC/STMAT.pdl	@7%s@: Switchover to manual mode
Operator message	@PG_FB_STLOC/STMAT.pdl	@7%s@: Switchover to automatic mode
OS process control error	FB_ST_AC	@7%s@: Unit could not be converted!
OS process control error	FB_ST_AC	@7%s@: different content formats!
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
OS process control error	Cyclic backup action: The export file could not be created.	@7%s@: Backup file (@2%s@) could not be exported

Message class	Event	Message text
OS process control error	Cyclic backup action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the backup file
OS process control error	Cyclic backup action: Directory 1 could not be created	@7%s@: Backup directory 1 does not exist / could not be created
OS process control error	Cyclic backup action: Directory 2 could not be created	@7%s@: Backup directory 2 does not exist / could not be created
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 1 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 2 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 1 is corrupt
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 2 is corrupt
OS process control error	Cyclic backup action: Call the external backup program failed	@7%s@: Initialization of the backup program failed
OS process control error	Incorrect user-specific filter, OS scripts with search functions	@7%s@: User-defined filter (probably) failed. Check filter @4%d@!
OS process control error	Incorrect user-specific sorting, OS scripts with search functions	@7%s@: User-defined sorting failed. Check filter @4%d@!
OS process control error	FB_ST_AC -> L_MAT_DEL input	@7%s@: Material ID: @5%d@ could not be deleted. Storage location not empty!

Message associated values

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%s@	Batch name
Message associated value 2	@2%s@	Storage location attribute / backup file
Message associated value 3	@3%d@	Batch ID OR partial record ID
Message associated value 4	@4%s/d@	Old value (string or number) OR storage location ID
Message associated value 5	@5%s/d@	New value (string or number) OR material ID OR material lot ID OR storage location ID OR release
Message associated value 6	@6%s@	Storage location name + ID, here: Storage location name (ID)
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	Unit / UOM
Message associated value 9	@9%s@	Material lot
Message associated value 10	@10%s@	Source, here: Storage location management

For additional information, refer to the PCS 7 online help.

4.8 Other

4.8.1 User-defined filter/sorting

4.8.1.1 Basic procedure

Note

You need basic knowledge of T-SQL and the C programming language to create the user-defined filter / sorting.

The following function blocks / faceplates support user-defined searches:

- FB_STLOC + corresponding faceplate: Storage location search
- FB_STMAT + corresponding faceplate: Material and material lot search
- @PG_FB_ST_SEARCH faceplate: Storage location search
- @PG_FB_MAT_SEARCH faceplate: Material and material lot search

During the engineering phase, you can define a range of user-defined SQL filters / sorting which then affect the display and processing sequence of the sought-after storage locations.

The filter and sorting are entered directly as SQL syntax. The operator or infeed from the automation system can include variable values during runtime. Six values are provided for this. They can be entered directly in the SQL filter with the following placeholders:

Name	Format	Placeholder
Value 1	Integer	@1@
Value 2	Integer	@2@
Value 3	Real / float	@3@
Value 4	Real / float	@4@
Value 5	Date	@5@
Value 6	Date	@6@

A search filter might appear as follows:

(REC_ID > @1@) AND (REC_ID < @2@)

This would mean that only storage locations with a storage location number (REC_ID) between *Value 1* and *Value 2* are displayed. The two values are specified at the block inputs (VALUE1 and VALUE2) for FB_STLOC / FB_STMAT and by the operator for the faceplates @PG_FB_ST_SEARCH / @PG_FB_MAT_SEARCH.

In the C project function FB_GetFilterAndSortValues.fct in the APF\FB_ReadAndWrite_AS_OS \FB_UA_USER_SPECIFIC folder, various cases can be created and configured (implemented via a C switch/case instruction) that can later be selected and used in runtime by the operator or by the automation system.

The following information must be saved in the C function for each case defined:

iFilterAndSortNameTextLibID = Text library ID for the name of the filter and / or sorting

lpszValue1UOM = Unit of the first free value (integer)
lpszValue2UOM = Unit of the second free value (integer)
lpszValue3UOM = Unit of the third free value (real / float)
lpszValue4UOM = Unit of the fourth free value (real / float)

iValueName1TextLibID = Text library ID of the first free value (integer)
iValueName2TextLibID = Text library ID of the second free value (integer)
iValueName3TextLibID = Text library ID of the third free value (real / float)
iValueName4TextLibID = Text library ID of the fourth free value (real / float)
iValueName5TextLibID = Text library ID of the fifth free value (date)
iValueName6TextLibID = Text library ID of the sixth free value (date)

lpszValueName1 = Name of the first free value (integer)
lpszValueName2 = Name of the second free value (integer)
lpszValueName3 = Name of the third free value (real / float)
lpszValueName4 = Name of the fourth free value (real / float)
lpszValueName5 = Name of the fifth free value (date)
lpszValueName6 = Name of the sixth free value (date)

iValue1LL = Low input limit for the first free value in the faceplate (integer)
iValue2LL = Low input limit for the second free value in the faceplate (integer)
iValue3LL = Low input limit for the third free value in the faceplate (real / float)
iValue4LL = Low input limit for the fourth free value in the faceplate (real / float)

iValue1HL = High input limit for the first free value in the faceplate (integer)
iValue2HL = High input limit for the second free value in the faceplate (integer)
iValue3HL = High input limit for the third free value in the faceplate (real / float)
iValue4HL = High input limit for the fourth free value in the faceplate (real / float)

bValue1Used = First free value used in search filter? (integer)
bValue2Used = Second free value used in search filter? (integer)
bValue3Used = Third free value used in search filter? (integer)
bValue4Used = Fourth free value used in search filter? (integer)
bValue5Used = Fifth free value used in search filter? (integer)
bValue6Used = Sixth free value used in search filter? (integer)

lpszSQLFilter = Search filter in SQL syntax
lpszSQLSort = Sorting for filtered search in SQL syntax

lpszSQLFilter and lpszSQLSort can be used simultaneously but do not have to be. You can, for example, define sorting only for a given search.

If lpszSQLFilter is left blank and only lpszSQLSort is configured, the search results are simply sorted.

The six different free values can be enabled and disabled with bValue1Used .. bValue6Used. These settings define whether the corresponding value is displayed and therefore whether the corresponding free value can be entered in the faceplate. It makes little sense to allow the operator to view or specify free value 3 if this value is not used in the filter.

4.8.1.2 Configuration example C function

The example below is for case 0:

```
switch (iSelection)
{
   case 0:
      *iFilterAndSortNameTextLibID = 1234; // text library id of the filter name
      strncpy(lpszValue1UOM, "", iValue1UOMLength-1); //uom of value 1
      strncpy(lpszValue2UOM, "", iValue2UOMLength-1); //uom of value 2
      strncpy(lpszValue3UOM, "kg", iValue3UOMLength-1); //uom of value 3
      strncpy(lpszValue4UOM, "kg", iValue4UOMLength-1); //uom of value 4
      *iValueName1TextLibID = 1235; //text library id of value 1 (int)
      *iValueName2TextLibID = 1236; //text library id of value 2 (int)
      *iValueName3TextLibID = 1237; //text library id of value 3 (real)
      *iValueName4TextLibID = 1238; //text library id of value 4 (real)
      *iValueName5TextLibID = 1239; //text library id of value 5 (date and time)
      *iValueName6TextLibID = 1240; //text library id of value 6 (date and time)
      *iValue1LL = -1; //low limit of value 1 (int)
      *iValue2LL = -1; //low limit of value 2 (int)
      *dValue3LL = 0.0; //low limit of value 3 (real)
      *dValue4LL = 0.0; //low limit of value 4 (real)
      *iValue1HL = 50; //high limit of value 1 (int)
      *iValue2HL = 50; //high limit of value 2 (int)
      *dValue3HL = 10000.0; //high limit of value 3 (real)
      *dValue4HL = 10000.0; //high limit of value 4 (real)
      *bValue1Used = TRUE; //value 1 used? (int)
      *bValue2Used = TRUE; //value 2 used? (int)
      *bValue3Used = TRUE; //value 3 used?(real)
      *bValue4Used = TRUE; //value 4 used?(real)
      *bValue5Used = TRUE; //value 4 used? (date and time)
      *bValue6Used = FALSE; //value 4 used? (date and time)
```

```
        strncpy(lpszSQLFilter, "(REC_ID >= @1@) AND (REC_ID <= @2@) AND
        ((STOR_CONT > @3@) AND (STOR_CONT < @4@)) AND (CHANGEDATE = @5@)",
        iSQLFilterLength-1); //filter string (length <= 1024)
        strncpy(lpszSQLSort, "STOR_CONT DESC, REC_ID ASC", iSQLSortLength-1); //sort
        string (length <= 1024)
        break;
}
```

The areas highlighted in **bold** must be specified during the engineering phase.

In the example, the search is for storage locations which have a storage location ID between the two values *Value 1* and *Value 2*, a minimum content of *Value 3* and a maximum content of *Value 4*, and which were last changed with the editor on *Value 5*.

These filtered storage locations are then sorted by content in descending order. If there are multiple storage locations with the same content, they are sorted by storage location ID in ascending order.

Important points when setting an SQL filter:

- Adhere to the syntax and subsequently check the filter / expected result
- If the values 5 and 6 (date) are used, they cannot be put in ' ' in the SQL filter.
- If no filter is specified, the C code must still be applied and not simply deleted (strncpy(lpszSQLSort, "", iSQLSortLength-1);).
- The empty sample case in the script must be copied and added under a new number ("case x:") for each new case .

4.8.1.3 Default values

AS function blocks

The following block inputs are available at the two AS function blocks:

Name	Format	Block input
Value 1	Integer	VALUE1
Value 2	Integer	VALUE2
Value 3	Real / float	VALUE3
Value 4	Real / float	VALUE4
Value 5	Date	VALUE5 + VALUE5B
Value 6	Date	VALUE6 + VALUE6B

The two date values are divided into two separate double words (as in other APF modules). These can be connected to the FB_BBD function block, for example. For more details, refer to the general APF description.

Faceplates

Faceplates display values, value names and value units in a popup window. If all six values are used, they are listed in row 1 to row 6.

The popup window in the faceplates for the FB_STLOC and FB_STMAT blocks is used simply to display the filters and values selected in the AS. The values can only be changed or specified in the @PG_FB_ST_SEARCH and @PG_FB_MAT_SEARCH faceplates.

4.8.1.4 Restrictions

If a quantity is used with a dynamic filter, direct conversion on the basis of the unit is no longer possible. This is because an SQL instruction is entered with a user-defined filter. For example, if you search for a quantity > 1000 kg (SQL syntax: "STOR_CONT > 1000"), the STOR_CONT column for all storage locations are checked for content greater than 1000 regardless of whether they use kilograms or tons.

The solution in this case is to adjust the filter. The following custom filter would produce the correct result: "((STOR_CONT > 1000) AND (UOM = 'kg')) OR ((STOR_CONT > 1) AND (UOM = 't'))"

The standard criteria for a search are not affected, which means the quantity is still converted and checked (LVAL_I or LVAL_R with FB_STLOC and FB_STMAT blocks).

4.8.2 Two-step operation

Using a binary input ("SEL_MODE") at the AS function blocks FB_STLOC and FB_STMAT, you can specify whether a click of the mouse applies storage location(s) directly after selection, or whether you need to confirm your selection using an additional button in the faceplate. If the bit is at SEL_MODE, two-step operation is active.

In remainder mode (applies to AS function block FB_STMAT only), two-step operation is always active.

4.8.3 Use of remainder

If you want to use up small quantities remaining in the storage location at the FB_STMAT AS function block, this functionality can be activated at the "EN_REM" binary input. In this mode, the block changes its characteristics in the following areas:

- Multiple storage locations can be selected
- Two separate filters are required
  - Remainder filter input "LFILTER"
  - Normal filter input "LFILTER2"
  - The two filters share the six free filter parameters ("LVALUE1" .. "LVALUE6B" inputs)
- "Split" point input "SPLIT_PT" is queried

- "L_NEXT" input switches selected storage locations one after the other to the active data range "QS_ID" .. "QRRAM"
- Only two-step operation is possible, regardless of the setting at the "SEL_MODE" input

4.8.3.1 "Split" point

To solve the problem of not achieving the required setpoint with the storage locations / quantities displayed with the set filter, there are two separate filters (LFILTER and LFILTER2 inputs) and a "split" point ("SPLIT_PT" input). The search returns a maximum of 10 storage locations for selection. The "Split" point determines how many of the 10 storage locations are searched for using filter 1 and how many with filter 2. The "Split" point of 7 means that a maximum of 7 storage locations are searched for using filter 1 and the remaining storage locations are searched for using filter 2.

Filter 1 is the "remainder" filter in which the setpoint query is deactivated. This means it is possible to find and display setpoints which do not contain enough material. The remaining storage locations searched for with filter 2 must meet the setpoint <= content criterion. This ensures that the required overall setpoint can always be achieved despite use of remainder.

Example:

If the "split" point = 5 and provided there are enough storage locations matching the set filters, the first 5 storage locations with remainders are returned and can, if required, all be emptied. The last 5 storage locations displayed each contain enough material to fulfill the job with the configured setpoint.

4.8.3.2 Selection of multiple storage locations

Manual mode:

Up to 10 storage locations can be selected in turn with a click of the mouse. Clicking again on a selected storage location row deselects it. If the setpoint (required quantity) is reached, the "Select" button is enabled and can be used to select the marked storage locations for the job.

The sum of the individual quantities selected is displayed in the faceplate. This clearly indicates how large a quantity has already been selected with the remainders and how much is still required to make up the setpoint.

Auto mode:

In auto mode, only the filters / sorting set ("LFILTER" and "LFILTER2" inputs) are relevant for storage location selection. The system adds storage locations (in the faceplate: "from top to bottom") until the required setpoint (required quantity) is reached.

A maximum of 10 storage locations can be displayed / selected even if more are found with the filters. If the total from all 10 possible storage locations does not reach the setpoint, the job cannot be started. Filters / sort criteria in the configuration must ensure that this scenario does not occur during the process.

4.8.3.3 Output of selected storage locations

The FB_STMAT function block has output areas for the selected storage locations (S_ID_SEL .. RRAM_SEL to S_IDSL10 .. RRAMSL10) and an output area for the active storage location (QS_ID .. QRRAM).

4.8.3.4 Switching the active storage location

A positive edge at the binary input "L_NEXT" "switches" to the next selected storage location. This means that the output area of the first selected storage location is automatically copied to the output area of the active storage location (QS_ID .. QRRAM) following "L_START" . A positive edge at the "L_NEXT" input copies the next selected storage location and its data to the output area of the active storage location. This continues until all selected storage locations have been processed. The process is then terminated by the "L_END" input. If the last possible storage location is active, any subsequent positive edge at the "L_NEXT" input is not evaluated. Explicit termination is only possible with the "L_END" input.

4.8.3.5 Reserving material quantities

When the job is started, the partial quantities of the setpoint are reserved in the STOR_USE_OUT column in the storage location user archive (default "FB_STORAGE") and are used in parallel queries. This is done for all selected storage locations after a positive edge at the "L_START" input. All partial quantities are therefore already reserved for all selected storage locations at this stage, instead of being reserved individually upon storage location change. The reservation for the storage location processed is deleted upon a positive edge at the "L_NEXT" input or upon "L_END" for the last storage location.

4.8.3.6 Plausibility check for storage locations

The plausibility check on the basis of the quantities and the selected filter criteria for the storage locations sought is performed during the search and once for all selected storage locations upon a positive edge at the "L_START" input. Once this final check is successful, no further checks are executed upon a storage location change with the "L_NEXT" block input.

4.8.3.7 Search with use of remainder

L_SEARCH

- Manual mode:
  In manual mode, the 10 possible storage locations are displayed in the faceplate following a positive edge at the L_SEARCH input. After selection and additional confirmation, the up to 10 storage locations selected are output at the 10 corresponding output areas.
- Auto mode:
  In automatic mode, up to 10 storage locations are selected in turn in line with the set sort criteria following a positive edge at the L_SEARCH input and output to the 10 corresponding output areas. The system continues to select new storage locations until the setpoint is reached.

L_START

Upon a positive edge at the input L_START, all selected storage locations are re-examined and those with a positive result are reserved in the respective quantities in the STOR_USE_OUT column of the user archive.

The first storage location in the list (the first selected in the faceplate) is also copied to the "Active storage location" output area.

L_NEXT

Following a positive edge at the L_NEXT input, the system switches to the next storage location in the sequence. The data for this subsequent storage location are then output at the output area for the active job. The quantity previously reserved for the storage location processed is released again (subtracted) in the "STOR_USE_OUT" column of the user archive.

L_END

After completion of the transfer / job, the quantity previously reserved for the storage location processed is released again (subtracted) in the "STOR_USE_OUT" column of the user archive with a positive edge at the L_END input.

Note

A new check is not performed upon storage location change. Storage locations are also reserved at the start of the process and not when it is "their turn".

4.8.4 Material class / material type releases

There are several ways to lock or release a storage location for certain types of materials for the storage location management. First, you can make a storage location release for a specific material depending on the material class. Second, you can define releases on the basis of material types. These two options can be used separately or together. Four 32-bit cells are available in the user archive (MATCLASS_RELx and MATTYPE_RELx columns, x = 1..4) for each option. Up to 128 (4*32) material classes or material types can be released or locked in these cells, namely the first 128 material classes or the first 128 material types of material class 1.

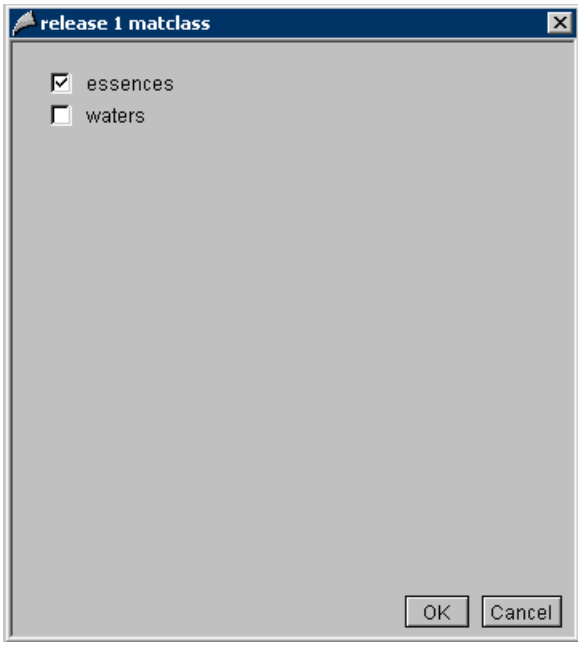
In the engineering tool, you can set whether the releases are to be used (see the engineering tool help for details). You can enable one of the two (class or type) or even both. Using both allows you to define storage locations for release/lock more precisely. For example, you can release a storage location for materials of class 1 AND type 3 only.

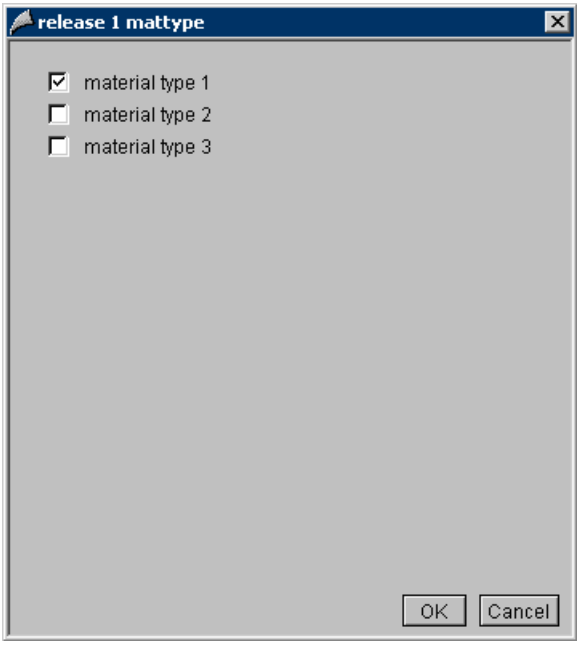
The release cells correspond to bit-coded 32-bit integer values (DINT). Negative values in the user archive are therefore perfectly possible. For example, -1 stands for "all 32 bits set".

An object which displays 32 releases is available for display in the editor screen FB_STOR_ED.pdl and the overview picture FB_STOR.pdl. In the figure below, the first four material classes or material types are released and the remaining 28 for this storage location are locked.



Details of the releases can be accessed by clicking on the object. The picture window that then opens shows the material classes / material types for the object and their releases. In the two following pictures, only two materials classes / three material types are available. If there are more in the archives, they are automatically displayed in these pictures. Releases can either be modified or read only depending on the call environment.





4.8.5 Static and dynamic procedure enables

In other libraries, so-called procedure enables are used to block or release certain user-defined operations on tanks. To achieve compatibility, these enables are also available in the storage location management of the APF Toolbox. They function in a similar way to material class / material type releases. If they are used, they can be enabled in the engineering tool (see the engineering tool help for details).

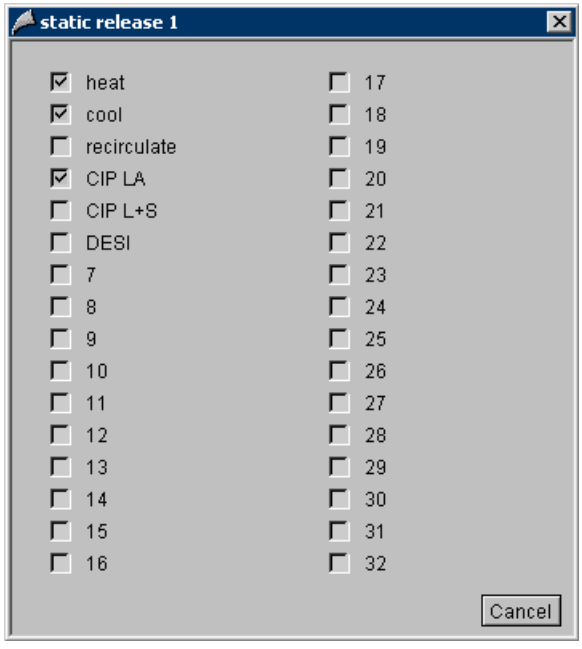
The enables are external and are therefore implemented as actual values appearing on the right side of the editor screen FB_STOR_ED.pdl. There are also 128 different user-defined releases (enables), split into four 32-bit cells (ACT_RELxSTAT and ACT_RELxDYN columns, x = 1..4). These releases are, however, split into static and dynamic releases. Dynamic releases are set and removed by the process, static releases by the operator. Releases are only accepted if both are set. For example, when these releases are used, the search function queries both (static and dynamic). The storage location only meets the criteria and is displayed if both releases are met.

The release cells correspond to bit-coded 32-bit integer values (DINT). Negative values in the user archive are therefore perfectly possible. For example, -1 stands for "all 32 bits set".

Example:

The second bit is set in the ACT_REL1_STAT cell. For this release to take effect, the second bit in the ACT_REL1DYN cell must also be set.

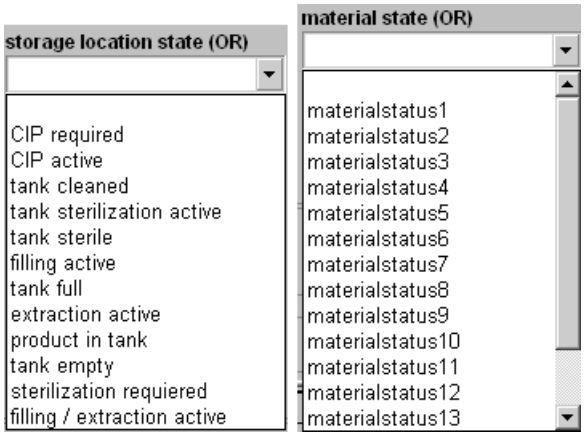
Static releases 1, which means archive column ACT_REL1STAT, are displayed in the following picture window. Bits 0, 1 and 3 are set in this window. The corresponding integer value would therefore be 11.



4.8.6 Storage / material state

A storage location can be assigned a range of states as integer values with storage / material state. There is an assigned text for this integer value in the FB_STOR_ST_STATUS and FB_STOR_MAT_STATUS archives. It is important to number the states in the archive consecutively starting with ID 1. The various states can usually be selected as a pull-down menu in the OS area and as a number in the AS area.

Storage / material state can be enabled and disabled separately using the engineering tool. Refer to the engineering tool help for more detailed information. The texts displayed below are read from the text library on the basis of the text library ID in the FB_STOR_ST_STATUS and FB_STOR_MAT_STATUS archives.



Parameter management

5

5.1 FB_XXXPA

5.1.1 Description

Object properties

The following table shows the object properties of FB_XXXPA:

Name	Number	Family	Module
FB_XXXPA	Assigning block numbers with the engineering tool.	APF	Parameter management

Area of application

The block is used to request and provide parameter set data in the AS.

Startup characteristics

The "FB_XXXPA" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

Corresponding scripts for the function block and its functionalities on the AS level are available for handling on the OS level.

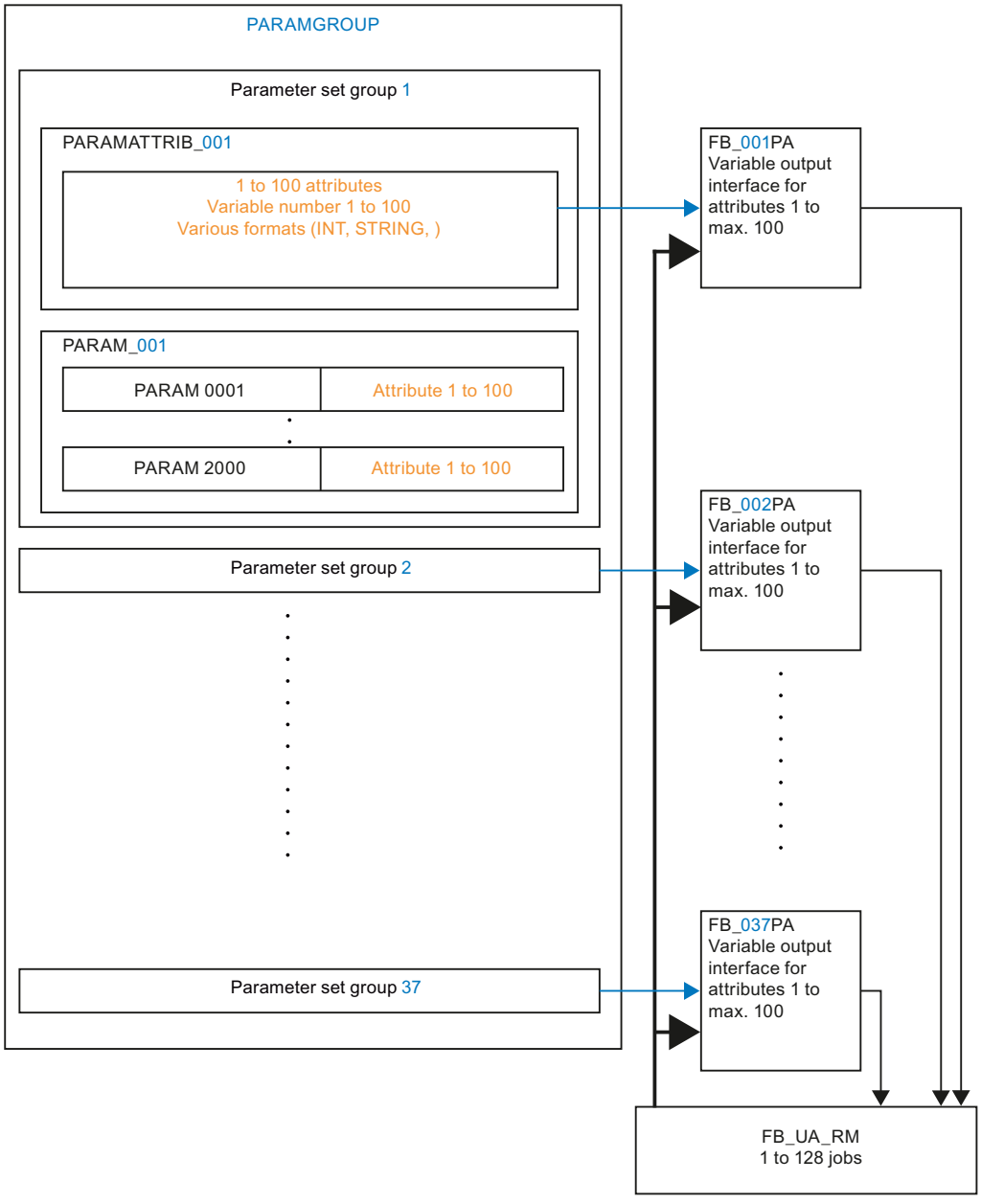
The "ALARM_8P" structure is used to implement data communication with the OS message system.

5.1.2 Generation

For each parameter group created using the engineering tool in the "FB_PARAMGROUP" user archive, a block of the type "FB_XXXPA" is generated and saved in the master data library. The generated blocks are numbered consecutively.

The block input and output parameters depend on the structure of the "FB_PARAMATTRIB_XXX" user archive in which the properties of a parameter group are defined. The corresponding block needs to be generated again if the attributes are changed.

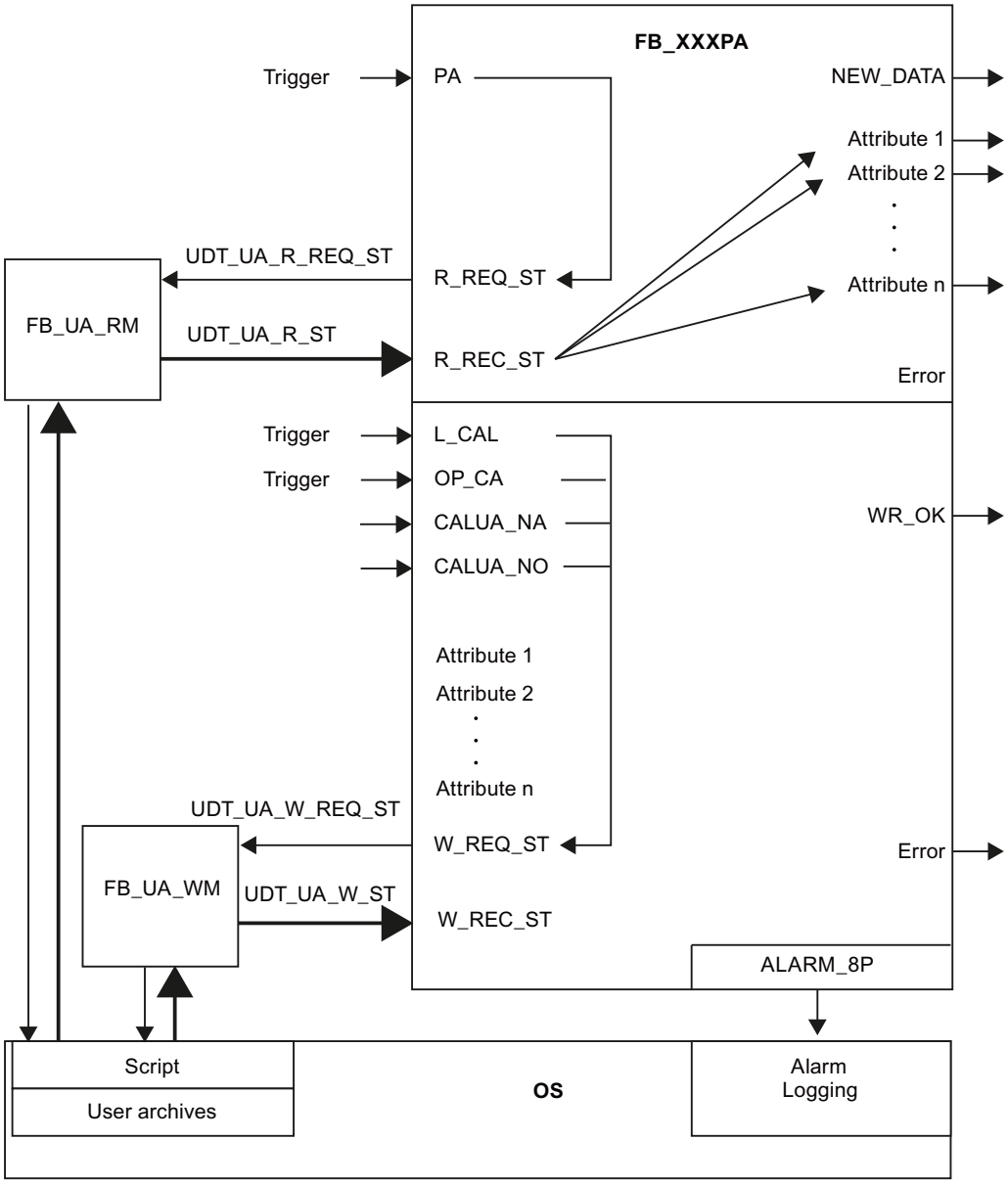
The figure below gives an overview of the structure of "FB_XXXPA" blocks:



5.1.3 Functions

Overview

The figure below gives an overview of how the "FB_XXXPA" block works:



Mode

The "FB_XXXPA" block can operate in normal or in multiplexer mode. The mode is set at the "MODE" input parameter.

- In normal mode, the block performs the following functions:
- Reading a parameter data record from the parameter set archive and providing the data at the outputs
- Recalculating the parameter values on request:
  - Writing the parameter data record to the calculation archive
  - Recalculating
  - Reading the newly calculated parameter data record back from the calculation archive and providing the data at the outputs

In multiplexer mode, the block performs the following functions:

- Providing parameter set data for the activated job:
  - Copying parameter set data from the corresponding parameter set block when a job is activated
  - Providing the data at the outputs
  - Resetting the parameter set block (making it available for creating other jobs)

The following table shows the configuration options for the "MODE" input parameter:

MODE	Meaning
Bit 0	0 = Normal mode, 1 = Multiplexer mode
Bit 1	Upon data request in normal mode: Data record at block output 0 = Do not delete 1 = Delete
Bit 2	Upon data request in multiplexer mode: 0 = No "RESET" at data source; 1 = "RESET" at data source;
Bit 3	In multiplexer mode: 0 = No release for recalculation; 1 = Release for recalculation;

Normalization/recalculation

You can use the APF Engineering Tool to specify whether normalization/recalculation is to be performed for each attribute of a parameter group. This is done using the "Norm" value:

- "Norm" = 0: No normalization/recalculation
- "Norm" = 1: Normalization/recalculation is performed according to the equations of the following functions:
  - FB_CalcValueByNormPM.fct
  - FB_ReCalcValueByNormPM.fct
- "Norm" > 1: The normalization/recalculation is performed according to user-defined equations. The "FB_CalcValueByNormPM.fct" and "FB_ReCalcValueByNormPM.fct" functions must be extended appropriately for this. The equations for "Norm" = 0 / 1 must not be changed.

The properties specified using the APF Engineering Tool are saved in the "FB_PARAMATTRIB_XXX" user archive. The structure of this archive determines the input and output parameters of the corresponding "FB_XXXPA" block.

The parameter set values to be normalized and recalculated and the reference quantity are defined with the process picture and saved in the "FB_PARAM_XXX" (NORMAMOUNT)" user archive. If you set the reference quantity to "0", normalization is not executed.

On request, the "FB_XXXPA" block reads the parameter data from the user archives and triggers normalization/recalculation of the requested data. Data normalization/recalculation is performed using the "FB_PARAM_CALC_XXX" archive and the relevant functions.

Requesting data

Following a request by a positive edge at the request input "PA.REC_REQ", the data of a data record written via input REC_ID (parameter set ID) and ARCH_NA /ARCH_NO are read and written to the output structure "UDT_UA_R_REQ_ST". The read job is activated by setting the "REC_REQ" bit. Depending on the reference quantity and normalization bit of the individual parameters, normalization may also be triggered and return the normalized data at the output.

On the OS, the individual parameters are recalculated using the C function "FB_CalcValueByNormPM.fct". The changed values are supplied to the "FB_UA_RM" block.

Once the read job has been processed, the communication block "FB_UA_RM" writes the request data (including REQ_ID) and the pointer to the parameter data record to the "UDT_UA_R_ST" input structure of the block, where it also sets the "NEW_DATA" input bit.

The input data are only assigned to the block's read job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" in the input structure.

Once the input structure data have been assigned to the read job, the data are checked for validity and plausibility. The system evaluates the following:

- Whether the request data in the input and output structure match
- Whether the data length calculated corresponds to the data length of the outputs at the block

If the data are plausible and valid, the "NEW_DATA" output is set and the data are written from the "FB_UA_RM" communication block to the outputs of the block, where they are available for further processing.

The "NEW_DATA" output is reset by resetting the "PA.REC_REQ" request input.

The "FB_UA_RM" communication block reports any errors it detects during execution of the read job to the "FB_XXXPA" block. The "FB_XXXPA" block deactivates the read job if an error occurs ("REC_REQ" = 0). This also happens if the "FB_XXXPA" block itself detects an error.

Normalization/recalculation of the requested data

Following a request by a positive edge at request inputs "L_CAL" or "OP_CAL", the data (e.g., QREC_ID (parameter set ID), CALUA_NA / CALUA_NO, parameter output data) are written to the output structure "UDT_UA_W_REQ_ST". The write job is activated by setting the "REC_REQ" bit.

The "L_CAL" input can be connected during CFC configuration. The "OP_CAL" input is set with the "Calculate" button in the faceplate. Whether or not the "L_CAL" parameter is used and

the "Calculate" button is enabled in the process picture depends on the "CALOP_EN" parameter of the "FB_XXXPA" block. Recalculation is only performed when "CALOP_EN = 1". Individual parameters are only calculated with formats 0-5 (INTEGER and REAL).

Once the write job has been processed, the "FB_UA_WM" block writes the confirmation data (including REQ_ID) to the "UDT_UA_W_ST" input structure of the block and sets the "QWR_OK" bit.

The input data are only assigned to the block's write job if the ID of the block (REQ_ID in the output structure) matches the "REQ_ID" of the input structure.

Once the input structure data have been assigned to the write job and the "QWR_OK" bit has been set, the data are checked for validity and plausibility. The system checks whether the request data in the input and output structure match.

If the data are plausible and valid, the "WR_OK" output of the "FB_XXXPA" block is set.

A new read job is then triggered and the data are read from the archive that was defined with the "CALUA_NA" and "CALUA_NO" inputs.

The C function "FB_ReCalcValueByNormPM.fct" is called upon recalculation. This function recalculates the values.

You can find more detailed information on normalization/recalculation in the section C functions for normalization/recalculation (Page 158).

The "WR_OK" output is reset when the "L_CAL or OP_CAL" request input is reset.

The "FB_UA_WM" communication block reports any errors it detects during execution of the write job to the "FB_XXXPA" block. The "FB_XXXPA" block deactivates the write job if an error occurs ("L_CAL" or "OP_CAL" = 0). This also happens if the "FB_XXXPA" block itself detects an error.

A positive edge at the "RESET" input resets the "WR_OK" output and the error messages relating to the write job.

Repetition of jobs

The "COUNT" input value defines how often a job is to be repeated if an error occurs (COUNT = 0 -> no repetition). Until the specified number of job repetitions has been reached, the job is reactivated by FB_XXXPA after a wait cycle.

An error message is output after the last repetition.

A new job is only possible if the request signal at the request inputs "PA.REC_REQ", "L_CAL" or "OP_CAL" is set again.

Whether or not recalculation can be run depends on the "CALOP_EN" bit.

Message box in the faceplate

A message box with user-defined texts can be displayed in the faceplate. These texts can be created beforehand in multilingual format in the text library. The corresponding text library ID must then be entered at the "MSBO_NO" input of the "FB_XXXPA" block.

The message box display is controlled using the "MSBO_ACT" input. The message box is displayed when "MSBO_ACT = 1".

You can use the "ACK_DIS" block input to specify whether a confirmation button is to be displayed in the message box. The value of the "ACK_DIS" input is transferred in the block to the "DIS_ACK" output.

Confirming with the button sets the in/out parameter "QMSBOACK". When the time set at the "IMP_T" input has expired, the "QMSBOACK" parameter is reset.

Release button in the faceplate

A user-defined release can be set up at the block. This does not affect the functions in the block, which means if you wish to implement a lock, pause or release, you need to use CFC interconnections. This ensures maximum flexibility for parameter assignment.

The block inputs REL_EN (auto) and RELOP_EN (manual) release the block inputs L_REL (auto) and OP_REL (manual). This means the values of these two inputs are not evaluated until the corresponding REL_EN / RELOP_EN input is set. If REL_EN and RELOP_EN are set, a "Release" button appears in the faceplate. The operator can use this button to set QRELEASE provided the L_REL is also set. This means that the three block inputs REL_EN, RELOP_EN and L_REL must be set for the release button in the faceplate to take effect.

The QRELEASE block output can only be set with the L_REL block input by the AS program. In this case, only the REL_EN input may be set (see also Figure 11).

Resetting the REL_EN input also resets the RELOP_EN input.

5.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_XXXPA" block:

Parameter	Description	Type	Default	WinCC tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
MSBO_ACT	Activates message box	BOOL	0	x
MSBO_NO	Text ID displayed in the message box	DINT	0	x
ID	ID of the block	INT	0	-
CALUA_NA	Archive name (recalculation)	STRING [16]	''	-
CALUA_NO	Archive number (recalculation)	DINT	0	-
EDITABLE	Parameter set can be edited in the faceplate: 0 = no, 1 = yes	BOOL	0	x
COUNT	Number of repetitions if an error occurs	INT	1	-
RESET	1 = Reset read interface	BOOL	0	-
MODE	Mode	WORD	0	-
PARA_1 …. PARA_5	Free parameter (1 to 5) for recalculation	DINT	0	x

Parameter	Description	Type	Default	WinCC tag
ACK_DIS	Display of button for confirmation	BOOL	0	x
REL_EN	Activates release	BOOL	0	-
RELOP_EN	Activates operator release	BOOL	0	x
CALOP_EN	Activates recalculation	BOOL	0	x
L_REL	Automatic release	BOOL	0	-
L_CAL	Triggers recalculation	BOOL	0	-
TIME_MON	Monitoring time for read job	REAL	30.0	-
IMP_T	Pulse duration	REAL	3.0	-
R_REC_ST	Feedback structure from FB_UA_RM	UDT_UA_R_ST	-	-
W_REC_ST	Feedback structure from FB_UA_WM	UDT_UA_W_ST	-	-
PA	FB_JOB feedback structure	UDT_PA_REQ	-	-
DBLINK01 …. DBLINK10	DB link (1 to 10)	ANY	-	-
MSG_EVID	ALARM_8P option: EV_ID	DWORD	0	-
"x": WinCC tag available; "-": WinCC tag not available				

Output parameters

The following table shows the output parameters of the "FB_XXXPA" block:

Parameter	Description	Type	Default	WinCC tag
QRUN_UP	Status display "Startup active" = 1	BOOL	0	-
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	x
QGR_ERR	Function block group error/fault	BOOL	0	
QERR_VER	1 = Version check error	BOOL	0	
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0	-
QERR	1 = Read error, data header: Requested data and read data do not match	BOOL	0	-
QMON_ERR	1 = Read job monitoring timeout	BOOL	0	-
QLEN_ERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0	-
QDB_ERR	DB link error flag 1=error	BOOL	0	-
QLINKERR	DB link error flag 1=error	BOOL	0	-
QRM_ERR	FB_UA_RM: 1 = Read error, requested data and read data do not match	BOOL	0	-
QRM_AERR	FB_UA_RM: 1 = Archive error, invalid data in receive buffer	BOOL	0	-
QRM_CERR	FB_UA_RM: 1 = Communication error, error message from "BRCV" coupling blocks	BOOL	0	-
QRM_PERR	FB_UA_RM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-

Parameter	Description	Type	Default	WinCC tag
QWM_ERR	FB_UA_WM: 1 = Write error, requested data and written data do not match	BOOL	0	-
QWM_AERR	FB_UA_WM: 1 = Archive error, invalid data in send buffer	BOOL	0	-
QWM_CERR	FB_UA_WM: 1 = Communication error, error message from "BSEND" coupling blocks	BOOL	0	-
QWM_PERR	FB_UA_WM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QREQ_RM	Read job active	BOOL	0	-
QREQ_WM	Write job active	BOOL	0	-
QREQ_ACT	Multiplexer job active	BOOL	0	-
NEW_DATA	Data have been read	BOOL	0	-
WR_OK	Data have been written	BOOL	0	-
R_REQ_ST	Request structure to FB_UA_RM	UDT_UA_R_REQ_ST	-	-
W_REQ_ST	Request structure to FB_UA_WM	UDT_UA_W_REQ_ST	-	-
QPA	Request structure to FB_JOB	UDT_PA_CON	-	-
QUNIT	Unit	STRING[32]	''	x
QSP	Setpoint	REAL	0.0	x
QBA_NA	Batch name	STRING[32]	''	x
QBA_ID	Batch ID	DWORD	0	x
QID	Read data: ID	INT	0	x
QARCH_NA	Read data: Archive name	STRING[16]	''	-
QARCH_NO	Read data: Archive number	DINT	0	-
QREC_ID	Read data: Data record number in user archive	DINT	0	x
QNO_REC	Read data: Number of rows to be read	DINT	0	-
QCOLUMN	Read data: Column number at which reading commences	DINT	0	-
QNO_COL	Read data: Number of columns to be read	DINT	0	-
QOFFSET	Read data: Offset, application-specific value	DINT	0	-
QCAL_LEN	Read data: Data length calculated in the OS script, application-specific value	DINT	0	-
QRESERVE	Read data: Reserve	DINT	0	-
QNAME	Read data: Parameter set name	STRING[64]	''	x
QDESCR	Read data: Description	STRING[128]	''	x
QUOM	Read data: Unit	STRING[16]	''	x
QNORMAM	Read data: Reference quantity	REAL	0.0	x
QVERSION	Read data: Version	REAL	0.0	x
QCH_USER	Read data: Login name of the operator who made the last change	STRING[32]	''	x
QCH_DAT	Read data: Time of change part 1	DWORD	16#08123123	x

Parameter	Description	Type	Default	WinCC tag
QCH_DATB	Read data: Time of change part 2	DWORD	16#59590000	x
QVAL_DAT	Read data: Validity time part 1	DWORD	16#08123123	x
QVAL_DATB	Read data: Validity time part 2	DWORD	16#59590000	x
QREL	Read data: Permission	BOOL	0	x
QREL_DAT	Read data: Release date part 1	DWORD	16#08123123	x
QREL_DATB	Read data: Release date part 2	DWORD	16#59590000	x
INT1	Read data: Example, parameter1	DINT	0	x
REAL1	Read data: Example, parameter2	REAL	0.0	x
BOOL1	Read data: Example, parameter3	BOOL	0	x
STRING1	Read data: Example, parameter4	STRING[32]	''	x
ENUM1	Read data: Example, parameter5	DINT	0	x
MAT1	Read data: Example, parameter6	DINT	0	x
DATE1	Read data: Example, parameter7 part 1	DWORD	0	x
DATE1B	Read data: Example, parameter7 part 2	DWORD	0	x
QMSBOACK	Message box confirmed	BOOL	0	x
DIS_ACK	Message box: Confirmation button is displayed	BOOL	0	x
QDBLINK	DB link	BOOL	0	x
QRELEASE	Release active	BOOL	0	x
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST	-	-
"x": WinCC tag available; "-": WinCC tag not available				

In/out parameters

The following table shows the in/out parameters of the "FB_XXXPA" block:

Parameter	Description	Type	Default	WinCC tag
OPMSBOAC	Operator input: Message box confirmation	INT	0	x
OP_REL	Operator input: Permission	BOOL	0	x
OP_CAL	Operator input: Recalculation	BOOL	0	x
APF_AS	APF function block version in AS	WORD	0	x

Parameter	Description	Type	Default	WinCC tag
APF_OS	APF function block version in OS	WORD	0	x
GEN_AS	Generation information in AS	DWORD	0	x
GEN_OS	Generation information in OS	DWORD	0	x
"x": WinCC tag available; "-": WinCC tag not available				

5.1.5 Error handling

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. The error bits are available at the block as an output signal. In the event of an error, the request signal "REC_REQ" must be reset to confirm that the error information has been processed.

The following error messages can be generated:

Output structure	Output	Message text	Meaning
	QERR	QERR data error read job ID "id" REC_ID "rec_id"	requested data and read data do not match
QMON_ERR	QMON_ERR	QMON_ERR runtime error read job ID "id" REC_ID "rec_id"	Read job monitoring timeout
	QLEN_ERR	QLEN_ERR data block length of receive data is not correct ID "id" REC_ID "rec_id"	Length of the read data does not match the length of the block outputs
	QPARAMF	QPARAMF parameter assignment error ID "id" REC_ID "rec_id"	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0
	QERR_VER	QERR_VER Version check error	Error in the version information

5.1.6 Messaging

The "FB_XXXPA" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The version error will additionally be linked to message structure (SIG_5). The message can either be passed through the message structure ALARM_ST to the message block FB_AL_XX or FB_JOB will report the error.

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Length of the read data does not match the length of the block outputs	$$BlockComment$$ QLEN_ERR data block length of the receive data is not correct ID @4%d@ REC_ID @5%d@ active ID @6%d@
SIG_2	AS process control Message - fault	Read job monitoring timeout	$$BlockComment$$ QMON_ERR runtime error read job ID @4%d@ REC_ID @5%d@ active ID @6%d@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QERR data error read job ID @4 %d@ REC_ID @5%d@ ID @6%d@
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@
SIG_5	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER error in version check
SIG_6	No message	-	-
SIG_7	No message	-	-
SIG_8	No message	-	-

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	-	
SD_7	-	
SD_8	-	
SD_9	-	
SD_10	-	

5.2 FB_PA_ST

5.2.1 Description

Object properties

The following table shows the object properties of FB_PA_ST:

Name	Number	Family	Module
FB_PA_ST	FB1795	APF	Parameter management

Area of application

The block is used for coding the structure UDT_PA_REQ_ST (QPA_REQ) and for decoding the structure UDT_PA_CON_ST (PA_CON). This allows you to use and trigger a parameter set block (FB_XXXPA) individually.

Startup characteristics

This block has no startup characteristics.

Time characteristics

The block must be called from a cyclic interrupt OB.

Communication

Communication on the AS level is performed via the function block inputs/outputs.

There is no data communication with the OS.

5.2.2 Generation

The "FB_PA_ST" block is not generated by the APF Engineering Tool. It is part of the library supplied with APF.

5.2.3 Functions

Overview

The block can be used to connect the inputs or the outputs of the FB_XXXPA block and has the following functionality:

- Combines individual inputs in the UDT_PA_REQ input structure of the FB_XXXPA block. This structure is used to transfer data and trigger reading of the parameter set data to FB_XXXPA.
- The UDT_PA_CON output structure of FB_XXXPA is output as individual outputs. This structure is used for feedback for the parameter set data request from FB_XXXPA

"Normal mode"

Together with a parameter set block in "Normal mode", the FB_PA_ST block can be used to read parameter sets from a parameter set archive corresponding to the parameter set block. The following links must be configured for this purpose:

- PA_ST.QPA_REQ -> XXXPA.PA
- XXXPA.QPA -> PA_ST.PA_CON

The inputs of the FB_PA_ST have the following functions here:

- REC_ID: The parameter set number to be read
- ACTIVEID: Not used
- REQ: Request bit to which the parameter set block responds upon a positive edge
- RESET: Reset of the parameter set outputs
- UNIT: Unit of the setpoint used for recalculating certain parameter set values
- ARCH_NA: Archive name (without "_001" ending) corresponding to the parameter set block (default: FB_PARAM)
- ARCH_NO: Archive number that belongs to the parameter set block (e.g.: 1 for ending "_001")
- BA_NA: Batch name
- BA_ID: Batch number
- SETPOINT: Setpoint displayed in the faceplate which is used for normalization / recalculation of certain values.
- PA_CON: Feedback information of the parameter set block output at the FB_PA_ST outputs

"Multiplexer mode"

Together with a parameter set block in "Multiplexer mode" (see description of parameter set management), the FB_PA_ST block can be used to read parameter set data from other parameter set blocks and copy them to the associated parameter set block. The following links must be configured for this purpose:

- PA_ST.QPA_REQ -> XXXPA.PA
- XXXPA.QPA -> PA_ST.PA_CON

The inputs of the FB_PA_ST have the following functions here:

- REC_ID: The parameter set number of the parameter set block to be read
- ACTIVEID: The block ID ("ID" input) of the parameter set block to be read.
- REQ: Request bit to which the parameter set block responds upon a positive edge
- RESET: Reset of the parameter set outputs
- UNIT: Not used
- ARCH_NA: Not used
- ARCH_NO: Not used
- BA_NA: Not used
- BA_ID: Not used
- SETPOINT: Not used
- PA_CON: Feedback information of the parameter set block output at the FB_PA_ST outputs

NOTICE
If the connected parameter set block is in multiplexer mode, its DBLINKx inputs must be connected to the parameter set blocks from which parameter set data are to be read. In multiplexer mode, the FB_PA_ST block must always be supplied with REC_ID and ACTIVEID.

5.2.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_PA_ST" block:

Parameter	Description	Type	Default	WinCC tag
REC_ID	Record ID	DINT	0	-
ACTIVEID	Block ID	INT	0	-
REQ	1 = Data request	BOOL	0	-
RESET	1 = Reset	BOOL	0	-
UNIT	Unit	STRING[24]	''	-
ARCH_NA	Archive name	STRING[16]	''	-

Parameter	Description	Type	Default	WinCC tag
ARCH_NO	Archive number	DINT	0	-
BA_NA	Batch name	STRING[32]	''	-
BA_ID	Batch ID	DWORD	0	-
SETPOINT	Setpoint	REAL	0.0	-
PA_CON	Feedback structure of the data request	UDT_PA_CON_ST	0	-
"x": WinCC tag available; "-": WinCC tag not available				

Output parameters

The following table shows the output parameters of the "FB_PA_ST" block:

Parameter	Description	Type	Default	WinCC tag
NEW_DATA	1 = New data available	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	X
QERR_VERR	1 = Version check error	BOOL	0	-
QERR	1 = Read error, data header: requested data and read data do not match	BOOL	0	-
QMON_ERR	1 = Read job monitoring timeout	BOOL	0	-
QPARAMF	1 = Parameter assignment error, ID=0 or REC_ID=0 or ARCH_NA=' ' or TIME_MON=0	BOOL	0	-
QLEN_ERR	1 = Length of the read data does not correspond to length of block outputs	BOOL	0	-
QARCHERR	1 = Archive error	BOOL	0	-
ACTIVE_ID	Block ID	INT	0	-
QPA_REQ	Request structure	UDT_PA_REQ	0	-
"x": WinCC tag available; "-": WinCC tag not available				

In/out parameters

The following table shows the in/out parameters of the "FB_PA_ST" block:

Parameter	Description	Type	Default	WinCC tag
APF_AS	APF function block version in AS	WORD	0	x
APF_OS	APF function block version in OS	WORD	0	x
"x": WinCC tag available; "-": WinCC tag not available				

5.3 C functions for normalization/recalculation

If a value <= 0 is entered in the NORMAMOUNT (reference quantity) cell of the FB_PARAM_XXX archive, the parameter set with the values originally entered is used for job control. A value in this cell of > 0 means that the data record is to be normalized automatically and the normalized values of the individual parameters saved directly at the block.

The FB_PARAM_CALC_XXX archives are used for this normalization or indeed for data record recalculation–either manual or triggered by the process. Recalculation can also be triggered

using the FB_XXXPA block (CFC connection or a button in the faceplate). This is done using either the L_CAL block input or the "Calculate" button in the faceplate. Whether or not this option is available depends on the CALOP_EN enable bit at the block. CALOP_EN shows and enables the button in the faceplate, and there is only a response to the L_REL input if CALOP_EN = 1.

Individual parameters are only calculated with formats 0-5 (INTEGER and REAL).

When creating the corresponding attribute archive, you can enter a number in the NORM column which is then assigned to a specified mathematical formula to be used to calculate these parameters. For maximum flexibility, two C functions have been created in which you can enter user-defined mathematical formulas.

The "FB_CalcValueByNormPM.fct" function used for automatic data record normalization and the "FB_ReCalcValueByNormPM.fct" function used for recalculations contain the following parameters, which can be used directly in the user-defined formulas:

Parameter	Meaning
iNormIndex	Corresponds to the entry in the "NORM" column of the "FB_PARAMAT-TRIB_XXX" user archive and is only used for differentiating between the mathematical formula to be applied
dNormParameter	Unchanged parameter value to be normalized or recalculated
fNormAmount	Reference quantity from the corresponding data record / parameter set
iParameter1	User-defined parameter in 32-bit integer format (FB_XXXPA block input "PARA_1")
iParameter2	User-defined parameter in 32-bit integer format (FB_XXXPA block input "PARA_2")
iParameter3	User-defined parameter in 32-bit integer format (FB_XXXPA block input "PARA_3")
fParameter4	User-defined parameter in real or float format (FB_XXXPA block input "PARA_4")
fParameter5	User-defined parameter in real or float format (FB_XXXPA block input "PARA_5")
fParameter6	Job setpoint (FB_XXXPA block output "QSP")
dRetValue	Result or feedback value of the C function

Cases 0 and 1 (NORM column in the "FB_PARAMATTRIB_XXX" archive or "iNormIndex" parameter of the function) have the following default values:

Function	NORM	Meaning
FB_CalcValueByNormPM.fct	0	Parameter is applied unchanged, which means it is not normalized
FB_CalcValueByNormPM.fct	1	Parameter is normalized on a linear basis
FB_ReCalcValueByNormPM.fct	0	Parameter is applied unchanged, which means it is not recalculated
FB_ReCalcValueByNormPM.fct	1	Parameter is applied unchanged, which means it is not recalculated

Example

The individual parameters to be normalized/recalculated with another formula can be adjusted either with the APF Engineering Tool or in the "FB_PARAMSTTRIB_XXX" archive. A unique number (>1) must be entered in the "NORM" column of the relevant individual parameters for this purpose. This figure must then be added to the two C functions as a case. The basic structure is already in place for cases 2 to 20. This always starts with "case NUMBER:" and ends with "break;". You must make sure that the text "return dRetValue;" always appears at the end of each case. This returns the result "dRetValue" which is ultimately transferred to the AS.

Example for case 2:

```
case 2:
    dRetValue = (dNormParameter / 2) + iParameter1;
    return dRetValue;
break;
```

In the example, the value is always halved regardless of the setpoint and of other parameters, and the freely interconnectable value "iParameter1" (corresponds to block input PARA_1) is added to the result.

With manual recalculation, the reference quantity is not used. When this function is used, the corresponding recalculation function (FB_ReCalcValueByNorm.fct) is also implemented for parameters which were not previously normalized (FB_CalcValueByNorm.fct) because there was no reference quantity. Only the NORM column is therefore relevant for recalculation and is applied accordingly. If a parameter is not to be changed in recalculation, the value must be returned unchanged in the "case" corresponding to the figure (NORM column) in the recalculation function.

5.4 User archives

5.4.1 FB_PARAMGROUP

The parameter set groups are defined in the "FB_PARAMGROUP" user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of parameter set group	INTEGER 32
NAME	Name of parameter set group	STRING[32]
DESCR	Description of parameter set group	STRING[128]
TXTLIB	ID for multilingual management in the text library	INTEGER 32

5.4.2 FB_PARAMATTRIB_XXX

Up to 100 parameters and their properties are defined in the "FB_PARAMATTRIB_XXX" user archives (XXX=number of the parameter set group). Each archive is assigned to a parameter set group and numbered accordingly.

The archive has the following structure:

Element	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Parameter name	STRING[32]
UOM	Unit of parameter	STRING[16]
FORMAT	Format of parameter	INTEGER 32
ENUM_MAT	Coding/enumeration of parameter	STRING[16]
HL	High limit of operator control for numerical values	DOUBLE
LL	Low limit of operator control for numerical values	DOUBLE
DESCR	Description of the parameter	STRING[128]
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the output in the corresponding function block	STRING[20]
VISIBLE	Display of the attribute in the process picture	INTEGER 32
EDITABLE	Can be edited on the AS	INTEGER 32
NORM	The value of the archive element "NORM" determines the algorithm in the "FB_CalcValueByNorm" / "FB_RecalcValueByNorm" functions which is used for normalizing the corresponding parameter.	INTEGER 32

Format coding

The following table shows the format coding in "INTEGER":

Integer value	Format
0	INTEGER 32
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal places
4	REAL with 3 decimal places
5	REAL with 4 decimal places
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
20xx	DATE
21xx	TIME
22xx	DATE AND TIME

Integer value	Format
30	BOOL
40	ENUMERATION
50	MATERIAL

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

The corresponding name of the S7 enumeration (format 40) is available for the "ENUM_MAT" archive element. The following material identifiers (format 50) are also possible:

Integer value	Format
001001	Material class 1 / material type 1
001002	Material class 1 / material type 2
001003	Material class 1 / material type 3
...	...
002001	Material class 2 / material type 1
002002	Material class 2 / material type 2
002003	Material class 2 / material type 3
...	...
010008	Material class 10 / material type 8
010009	Material class 10 / material type 9
010010	Material class 10 / material type 10

5.4.3 FB_PARAM_XXX

Up to 200 parameter sets and their parameter values are saved in the "FB_PARAM_XXX" user archives (XXX=number of the parameter set group).

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID of the parameter set	INTEGER 32 *
NAME	Name of the parameter set	STRING 64
DESCR	Description of the parameter set	STRING 128
UOM	Unit of the parameter set	STRING 16
NORMAMOUNT	Reference quantity of the parameter set	REAL
VERSION	Current version of the parameter set	REAL
CHANGEUSER	Name of the operator who made the last change	STRING 32
CHANGEDATE	Date of the last change	DATE
VALIDDATE	Date of validity	DATE
RELEASED	Permission	BOOL
RELEASEDATE	Release date	DATE
PARAM_<ioname> ….. PARAM_<ioname>	Value of parameters 1 to n	Represents the formats of the attributes from the corresponding user archive "FB_PARAMATTRIB_XXX"

Note

*The manual input in the editor dialog is limited to 2,000,000,000.

The parameter attributes are assigned the column name "PARAM_" with the "FCT_BLOCK_IO_NAME" of the FB_PARAMATTRIB_XXX table.

Note

Forming the parameter column name (a maximum of 20 characters) from "PARAM_" and "<ioname>" limits the actual useful length of the IO name to 14 characters.

5.4.4 FB_PARAM_CALC_XXX

In the "FB_PARAM_CALC_XXX" user archives (XXX = Number of the parameter set group), one parameter set and its parameter values are saved for each "FB_XXXPA" and "FB_UA_WM" block.

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID of the parameter set	INTEGER 32
NAME	Name of the parameter set	STRING 64
DESCR	Description of the parameter set	STRING 128
UOM	Unit of the parameter set	STRING 16
NORMAMOUNT	Reference quantity of the parameter set	REAL

Element	Description	Format
VERSION	Current version of the parameter set	REAL
CHANGEUSER	Name of the operator who made the last change	STRING 32
CHANGEDATE	Date of the last change	DATE
VALIDDATE	Date of validity	DATE
RELEASED	Permission	BOOL
RELEASEDATE	Release date	DATE
PARAM_<ioname> ….. PARAM_<ioname>	Value of parameters 1 to n	Represents the formats of the attributes from the corresponding user archive "FB_PARAMAT-TRIB_XXX"
REQ_ID	ID of the "FB_XXXPA" block used	INTEGER
WM_ID	ID of the "FB_UA_RM" block interconnected	INTEGER

5.5 Operator control and monitoring

5.5.1 Block icons

@TypicalsAPF.pdl

The "@TypicalsAPF.pdl" template picture contains templates for automatically generating block icons from the plant hierarchy.

@Template_APF.pdl

The "@Template_APF.pdl" template picture contains an additional template for configuring the block icon in the process pictures.

5.5.2 Picture objects

The interface for parameter management operator control and monitoring consists of the following process pictures:

WinCC flow charts

- FB_PARAM_ED.pdl
  The picture allows you to create, change, delete and copy parameter sets.
- @FB_PARAM_ED_WINDOW.pdl
  You can use this picture to edit the attribute parameters of the parameter sets. The picture is implemented as a window in the "FB_PARAM_ED" picture and cannot be used separately.

Faceplates

- @PG_FB_XXXPA.PDL
  Faceplate for editing the data of the parameter set loaded to the AS.
- @FB_PARAM_WINDOW.PDL
  You can use this picture to edit the attribute parameters of the parameter sets. This picture is integrated as a window in the "@PG_FB_XXXPA" faceplate and cannot be used separately.

Note

The process pictures are pre-configured and do not need to be edited.

For detailed information on interface operator control and monitoring in process mode, see the APF Operator Manual.

5.5.3 Messages

The following messages can be output:

Message class	Event	Message text
Operator message	Save button	@7%s@: @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ parameter set: @6%s@
Operator message	Save button	@7%s@: @2%s@: new = @5%s@ @8%s@ old = @4%s@ @8%s@ parameter set: @6%s@
Operator message	Save button	@7%s@: parameter set @6%s@ changed
Operator message	Save button	@7%s@: parameter set @6%s@ created
Operator message	Delete button	@7%s@: parameter set @6%s@ deleted
Operator message	Save button	@1%s@,@7%s@: @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ parameter set: @6%s@
Operator message	Save button	@1%s@,@7%s@: @2%s@: new = @5%s@ @8%s@ old = @4%s@ @8%s@ parameter set: @6%s@
Operator message	Save button	@1%s@,@7%s@: parameter set @6%s@ changed
OS process control	Error in the screen	Parameter set: @6%d@ error code: @4%d@
OS process control	Error in AS - OS communication via "FB_UA_RM" / "FB_UA_WM" blocks	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control	Error in AS - OS communication via "FB_UA_RM" / "FB_UA_WM" blocks	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
Operator message	Save button	@7%s@: High limit @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ parameter set group: @6%s@

Message class	Event	Message text
Operator message	Save button	@7%s@: Low limit @2%s@: new = @5%.4f@ @8%s@ old = @4%.4f@ @8%s@ parameter set group: @6%s@
Operator message	Save button	@7%s@: Parameter set limits @6%s@ changed
OS process control	Error in the screen	Parameter set: @6%d@ error code: @4%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
OS process control error	Cyclic backup action: The export file could not be created.	@7%s@: Backup file (@2%s@) could not be exported
OS process control error	Cyclic backup action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the backup file
OS process control error	Cyclic backup action: Directory 1 could not be created	@7%s@: Backup directory 1 does not exist / could not be created
OS process control error	Cyclic backup action: Directory 2 could not be created	@7%s@: Backup directory 2 does not exist / could not be created
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 1 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 2 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 1 is corrupt
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 2 is corrupt
OS process control error	Cyclic backup action: Call the external backup program failed	@7%s@: Initialization of the backup program failed

Message associated values

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%d@	Not used / QBA_NA
Message associated value 2	@2%s@	Parameter set attribute / backup file
Message associated value 3	@3%d@	Not used / ID of the active FB_XXXPA block / QBA_ID

Message associated value	Format	Value
Message associated value 4	@4%s/d@	Old value (string or number)
Message associated value 5	@5%s/d@	New value (string or number)
Message associated value 6	@6%s@	Parameter set name + ID, here: Parameter set name (ID)
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	Unit / UOM
Message associated value 9	@9%s@	Not used
Message associated value 10	@10%s@	Source, here: Parameter management / QUNIT

For additional information, refer to the PCS 7 online help.

Order management

6

6.1 FB_JOB

6.1.1 Description

Object properties

The following table shows the object properties of FB_JOB:

Name	Number	Family	Module
FB_JOB	FB1785	APF	Order management

Area of application

The "FB_JOB" block has the following functions:

- Managing of jobs (job data records) in a job list
- Providing of job data records of the job list for the AS and OS
- Requesting parameter sets associated with the job data records using the "FB_XXXPA" block
- Interfaces for creating jobs using OS / AS / IT
- Interfaces for job control using OS / AS / IT
- Generating a status graph using signals from the job control to the process and feedback signals from the process to the job control (process = job processing)
- Signal exchange for job execution with SFC possible

Startup characteristics

The "FB_JOB" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

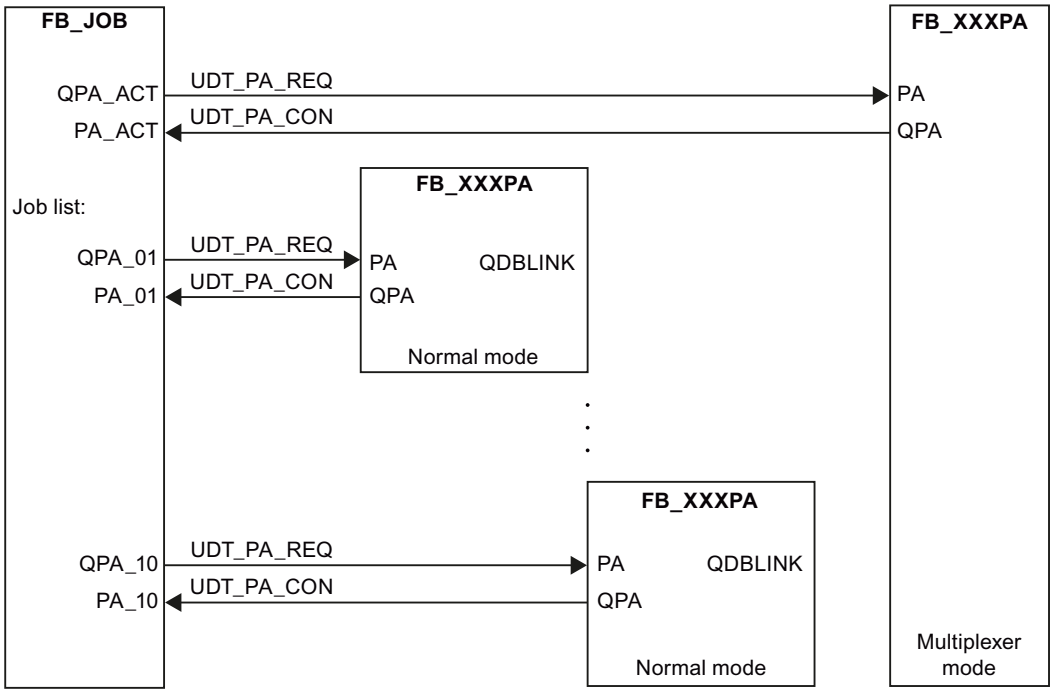
The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

AS communication

Data exchange with the "FB_XXXPA" block on the AS level is performed via the "UDT_PA_REQ_ST" (request) and "UDT_PA_CON_ST" (feedback) structures.

Up to 10 jobs can be managed in a job list. Each item in the job list is assigned an "FB_XXXPA" function block (normal mode) in the AS. The "FB_XXXPA" block manages the parameter set belonging to the job. A second "FB_XXXPA" block in multiplexer mode can copy the parameter set of the activated job from the corresponding FB_XXXPA (normal mode) and manage it (parameter set of the job in the job interface).

The following figure depicts the data communication:



Note

The "FB_JOB" function block and all corresponding "FB_XXXPA" function blocks must be located in the same block folder.

The "FB_XXXPA" function block in multiplexer mode must be assigned the name of the "FB_JOB" function block plus the index 00. The function blocks in normal mode must be assigned the name of the "FB_JOB" function block plus the index 01 to 10.

Example:

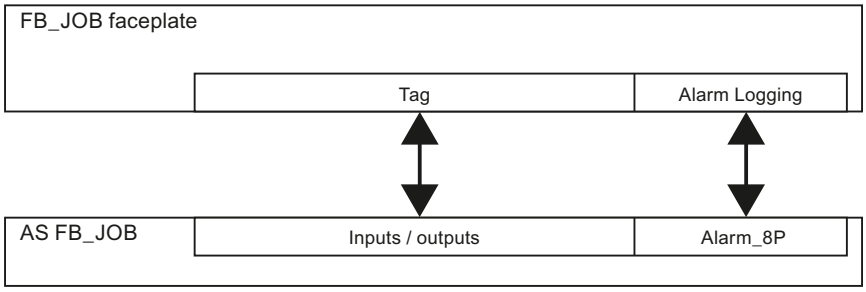
The name "FB_JOB" has been assigned to the "FB_JOB" function block in the CFC.

This results in the following naming convention in the CFC:

The "FB_XXXPA" function block in multiplexer mode has the name "FB_JOB00", and the "FB_XXXPA" function blocks in normal mode have the names "FB_JOB01", "FB_JOB02, "FB_JOB03, …, "FB_JOB10".

AS - OS communication

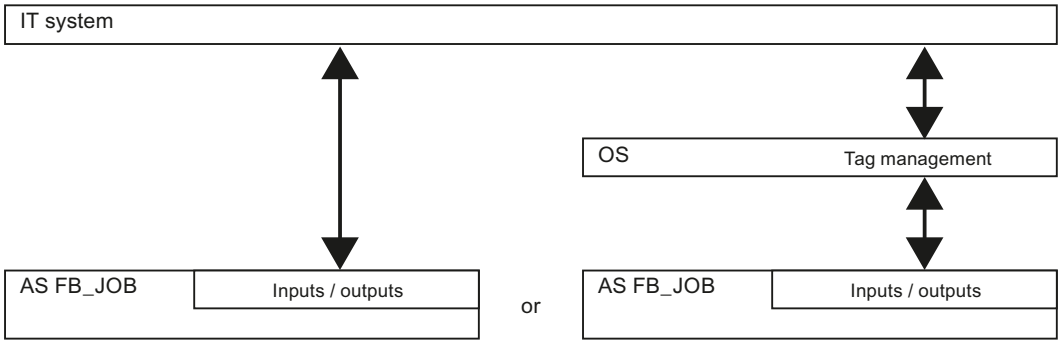
For operator control and monitoring, signal exchange over the inputs/outputs of the "FB_JOB" takes place between the AS and the OS. ALARM_8P blocks are used to implement data communication for operational / fault messages.



The scripts used in the OS are only for display / input / message control purposes in the faceplate. Job management is using the "FB_JOB" block only.

AS-IT system communication

Data exchange between the AS and an IT system can take place either directly over the inputs and outputs of the "FB_JOB" block, or using OS Tag Management. Separate "IT_..." inputs and outputs are available at the "FB_JOB" block for data communication.



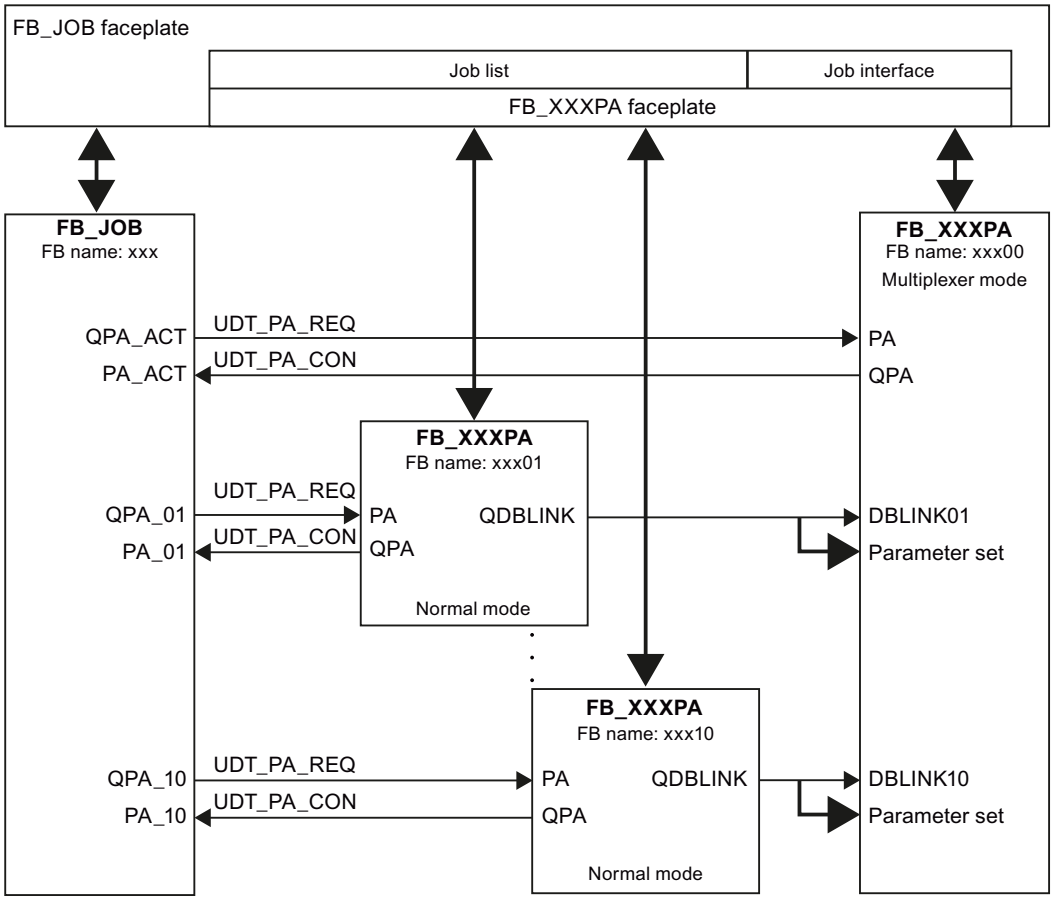
6.1.2 Generation

The "FB_JOB" block is not generated by the APF Engineering Tool. It is available in the APF library.

6.1.3 Functions

Overview

The figure below provides an overview of how the "FB_JOB" block works:



Operating mode

The mode is set using the "MODE" parameter:

MODE	Meaning	Symbol
Bit 1	Specifies the job name: 0 = Automatic 1 = Manual	DENOM_TITLE
Bit 2	Specifies job ID 0 = Automatic 1 = Manual	DENOM_ID

MODE	Meaning	Symbol
Bit 8	1 = Automatic start if a job is available; "Activate job" automatically starts the jobs in the job list one after the other; even following an empty job list, the next job in the job list is automatically started again.	CYCLE_AUTO_START
Bit 9	1 = Automatic start with "Activate job"	AUTO_JOB_START
Bit 10	Feedback monitoring of the control commands; 0 = No, 1 = yes	MONITORING
Bit 11	Status tracking; 0 = No, 1 = Yes	FEEDBACK_TRACKING

Operating mode

You can define the following modes at the inputs of the "FB_JOB" block:

- Internal/external
  You can use the "LIOPSELJ" and "EXT_L" inputs to specify whether the system switches between "Internal" and "External" mode via the AS or the OS:

LIOPSELJ	EXT_L	Operating mode
0	-	Switch via OS
1	0	Internal
1	1	External

- Manual/Automatic
  You can use the "LIOPSELC" and "AUT_L" inputs to specify whether the system switches between "Manual" and "Automatic" mode via the AS or the OS:

LIOPSELC	AUT_L	Operating mode
0	-	Switch via OS
1	0	MANUAL
1	1	AUTO

- One time/Cyclic
  You can use the "LIOPSELM" and "CYCLIC_L" inputs to specify whether the system switches between "One time" and "Cyclic" mode via the AS or the OS:

LIOPSELM	CYCLIC_L	Operating mode
0	-	Switch via OS
1	0	One time
1	1	Cyclic

Displaying mode / operating mode

The status word "QSTATUS" is used to display the selected modes:

QSTATUS	SYMBOL	Meaning
Bit 0	EXTINT	0 = Internal mode, 1 = External mode
Bit 1	MANAUT	0 = Manual mode 1 = Automatic mode
Bit 2	LINKIT	With external mode: 0 = "LINKED" 1 = "IT"
Bit 8	DENOM_TITLE	Specifies the name: 0 = Automatic in the AS 1 = In the faceplate
Bit 9	DENOM_ID	Specifies the ID: 0 = Automatic in the AS 1 = In the faceplate
Bit 10	CYCLE_MODE	0 = One time mode 1 = Cyclic mode
Bit 15	CSF	1 = External error, control system

Format

The "SP_FORMAT" parameter is used to define the decimal places and the sign.

SP_FORMAT	Meaning
<= -7	No decimal places, with sign
- 1 to - 6	1 to 6 decimal places, with sign
0	No decimal places
1 to 6	1 to 6 decimal places
>= 7	No decimal places

Function releases

The following outputs of the "FB_JOB" function block provide information on the function enables (1 = Enabled, 0 = Not enabled).

Function	Output	Conditions
"Start"	RELSTART	• Job processing is not active • Job available • Job has been released (RUN_EN) • "Abort" job is not active • Job feedback has not been started • Job feedback not running
"Hold"	RELHOLD	• Job processing is active • "Hold" job is not active
"Resume"	RELRESUM	• Job processing is active • "Resume" job is not active • "Abort" job is not active • Job has been released (RUN_EN) • "JOB_ON"
"End"	RELEND	• Job processing is active • "End" job is not active • "Abort" job is not active • "JOB_ON"
"Abort"	RELABORT	• "Abort" job is not active • Job available
"Up"	RELPOSUP	Job position 2 or higher selected
"Down"	RELPOSDO	Job position between first and second-to-last possible position selected
"Create job"	RELNEW	• Job list not full • "New" job is not active • Data request via "FB_XXXPA" is not active • No pending data request errors
"Delete job"	RELDEL	Job position selected
"Activate job"	RELACJO	No job in the job interface and at least one job available in the job list
"Apply"	RELTAOV	"New" job is active

Enabled operations

You can use the enable bits of the "FB_JOB" block input word "OP_EN" to specify whether buttons/functions in the "FB_JOB" faceplate are enabled (1 = Enable, 0 = No enable).

The operator control enable signals are available at the corresponding block outputs.

Function	OP_EN	Output	Additional conditions
"Start"	Bit 0	EN_START	• "Manual" mode • Job available • Job released • Job processing not started
"Hold"	Bit 1	EN_HOLD	• "Manual" mode • Job not paused • Job processing ON • "Hold" job is not active
"Resume"	Bit 2	EN_RESUM	• "Manual" mode • Job paused • Job released • Job not aborted • Job processing ON • "Resume" job is not active • "JOB_ON"
"End"	Bit 3	EN_END	• "Manual" mode • Job not aborted • Job processing ON • "End" job is not active • "JOB_ON"
"Abort"	Bit 4	EN_ABORT	• "Manual" mode • Job available • "Abort" job is not active
"Automatic"	Bit 5	EN_AUT	"Manual" mode
"Manual"		EN_MAN	"Automatic" mode
"Internal"	Bit 6	EN_INT	"External" mode
"External"		EN_EXT	"Internal" mode "Create job" not active
"Cyclic mode"	Bit 7	EN_CYCL	"One time" mode
"One time mode"		EN_ONETI	"Cyclic" mode
"New"	Bit 8	EN_NEW	• "Internal" mode • Job list not full • Data request via "FB_XXXPA" is not active • No pending data request errors
"Delete"	Bit 9	EN_DEL	• "Internal" mode • Job selected in list

Function	OP_EN	Output	Additional conditions
"Up"	Bit 11	EN_POSUP	• "Internal" mode • Job selected • New position available
"Down"		EN_POSDO	• "Internal" mode • Job selected in list • New position available
"Activate job"	Bit 12	EN_TAOV	"Internal" mode
"Apply"	Bit 13	EN_ACJO	• "Internal" mode • Job list not empty • Job interface empty
Manual name specification		EN_TITL	"MODE" bit 1 = 1
Manual ID specification		EN_ID	"MODE" bit 2 = 1

Showing/hiding buttons in the faceplate

You can use the enable bits of the "FB_JOB" block input word "HIDE_BUT" to specify whether buttons/functions in the "FB_JOB" faceplate are visible (1 = Visible, 0 = Hidden).

The operator control enable signals are available at the corresponding block outputs (1 = Visible, 0 = Hidden).

Button	HIDE_BUT	Output
"Manual / Auto"	Bit 0	DIS_MAAU
"Internal"/ "External"	Bit 1	DIS_EXIN
"One time / "Cyclic"	Bit 2	DIS_MODE
"New"	Bit 3	DIS_EDIT
"Delete"	Bit 4	DIS_DEL
"Up"/"Down"	Bit 5	DIS_UPDO
"End"	Bit 6	DIS_END
"Abort"	Bit 7	DIS_ABOR
"Activate job"	Bit 8	DIS_ACJO
"Confirmation"	Bit 9	DIS_ACK

Message box

A message box with user-defined texts can be displayed in the faceplate. These texts can be created beforehand in multilingual format in the text library. The message box display is controlled using the "MSBO_ACT" (display) and "MSBO_NO" (text number) block inputs.

Input	Meaning
MSBO_ACT	"0" = Message box hidden "1" = Message box shown
MSBO_NO	Select message text = WinCC text library ID

You can use bit 9 of the "HIDE_BUT" block input to configure whether or not a confirmation button is to be displayed in the message box. The "Confirmation" button sets the "QMSBOACK" block output. When the time set at the "IMP_T" input has expired, the "QMSBOACK" output is reset.

Signal specification

In "External" mode, you use the "EXT_IT" input to specify which signals are to be evaluated for job management:

- EXT_IT = 0: Signals from inputs that are interconnected in the AS
- EXT_IT = 1: Signals from inputs that are interconnected with an IT system

In "Automatic" mode, you use the "L_IT" input to specify which signals are to be evaluated for job management:

- L_IT = 0: Signals from inputs that are interconnected in the AS
- L_IT = 1: Signals from inputs that are interconnected with an IT system

Creating a job

In "External" mode, a job can be created using the following inputs. For a job to be successfully applied, it must have a name, an ID, a valid parameter set number and possibly also a setpoint (depending on whether or not it is a reference quantity).

EXT_IT	Input	Description
0	EXT_NEW	"EXT_NEW" = 1 initiates the creation of a job. After the job data has been specified at the inputs, the order data record is entered in the job list with "EXT_TAOV" = 1. "EXT_NEW" = 0 cancels the creation of a job.
	EXT_RCTL	The name is specified at the "EXT_RCTL" (record title) input. This input is only relevant if "automatic name specification" is not activated at the AS "MODE" input.
	EXT_RCID	The ID is specified at the "EXT_RCID" (record id) input. This input is only relevant if "automatic ID specification" is not activated at the AS "MODE" input.
	EXT_RCPN	The parameter set number is specified at the "EXT_RCPN" (record parameter set number) input. The parameter set number must be available in the corresponding user archive.
	EXT_RCSP	The "setpoint" is specified at the "EXT_RCSP" (record setpoint) input. This input is only relevant if the preselected parameter set number is a reference quantity.
	EXT_RCDS	A job description can be specified at the "EXT_RCDS" (record description) input.
1	IT_NEW	"IT_NEW" = 1 initiates the creation of a job. After the job data has been specified at the inputs, the order data record is entered in the job list with "IT_TAOV" = 1. "IT_NEW" = 0 cancels the creation of a job.
	IT_RCTL	The name is specified at the "IT_RCTL" (record title) input. This input is only relevant if "automatic name specification" is not activated at the AS "MODE" input.
	IT_RCID	The ID is specified at the "IT_RCID" (record id) input. This input is only relevant if "automatic ID specification" is not activated at the AS "MODE" input.
	IT_RCPN	The parameter set number is specified at the "IT_RCPN" (record parameter set number) input. The parameter set number must be available in the corresponding user archive.
	IT_RCSP	The "setpoint" is specified at the "IT_RCSP" (record setpoint) input. This input is only relevant if the preselected parameter set number is a reference quantity.
	IT_RCDS	A job description can be specified at the "IT_RCDS" (record description) input.

Selecting a job (sorting)

In "External" mode, a job can be created using the following inputs of the "FB_JOB" block:

EXT_IT	Input	Description
0	EXT_SEL	Selection of a job from the job list with the ID (DWORD) of the job
	EXT_POUP	A positive edge moves the job selected at the "EXT_SEL" input up one position in the job list.
	EXT_PODO	A positive edge moves the job selected at the "EXT_SEL" input down one position in the job list.
1	IT_SEL	Selection of a job from the job list with the ID (DWORD) of the job
	IT_POUP	A positive edge moves the job selected at the "IT_SEL" input up one position in the job list.
	IT_PODO	A positive edge moves the job selected at the "IT_SEL" input down one position in the job list.

Activating a job

In "External" mode, a job can be activated using the following inputs of the "FB_JOB" block:

EXT_IT	Input	Description
0	EXT_ACJO	A positive edge at the "EXT_ACJO" input activates the job entered in position 1 in the job list. This removes the job from the job list and enters it in the job interface.
1	IT_ACJO	A positive edge at the "IT_ACJO" input activates the job entered in position 1 in the job list. This removes the job from the job list and enters it in the job interface.

Deleting a job

In "External" mode, a job can be deleted using the following inputs of the "FB_JOB" block:

EXT_IT	Input	Description
0	EXT_DEL	A positive edge at the "EXT_DEL" input deletes the job selected at the "EXT_SEL" input from the job list.
1	IT_DEL	A positive edge at the "IT_DEL" input deletes the job selected at the "IT_SEL" input from the job list.

Job interface

The data of the current job are available at the following outputs for further processing:

Output	Meaning
ACT_TITL	Job title
ACT_ID	Job ID
ACT_PNO	Parameter set
ACT_SP	Setpoint
ACT_AV	Actual value

Output		Meaning
ACT_DESC		Description
ACT_BDT1 ACT_BDT2		Time of creation
ACT_ORIG		Creator
ACT_SDT1 ACT_SDT2		Start time
ACT_STAT		With bit 0 = 1 "reference quantity"
ACTSTATE	Bit 0	1 = An "FB_ERROR" (AS input) has occurred during job execution
	Bit 1	1 = A control system error has occurred during job execution
	Bit 5	1 = Job completed normally
	Bit 6	1 = Job completed after "END" selected
	Bit 7	1 = Job ended with "Abort"

Once the job has ended, the job data from the job interface are saved and supplemented with the end time, and are then available at the following outputs for further processing:

Output		Meaning
LAS_TITL		Job title
LAS_ID		Job ID
LAS_PNO		Parameter set
LAS_SP		Setpoint
LAS_AVAL		Actual value
LAS_DES		Description
LAS_BDT1 LAS_BDT2		Time of creation
LAS_ORI		Creator
LAS_SDT1 LAS_SDT2		Start time
LAS_EDT1 LAS_EDT2		End time
LAS_STAT		With bit 0 = 1 "reference quantity"
LASSTATE	Bit 0	1 = "FB_ERROR" (AS input) occurred during job execution
	Bit 1	1 = Control system error occurred during job execution
	Bit 5	1 = Job completed normally
	Bit 6	1 = Job completed after "END" selected
	Bit 7	1 = Job ended with "Abort"

Automatic specifications

- Job title
  You can activate "automatic job title specification" mode at the AS "MODE" input. "QREQ_TL" = 1 at the AS output triggers a job title request. The job title specified at the "TLIN" input is applied with a positive edge at the "ACC_TLIN" input.
- Job ID
  You can activate "automatic ID specification" mode at the AS "MODE" input. "QREQ_ID" = 1 at the AS output triggers a job ID request. The job ID specified at the "IDIN" input is applied with a positive edge at the "ACC_IDIN" input.

Job list

The data records of the jobs created are available at the following outputs for further processing:

Output	Meaning
TIT_Pxx	Job list item xx: Name
ID_Pxx	Job list item xx: ID
PNO_Pxx	Job list item xx: Parameter set number
SP_Pxx	Job list item xx: Setpoint
DES_Pxx	Job list item xx: Description
BDT1_Pxx	Job list item xx: Time of creation part 1
BDT2_Pxx	Job list item xx: Time of creation part 2
ORI_Pxx	Job list item xx: Creator
STAT_Pxx	Job list item xx: Data record status (bit 0 = 1 -> reference quantity)
xx = Position in the job list 01 to 10	

Status signals for the job list

The status signals for the job list are available at the following outputs:

Output	Meaning
QRC_EDIT	Job is being edited / created = 1
QPA_REQ	Request to FB_XXXPA to read parameter set data = 1
QACC	New job successfully created = 1
QNOTACC	New job could not be successfully created = 1
QLI_FULL	Job list full = 1
QLI_EMPTY	Job list empty = 1
QVOID	Job interface is free
QJOB_EN	Next job can be activated
QPA_COPY	Request to FB_XXXPA (multiplexer mode) to copy parameter set data upon job activation = 1
Q_POS_CN	Number of occupied positions in the job list
Q_POS_PA	Data record number for the job selected in the job list
Q_FREEPO	Next free position in the job list
Q_FREERC	Next free data record number

Releasing a job

A job can be released for launch using the "RUN_EN" input. A job in process is paused if job release is reset. The input signal is available at the "QRUN_EN" output for further processing.

Input	Output	Meaning
RUN_EN	QRUN_EN	0 = Job disabled 1 = Job released

Controlling a job

In "Automatic" mode, a job can be controlled using the following inputs of the "FB_JOB" block:

L_IT	Input	Description
0	L_START	Once a job has been activated, it is ready for starting in the job interface. Following a positive edge at the "L_START" input, the outputs for starting the process are set and the job status is displayed in line with the feedback signals.
	L_HOLD	Following a positive edge at the "L_HOLD" input, the outputs for pausing the process are set and the job status is displayed in line with the feedback signals.
	L_RESUM	Following a positive edge at the "L_RESUM" input, the outputs for continuing the process are set and the job status is displayed in line with the feedback signals.
	L_END	Following a positive edge at the "L_END" input, the outputs for ending the process are set and the job status is displayed in line with the feedback signals from the AS inputs.
	L_ABORT	Following a positive edge at the "L_ABORT" input, the outputs for canceling the process are set and the job status is displayed in line with the feedback signals.
1	IT_START	Once a job has been activated, it is ready for starting in the job interface. Following a positive edge at the "IT_START" input, the outputs for starting the process are set and the job status is displayed in line with the feedback signals.
	IT_HOLD	Following a positive edge at the "IT_HOLD" input, the outputs for pausing the process are set and the job status is displayed in line with the feedback signals.
	IT_RESUM	Following a positive edge at the "IT_RESUM" input, the outputs for continuing the process are set and the job status is displayed in line with the feedback signals.
	IT_END	Following a positive edge at the "IT_END" input, the outputs for ending the process are set and the job status is displayed in line with the feedback signals from the AS inputs.
	IT_ABORT	Following a positive edge at the "IT_ABORT" input, the outputs for canceling the process are set and the job status is displayed in line with the feedback signals.

Job control status signals

The status signals for job control are available at the following outputs:

Input	Output	Meaning
FB_AV		Current actual value of the job in progress
	QJOB_ON	Job processing ON = 1, OFF = 0
	QJOB_ACTIV	Job in progress = 1, paused = 0

Control signals to/from the process

There are two alternative interfaces for signal exchange with the process. If an SFC block (sequential control system) is being used for process control, the signals from the SFC are interconnected directly with "FB_JOB". The SFC interface is activated at the "EN_SFC" input of the "FB_JOB" block:

- EN_SFC = 0: SFC interface deactivated
- EN_SFC = 1: SFC interface activated

EN_SFC	Input	Output	Meaning
0	FB_ON		Feedback: job execution on = 1
	FB_RUN		Feedback: job in progress = 1
	FB_HELD		Feedback: job paused = 1
	FB_END		Feedback: job completed = 1
	FB_ERROR		Error occurred during job execution = 1
		QSTART	Job processing ON = 1
		QHOLD	Job on hold = 1
		QRUN	Job start / resume = 1
		QGOEND	End job = 1
		QABORT	Abort job = 1
1	SFC_IDLE		Output signal from the SFC: IDLE - idle
	SFC_RUN		Output signal from the SFC: RUN - running
	SFC_HELD		Output signal from the SFC: HELD - held
	SFC_COMPLETED		Output signal from the SFC: COMPLETED - completed
	SFC_ABORTED		Output signal from the SFC: ABORTED - aborted
		SFC_START	For input signal at the SFC: START - start
		SFC_HOLD	For input signal at the SFC: HOLD - hold
		SFC_RESUME	For input signal at the SFC: RESUME - resume
		SFC_ABORT	For input signal at the SFC: ABORT - abort

"QJOBSTAT" status word

QJOBSTAT	Symbol	Meaning
Bit 8	ACT_PRES	1 = Job available in standard view (current job)
Bit 9	NEX_PRES	1 = At least one job available in job list
Bit 10	LIST_EMPTY	1 = Job list empty
Bit 11	LIST_FULL	1 = Job list full
Bit 12	EDIT	1 = Job is being entered
Bit 18	CYCLIC_ACTIVE	1 = Cyclic mode active
Bit 19	JOB_EN	1 = No job available in standard view (current job)
Bit 21	MSBO_ACK	1 = During message box acknowledgement time
Bit 22	JOB_ON	1 = Job started

QJOBSTAT	Symbol	Meaning
Bit 23	JOB_ACTIV	1 = Job running (not paused)
Bit 25	START	1 = Job "start
Bit 26	HOLD	1 = Job "hold"
Bit 27	RESUM	1 = Job "resume"
Bit 28	GO_END	1 = Job "end"
Bit 29	ABORTION	1 = Job "abort"

"QMSG" status word

QMSG	Symbol	Meaning
Bit 0	START	1 = Job start message
Bit 1	HOLD	1 = Job hold message
Bit 2	RESUM	1 = Job resume message
Bit 3	GOEND	1 = Job end message
Bit 4	ABORT	1 = Job abort message
Bit 8	NEW	1 = Create job message
Bit 9	DEL	1 = Delete job message
Bit 10	ACTIVE	1 = Activate job message

Feedback monitoring of the control commands

Control command feedback monitoring is activated by setting bit "10" of the "MODE" input (bit 10 = 1). The monitoring time is specified at the "TIME_MON" input. The following monitoring options have been implemented:

- "START" and no "JOB_ON" feedback
- No "START" and "JOB_ON" or "RUN" or "HELD" feedback
- "RESUME" and no "RUN" feedback
- "HOLD" and no "HELD" feedback

A fault message is sent when the monitoring time has elapsed.

Status tracking

Status tracking can be activated with "MODE" bit 11 = 1 and is executed in the "JOB_ON" = 1 state.

- Tracking to "HOLD"
  Following a positive edge at the input "SFC_HELD" or "FB_HELD" and "SFC_RESUME" or "QRUN" = 1
- Tracking to "RESUME"
  Following a positive edge at the input "SFC_RUN" or "FB_RUN" and "SFC_HOLD" or "QHOLD" = 1

6.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_JOB" block:

Parameter	Description	Type	Default	WinCC tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
NO_OF_PS	Number of job data records (max. of 10)	INT	10	x
UNIT	Plant unit	STRING[24]	'unit'	x
ARCH_NA	Archive name	STRING[16]	''	x
ARCH_NO	Archive number	DINT	0	x
SP_UOM	Unit of measurement for the setpoint	STRING[16]	''	x
SP_FORMAT	Format of the setpoint	DINT	2	x
MSBO_ACT	0 = Message window hidden; 1 = Message window displayed	BOOL	0	x
MSBO_NO	Number of the message text displayed in the message window	DINT	0	x
CSF	Control system error = 1	BOOL	0	-
EXT_MSG5	ALARM_8P: External message signal SIG_5	BOOL	0	-
EXT_MSG6	ALARM_8P: External message signal SIG_6	BOOL	0	-
EXT_MSG7	ALARM_8P: External message signal SIG_7	BOOL	0	-
EXT_MSG8	ALARM_8P: External message signal SIG_8	BOOL	0	-
RUN_EN	Job release = 1	BOOL	1	x
FB_ON	Feedback: job execution on = 1	BOOL	0	-
FB_RUN	Feedback: job in progress = 1	BOOL	0	-
FB_HELD	Feedback: job paused = 1	BOOL	0	-
FB_END	Feedback: job completed = 1	BOOL	0	-
FB_ERROR	Feedback: error occurred during job execution = 1	BOOL	0	-
EN_SFC	Signal exchange for job execution via SFC signals = 1	BOOL	0	-
SFC_IDLE	SFC output signal "IDLE"	BOOL	0	-
SFC_RUN	SFC output signal "RUN"	BOOL	0	-
SFC_HELD	SFC output signal "HELD"	BOOL	0	-
SFC_COMPLETED	SFC output signal "COMPLETED"	BOOL	0	-
SFC_ABORTED	SFC output signal "ABORTED"	BOOL	0	-
FB_AV	Feedback for actual value of job in progress	REAL	0.0	-
AS_DT1	Current date/time part 1 (time stamp)	DWORD	16#08123123	-
AS_DT2	Current date/time part 2 (time stamp)	DWORD	16#59590000	-
EDI_RCTL	Create job internal: Name	STRING[32]	''	x
EDI_RCID	Create job internal: ID	DWORD	0	x
EDI_RCPN	Create job internal: Parameter set number	DINT	16#0	x

Parameter	Description	Type	Default	WinCC tag
EDI_RCSP	Create job internal: Setpoint	REAL	0.0	x
EDI_RCDS	Create job internal: Description	STRING[128]	''	x
EDI_RCD1	Create job internal: Time of creation part 1	DWORD	16#08123123	x
EDI_RCD2	Create job internal: Time of creation part 2	DWORD	16#59590000	x
EDI_RCOR	Create job internal: Creator	STRING[32]	''	x
EDI_NORM	Create job internal: Reference quantity	BOOL	0	x
TLIN	Create job internal: Default job name	STRING[32]	''	-
IDIN	Create job internal: Default job ID	DWORD	0	-
ACC_TLIN	Create job internal: Apply default job name = 1	BOOL	0	-
ACC_IDIN	Create job internal: Apply default Job ID = 1	BOOL	0	-
LIOPSELJ	0 = Internal / external switchover via OS 1 = Internal / external switchover via AS	BOOL	0	x
EXT_L	0 = Internal 1 = External	BOOL	0	-
LIOPSELC	0 = Manual / automatic switchover via OS 1 = Manual / automatic switchover via AS	BOOL	0	x
AUT_L	0 = Manual 1 = Automatic	BOOL	0	-
LIOPSELM	0 = One time / cyclic switchover via OS 1 = One time / cyclic switchover via AS	BOOL	0	x
CYCLIC_L	0 = One time mode 1 = Cyclic mode	BOOL	0	-
EXT_IT	Create job external: Default 0 = External interconnected signals, 1 = External IT signals	BOOL	0	-
EXT_RCTL	Create job external, interconnected signals: Name	STRING[32]	''	-
EXT_RCID	Create job external, interconnected signals: ID	DWORD	0	-
EXT_RCPN	Create job external, interconnected signals: Parameter set number	DINT	16#0	-
EXT_RCSP	Create job external, interconnected signals: Setpoint	REAL	0.0	-
EXT_RCDS	Create job external, interconnected signals: Description	STRING[128]	''	--
EXT_RCOR	Create job external, interconnected signals: Creator	STRING[32]	''	-
EXT_NEW	Create job external, interconnected signals: Create new job	BOOL	0	-
EXT_TAOV	Create job external, interconnected signals: Apply job data	BOOL	0	-
EXT_DEL	External job list interconnected signals: Delete job with "EXT_SEL" ID	BOOL	0	-

Parameter	Description	Type	Default	WinCC tag
EXT_ACJO	External job list interconnected signals: Activate first job in job list	BOOL	0	-
EXT_POUP	External job list interconnected signals: Move job with ID "EXT_SEL" up in the list	BOOL	0	-
EXT_PODO	External job list interconnected signals: Move job with ID "EXT_SEL" down in the list	BOOL	0	-
EXT_SEL	External job list interconnected signals: Selection of a job ID from the job list	DWORD	0	-
L_IT	Job control automatic: Default 0 = Automatic interconnected signals, 1 = Automatic IT signals	BOOL	0	-
L_START	Job control automatic, interconnected signals: 1 = Start	BOOL	0	-
L_HOLD	Job control automatic, interconnected signals: 1 = Hold	BOOL	0	-
L_RESUM	Job control automatic, interconnected signals: 1 = Resume	BOOL	0	-
L_END	Job control automatic, interconnected signals: 1 = End	BOOL	0	-
L_ABORT	Job control automatic, interconnected signals: 1 = Abort	BOOL	0	-
IT_RCTL	Create job external IT signals: Name	STRING[32]	''	x
IT_RCID	Create job external IT signals: ID	DWORD	0	x
IT_RCPN	Create job external IT signals: Parameter set number	DINT	16#0	x
IT_RCSP	Create job external IT signals: Setpoint	REAL	0.0	x
IT_RCDS	Create job external IT signals: Description	STRING[128]	''	x
IT_RCOR	Create job external IT signals: Creator	STRING[32]	''	x
IT_NEW	Create job external IT signals: Create new job	BOOL	0	x
IT_TAOV	Create job external IT signals: Job data Apply	BOOL	0	x
IT_DEL	Job list external IT signals: Delete job with "IT_SEL" ID	BOOL	0	x
IT_ACJO	Job list external IT signals: Activate first job in job list	BOOL	0	x
IT_POUP	Job list external IT signals: Move job with ID "IT_SEL" up in the list	BOOL	0	x
IT_PODO	Job list external IT signals: Move job with ID "IT_SEL" down in the list	BOOL	0	x
IT_SEL	Job list external IT signals: Selection of a job ID from the job list	DWORD	0	x
IT_START	Job control automatic IT signals: 1 = Start	BOOL	0	x
IT_HOLD	Job control automatic IT signals: 1 = Hold	BOOL	0	x
IT_RESUM	Job control automatic IT signals: 1 = Resume	BOOL	0	x

Parameter	Description	Type	Default	WinCC tag
IT_END	Job control automatic IT signals: 1 = End	BOOL	0	x
IT_ABORT	Job control automatic IT signals: 1 = Abort	BOOL	0	x
OP_EN	Operator control enables buttons	WORD	16#FFFF	-
HIDE_BUT	Display enables buttons	WORD	0	-
MODE	Function mode	WORD	0	-
USTATUS	User status byte for "QSTATUS"	BYTE	0	-
IMP_T	Pulse time [s] confirmation button message window	REAL	3.0	-
REQ_T	Monitoring time data feedback [s]	REAL	10.0	-
TIME_MON	Monitoring time feedback control commands [s]	REAL	5.0	-
MSGEVID1	ALARM_8P1: EV_ID	DWORD	0	x
MSGEVID2	ALARM_8P2: EV_ID	DWORD	0	x
MSGEVID3	ALARM_8P3: EV_ID	DWORD	0	x
MSGEVID4	ALARM_8P4: EV_ID	DWORD	0	x
MSGEVID5	ALARM_8P5: EV_ID	DWORD	0	x
MSGEVID6	ALARM_8P6: EV_ID	DWORD	0	x
MSGEVID7	ALARM_8P7: EV_ID	DWORD	0	x
PA_ACT	Feedback on parameter set request for job in job interface	UDT_PA_CON	-	-
PA_01 …. PA_10	Feedback on parameter set request for the job with data record 01 to 10	UDT_PA_CON	-	-
"x": WinCC tag available; "-": WinCC tag not available				

Output parameters

The following table shows the output parameters of the "FB_JOB" block:

Parameter	Description	Type	Default	WinCC tag
QRUN_UP	Status display startup active = 1	BOOL	0	-
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	x
QGR_ERR	Function block group error/fault	BOOL	0	x
QERR_VER	1 = Version check error	BOOL	0	x
QCSF	Status display control system error = 1	BOOL	0	x
QMSG_ERR	ALARM_8P: ERROR	BOOL	0	x
MSGSTAT1	ALARM_8P1: STATUS	WORD	0	x
MSG_ACK1	ALARM_8P1: ACK_STATE	WORD	0	x
MSGSTAT2	ALARM_8P2: STATUS	WORD	0	x
MSG_ACK2	ALARM_8P2: ACK_STATE	WORD	0	x
MSGSTAT3	ALARM_8P3: STATUS	WORD	0	x
MSG_ACK3	ALARM_8P3: ACK_STATE	WORD	0	x
MSGSTAT4	ALARM_8P4: STATUS	WORD	0	x
MSG_ACK4	ALARM_8P4: ACK_STATE	WORD	0	x

Parameter	Description	Type	Default	WinCC tag
MSGSTAT5	ALARM_8P5: STATUS	WORD	0	x
MSG_ACK5	ALARM_8P5: ACK_STATE	WORD	0	x
MSGSTAT6	ALARM_8P6: STATUS	WORD	0	x
MSG_ACK6	ALARM_8P6: ACK_STATE	WORD	0	x
MSGSTAT7	ALARM_8P7: STATUS	WORD	0	x
MSG_ACK7	ALARM_8P7: ACK_STATE	WORD	0	x
QJOBSTAT	Job status	DWORD	0	x
QSTATUS	Status double word	DWORD	0	x
QERROR	Error double word	DWORD	0	-
QMSG	Message double word	DWORD	0	-
QRUN_EN	Status display job release available = 1	BOOL	0	x
QEXTINT	Status display 0 = Internal 1 = External	BOOL	0	x
QMAN_AUT	Status display 0 = Manual 1 = Automatic	BOOL	0	x
QRC_EDIT	Status display 1 = Job is being created	BOOL	0	x
QL_IT	Status display job control automatic: 0 = Automatic interconnected signals, 1 = Automatic IT signals	BOOL	0	x
QPA_REQ	Status display Request to FB_XXXPA (normal mode) Read parameter set data, during create job	BOOL	0	x
QPA_COPY	Status display Request to FB_XXXPA (multiplexer mode) Copy parameter set data, during activate job	BOOL	0	x
QCYCLIC	Status display: One time mode = 0, Cyclic mode = 1	BOOL	0	x
QLI_FULL	Status display, job list full = 1	BOOL	0	x
QLI_EMPT	Status display, job list empty = 1	BOOL	0	x
QACC	Create job: Job data were applied = 1	BOOL	0	x
QNOTACC	Create job: Job data were not applied = 1	BOOL	0	x
QVOID	No job in job interface = 1	BOOL	0	x
QREQ_TL	Create job internal: Prompt job name = 1	BOOL	0	x
QREQ_ID	Create job internal: Prompt job ID = 1	BOOL	0	x
QMSBOACK	Message window: Pulse confirmation button (time = "IMP_T")	BOOL	0	x
QJOB_EN	Job control: next job can be activated = 1	BOOL	0	x
QJOB_ON	Job status: Job processing ON = 1	BOOL	0	x
QJOB_ACTIV	Job status: job in progress = 1	BOOL	0	x
QSTART	Job control: Job processing ON = 1	BOOL	0	x
QHOLD	Job control: Job on hold = 1	BOOL	0	x
QRUN	Job control: Job start / resume = 1	BOOL	0	x
QGOEND	Job control: End job = 1	BOOL	0	x

Parameter	Description	Type	Default	WinCC tag
QABORT	Job control: Abort job = 1	BOOL	0	x
SFC_START	Job control, SFC signal "START"	BOOL	0	-
SFC_HOLD	Job control, SFC signal "HOLD"	BOOL	0	-
SFC_RESUME	Job control, SFC signal "RESUME"	BOOL	0	-
SFC_ABORT	Job control, SFC signal "ABORT"	BOOL	0	-
RC_POS01	Data record number at job list item 1	INT	0	x
RC_POS02	Data record number at job list item 2	INT	0	x
RC_POS03	Data record number at job list item 3	INT	0	x
RC_POS04	Data record number at job list item 4	INT	0	x
RC_POS05	Data record number at job list item 5	INT	0	x
RC_POS06	Data record number at job list item 6	INT	0	x
RC_POS07	Data record number at job list item 7	INT	0	x
RC_POS08	Data record number at job list item 8	INT	0	x
RC_POS09	Data record number at job list item 9	INT	0	x
RC_POS10	Data record number at job list item 10	INT	0	x
Q_POS_CN	Status display, number of positions occupied in the job list	INT	0	x
Q_POS_PA	Status display, data record number for the job selected in the job list	INT	0	x
Q_FREEPO	Next free job list item	INT	0	x
Q_FREERC	Next free data record number	INT	0	x
ACT_TITL	Current job in the job interface: Name	STRING[32]	''	x
ACT_ID	Current job in the job interface: ID	DWORD	0	x
ACT_PNO	Current job in the job interface: Parameter set number	DINT	16#0	x
ACT_SP	Current job in the job interface: Setpoint	REAL	0.0	x
ACT_AV	Current job in the job interface: Actual value	REAL	0.0	x
ACT_DESC	Current job in the job interface: Description	STRING[128]	''	x
ACT_BDT1	Current job in the job interface: Start time part 1	DWORD	16#08123123	x
ACT_BDT2	Current job in the job interface: Start time part 2	DWORD	16#59590000	x
ACT_ORIG	Current job in the job interface: Creator	STRING[32]	''	x
ACT_SDT1	Current job in the job interface: Start time part 1	DWORD	16#08123123	x
ACT_SDT2	Current job in the job interface: Start time part 2	DWORD	16#59590000	x
ACT_STAT	Current job in the job interface: Data record status (bit 0 = 1 -> reference quantity)	WORD	0	x
ACTSTATE	Current job in the job interface: Job progress status (normal/abort/end/fault)	WORD	0	x

Parameter	Description	Type	Default	WinCC tag
QSP_UOM	Unit of measurement for the setpoint	STRING[16]	''	x
TIT_P01	Job list item 01: Name	STRING[32]	''	x
ID_P01	Job list item 01: ID	DWORD	0	x
PNO_P01	Job list item 01: Parameter set number	DINT	16#0	x
SP_P01	Job list item 01: Setpoint	REAL	0.0	x
DES_P01	Job list item 01: Description	STRING[128]	''	x
BDT1_P01	Job list item 01: Time of creation part 1	DWORD	16#08123123	x
BDT2_P01	Job list item 01: Time of creation part 2	DWORD	16#59590000	x
ORI_P01	Job list item 01: Creator	STRING[32]	''	x
STAT_P01	Job list item 01: Status data record (bit 0 = 1 -> reference quantity)	WORD	0	x
TIT_P02	Job list item 02: Name	STRING[32]	''	x
ID_P02	Job list item 02: ID	DWORD	0	x
PNO_P02	Job list item 02: Parameter set number	DINT	16#0	x
SP_P02	Job list item 02: Setpoint	REAL	0.1	x
DES_P02	Job list item 02: Description	STRING[128]	''	x
BDT1_P02	Job list item 02: Time of creation part 1	DWORD	16#08123123	x
BDT2_P02	Job list item 02: Time of creation part 2	DWORD	16#59590000	x
ORI_P02	Job list item 02: Creator	STRING[32]	''	x
STAT_P02	Job list item 02: Status data record (bit 0 = 1 -> reference quantity)	WORD	1	x
TIT_P03	Job list item 03: Name	STRING[32]	''	x
ID_P03	Job list item 03: ID	DWORD	0	x
PNO_P03	Job list item 03: Parameter set number	DINT	16#0	x
SP_P03	Job list item 03: Setpoint	REAL	0.2	x
DES_P03	Job list item 03: Description	STRING[128]	''	x
BDT1_P03	Job list item 03: Time of creation part 1	DWORD	16#08123123	x
BDT2_P03	Job list item 03: Time of creation part 2	DWORD	16#59590000	x
ORI_P03	Job list item 03: Creator	STRING[32]	''	x
STAT_P03	Job list item 03: Status data record (bit 0 = 1 -> reference quantity)	WORD	2	x
TIT_P04	Job list item 04: Name	STRING[32]	''	x
ID_P04	Job list item 04: ID	DWORD	0	x
PNO_P04	Job list item 04: Parameter set number	DINT	16#0	x
SP_P04	Job list item 04: Setpoint	REAL	0.3	x
DES_P04	Job list item 04: Description	STRING[128]	''	x
BDT1_P04	Job list item 04: Time of creation part 1	DWORD	16#08123123	x
BDT2_P04	Job list item 04: Time of creation part 2	DWORD	16#59590000	x
ORI_P04	Job list item 04: Creator	STRING[32]	''	x
STAT_P04	Job list item 04: Status data record (bit 0 = 1 -> reference quantity)	WORD	3	x
TIT_P05	Job list item 05: Name	STRING[32]	''	x
ID_P05	Job list item 05: ID	DWORD	0	x

Parameter	Description	Type	Default	WinCC tag
PNO_P05	Job list item 05: Parameter set number	DINT	16#0	x
SP_P05	Job list item 05: Setpoint	REAL	0.4	x
DES_P05	Job list item 05: Description	STRING[128]	''	x
BDT1_P05	Job list item 05: Time of creation part 1	DWORD	16#08123123	x
BDT2_P05	Job list item 05: Time of creation part 2	DWORD	16#59590000	x
ORI_P05	Job list item 05: Creator	STRING[32]	''	x
STAT_P05	Job list item 05: Status data record (bit 0 = 1 -> reference quantity)	WORD	4	x
TIT_P06	Job list item 06: Name	STRING[32]	''	x
ID_P06	Job list item 06: ID	DWORD	0	x
PNO_P06	Job list item 06: Parameter set number	DINT	16#0	x
SP_P06	Job list item 06: Setpoint	REAL	0.5	x
DES_P06	Job list item 06: Description	STRING[128]	''	x
BDT1_P06	Job list item 06: Time of creation part 1	DWORD	16#08123123	x
BDT2_P06	Job list item 06: Time of creation part 2	DWORD	16#59590000	x
ORI_P06	Job list item 06: Creator	STRING[32]	''	x
STAT_P06	Job list item 06: Status data record (bit 0 = 1 -> reference quantity)	WORD	5	x
TIT_P07	Job list item 07: Name	STRING[32]	''	x
ID_P07	Job list item 07: ID	DWORD	0	x
PNO_P07	Job list item 07: Parameter set number	DINT	16#0	x
SP_P07	Job list item 07: Setpoint	REAL	0.6	x
DES_P07	Job list item 07: Description	STRING[128]	''	x
BDT1_P07	Job list item 07: Time of creation part 1	DWORD	16#08123123	x
BDT2_P07	Job list item 07: Time of creation part 2	DWORD	16#59590000	x
ORI_P07	Job list item 07: Creator	STRING[32]	''	x
STAT_P07	Job list item 07: Status data record (bit 0 = 1 -> reference quantity)	WORD	6	x
TIT_P08	Job list item 08: Name	STRING[32]	''	x
ID_P08	Job list item 08: ID	DWORD	0	x
PNO_P08	Job list item 08: Parameter set number	DINT	16#0	x
SP_P08	Job list item 08: Setpoint	REAL	0.7	x
DES_P08	Job list item 08: Description	STRING[128]	''	x
BDT1_P08	Job list item 08: Time of creation part 1	DWORD	16#08123123	x
BDT2_P08	Job list item 08: Time of creation part 2	DWORD	16#59590000	x
ORI_P08	Job list item 08: Creator	STRING[32]	''	x
STAT_P08	Job list item 08: Status data record (bit 0 = 1 -> reference quantity)	WORD	7	x
TIT_P09	Job list item 09: Name	STRING[32]	''	x
ID_P09	Job list item 09: ID	DWORD	0	x
PNO_P09	Job list item 09: Parameter set number	DINT	16#0	x
SP_P09	Job list item 09: Setpoint	REAL	0.8	x
DES_P09	Job list item 09: Description	STRING[128]	''	x

Parameter	Description	Type	Default	WinCC tag
BDT1_P09	Job list item 09: Time of creation part 1	DWORD	16#08123123	x
BDT2_P09	Job list item 09: Time of creation part 2	DWORD	16#59590000	x
ORI_P09	Job list item 09: Creator	STRING[32]	''	x
STAT_P09	Job list item 09: Status data record (bit 0 = 1 -> reference quantity)	WORD	8	x
TIT_P10	Job list item 10: Name	STRING[32]	''	x
ID_P10	Job list item 10: ID	DWORD	0	x
PNO_P10	Job list item 10: Parameter set number	DINT	16#0	x
SP_P10	Job list item 10: Setpoint	REAL	0.9	x
DES_P10	Job list item 10: Description	STRING[128]	''	x
BDT1_P10	Job list item 10: Time of creation part 1	DWORD	16#08123123	x
BDT2_P10	Job list item 10: Time of creation part 2	DWORD	16#59590000	x
ORI_P10	Job list item 10: Creator	STRING[32]	''	x
STAT_P10	Job list item 10: Status data record (bit 0 = 1 -> reference quantity)	WORD	9	x
LAS_TITL	Last job: Name	STRING[32]	''	x
LAS_ID	Last job: ID	DWORD	0	x
LAS_PNO	Last job: Parameter set number	DINT	16#0	x
LAS_SP	Last job: Setpoint	REAL	0.0	x
LAS_DES	Last job: Description	STRING[128]	''	x
LAS_BDT1	Last job: Time of creation part 1	DWORD	16#08123123	x
LAS_BDT2	Last job: Time of creation part 2	DWORD	16#59590000	x
LAS_ORI	Last job: Creator	STRING[32]	''	x
LAS_SDT1	Last job: Start time part 1	DWORD	16#08123123	x
LAS_SDT2	Last job: Start time part 2	DWORD	16#59590000	x
LAS_EDT1	Last job: End time part 1	DWORD	16#08123123	x
LAS_EDT2	Last job: End time part 2	DWORD	16#59590000	x
LAS_AVAL	Last job: Actual value	REAL	0.0	x
LAS_STAT	Last job: Status data record (bit 0 = 1 -> reference quantity)	WORD	0	x
LASSTATE	Last job: Status job history (normal/abort/end/fault)	WORD	0	x
EN_INT	Operator control enable: Internal	BOOL	0	x
EN_EXT	Operator control enable: External	BOOL	0	x
EN_AUT	Operator control enable: Automatic	BOOL	0	x
EN_MAN	Operator control enable: Manual	BOOL	0	x
EN_START	Operator control enable: Start	BOOL	0	x
EN_HOLD	Operator control enable: Hold	BOOL	0	x
EN_RESUM	Operator control enable: Next	BOOL	0	x
EN_END	Operator control enable: End	BOOL	0	x
EN_ABORT	Operator control enable: Abort	BOOL	0	x
EN_CYCL	Operator control enable: Cyclic mode	BOOL	0	x
EN_ONETI	Operator control enable: One time mode	BOOL	0	x

Parameter	Description	Type	Default	WinCC tag
EN_POSUP	Operator control enable: Up	BOOL	0	x
EN_POSDO	Operator control enable: Down	BOOL	0	x
EN_NEW	Operator control enable: Creating a job	BOOL	0	x
EN_DEL	Operator control enable: Deleting a job	BOOL	0	x
EN_ACJO	Operator control enable: Activating a job	BOOL	0	x
EN_TAOV	Operator control enable: Apply	BOOL	0	x
EN_TITL	Operator control enable: Name	BOOL	0	x
EN_ID	Operator control enable: ID	BOOL	0	x
RELSTART	"START" function enable	BOOL	0	-
RELHOLD	"Hold" function enable	BOOL	0	-
RELRESUM	"Resume" function enable	BOOL	0	-
RELEND	"End" function enable	BOOL	0	-
RELABORT	"Abort" function enable	BOOL	0	-
RELPOSUP	"Up" function enable	BOOL	0	-
RELPOSDO	"Down" function enable	BOOL	0	-
RELNEW	"Create job" function enable	BOOL	0	-
RELDEL	"Delete job" function enable	BOOL	0	-
RELACJO	"Activate job" function enable	BOOL	0	-
RELTAOV	"Apply" function enable	BOOL	0	-
DIS_EXIN	Display "Internal/External" button =1	BOOL	0	x
DIS_MAAU	Display "Manual/Automatic" button = 1	BOOL	0	x
DIS_END	Display "End" button = 1	BOOL	0	x
DIS_ABOR	Display "Abort" button = 1	BOOL	0	x
DIS_ACK	Display "Confirmation" button = 1	BOOL	0	x
DIS_MODE	Display "One time / Cyclic" button = 1	BOOL	0	x
DIS_UPDO	Display "Up/Down" button = 1	BOOL	0	x
DIS_EDIT	Display "New" button = 1	BOOL	0	x
DIS_DEL	Display "Delete" button = 1	BOOL	0	x
DIS_ACJO	Display "Activate job" button = 1	BOOL	0	x
QPA_ACT	Parameter set request for job in job interface	UDT_PA_REQ	-	-
QPA_01	Parameter set request for job with data record number 01	UDT_PA_REQ	-	-
QPA_02	Parameter set request for job with data record number 02	UDT_PA_REQ	-	-
QPA_03	Parameter set request for job with data record number 03	UDT_PA_REQ	-	-
QPA_04	Parameter set request for job with data record number 04	UDT_PA_REQ	-	-
QPA_05	Parameter set request for job with data record number 05	UDT_PA_REQ	-	-
QPA_06	Parameter set request for job with data record number 06	UDT_PA_REQ	-	-

Parameter	Description	Type	Default	WinCC tag
QPA_07	Parameter set request for job with data record number 07	UDT_PA_REQ	-	-
QPA_08	Parameter set request for job with data record number 08	UDT_PA_REQ	-	-
QPA_09	Parameter set request for job with data record number 09	UDT_PA_REQ	-	-
QPA_10	Parameter set request for job with data record number 10	UDT_PA_REQ	-	-
"x": WinCC tag available; "-": WinCC tag not available;				

In/out parameters

The following table shows the in/out parameters of the "FB_JOB" block:

Parameter	Description	Type	Default	WinCC tag
OPEXTINT	Operator input: 0 = Internal, 1 = External	BOOL	1	x
OPMANAUT	Operator input: 0 = Manual, 1 = Automatic	BOOL	0	x
OP_CYCLI	Operator input: 0 = One time, 1 = Cyclic	BOOL	0	x
OP_START	Operator input: 1 = Start job	BOOL	0	x
OP_HOLD	Operator input: 1 = Hold job	BOOL	0	x
OP_RESUM	Operator input: 1 = Resume job	BOOL	0	x
OP_GOEND	Operator input: 1 = End job	BOOL	0	x
OP_ABORT	Operator input: 1 = Abort job	BOOL	0	x
OP_NEW	Operator input: 1 = Create new job	BOOL	0	x
OP_TAOV	Operator input: 1 = Apply job data	BOOL	0	x
OP_ESC	Operator input: 1 = Abort job	BOOL	0	x
OP_DEL	Operator input: 1 = Delete job	BOOL	0	x
OP_ACJO	Operator input: 1 = Activate first job	BOOL	0	x
OP_POSUP	Operator input: 1 = Move job up in list	BOOL	0	x
OP_POSDO	Operator input: 1 = Move job down in list	BOOL	0	x
OP_POS	Operator input: Item number of the selected job in the job list	INT	0	x
OPMSBOAC	Operator input: 1 = Confirmation button in message window activated	INT	0	x
APF_AS	APF function block version in AS	WORD	0	x
APF_OS	APF function block version in OS	WORD	0	X
AUX_PR06	ALARM_8P: external associated value SD_6	ANY	-	-
AUX_PR07	ALARM_8P: external associated value SD_7	ANY	-	-
AUX_PR08	ALARM_8P: external associated value SD_8	ANY	-	-
AUX_PR09	ALARM_8P: external associated value SD_9	ANY	-	-

Parameter	Description	Type	Default	WinCC tag
AUX_PR10	ALARM_8P: external associated value SD_10	ANY	-	-
"x": WinCC tag available; "-": WinCC tag not available				

6.1.5 Error handling

Errors occurring in the block are signaled via integrated "ALARM_8P" blocks and displayed at the outputs of the block. The error bits are available at the block as an output signal.

The following error messages can be generated:

Output	Meaning	Cause
QNOT_OK	Block error	"OK" flag of the block = 0
QGR_ERR	Group error / fault	QCSF or QERROR >< 0
QCSF	Control system error	AS input control system error CFC = 1
QMSG_ERR	Group error ALARM_8P blocks	ALARM_8P; .ERROR bit in instance DB
QERROR	Error bits	Refer to table below
QERR_VER	QERR_VER Version check error	Error in the version information

QERROR

The following table shows the error bits of the "QERROR" status word:

QERROR	Symbol	Meaning
Bit 0	MONITORING	1 = Control command feedback error
Bit 8	RUN_EN	1 = AS input "RUN_EN" = 0 while job is running
Bit 16	REQ_QERR	1 = "QERR" from "FB_XXXPA" feedback structure (normal mode) upon apply job
Bit 17	REQ_MON_ERR	1 = "MON_ERR" from "FB_XXXPA" feedback structure (normal mode) upon apply job
Bit 18	REQ_QPARAMF	1 = "QPARAMF" from "FB_XXXPA" feedback structure (normal mode) upon apply job
Bit 19	REQ_QLEN_ERR	1 = "QLEN_ERR" from "FB_XXXPA" feedback structure (normal mode) upon apply job
Bit 20	REQ_QARCHERR	1 = "QARCHERR" from "FB_XXXPA" feedback structure (normal mode) upon apply job
Bit 21	REQ_TIMEMON	1 = Monitoring timeout with apply job
Bit 22	COPY_QERR	1 = "QERR" from "FB_XXXPA" feedback structure (multiplexer mode) or "COPY_TIMEMON" upon activate job
Bit 23	COPY_TIMEMON	1 = Monitoring timeout with activate job
Bit 24	SEL_ID	1 = ID not available in job list (EXTERNAL / IT job delete)
Bit 25	NEW_ID	1 = Job ID <= 0 (create job)
Bit 26	NEW_PSN	1 = Parameter set number <= 0 (create job)
Bit 27	NEW_TITLE	1 = Title was not specified (create job)
Bit 28	NEW_REQ_TL	1 = Error requesting title (create job, internal)
Bit 29	NEW_REQ_ID	1 = Error in job ID request (create job, internal)

6.1.6 Messaging

The "FB_JOB" block uses six "ALARM_8P" blocks to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

ALARM_8P_1

ALARM_8P_1 is used to output fault messages and free external messages:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Input "CSF"	$$BlockComment$$ control system fault
SIG_2		Input "FB_ERROR"	$$BlockComment$$ external fault
SIG_3		None	$$BlockComment$$ A8P_1 SIG3
SIG_4		None	$$BlockComment$$ A8P_1 SIG4
SIG_5		Input "EXT_MSG5"	$$BlockComment$$ external message input EXT_MSG5
SIG_6	AS process control Message - error	Input "EXT_MSG6"	$$BlockComment$$ external message input EXT_MSG6
SIG_7	Process message – with acknowledgment	Input "EXT_MSG7"	$$BlockComment$$ external message input EXT_MSG7
SIG_8	Operating message – without acknowledgment	Input "EXT_MSG8"	$$BlockComment$$ external message input EXT_MSG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	-	
SD_3	-	
SD_4	-	
SD_5	-	
SD_6	@6	AS input "AUX_PR06"
SD_7	@7	AS input "AUX_PR07"
SD_8	@8	AS input "AUX_PR08"
SD_9	@9	AS input "AUX_PR09"
SD_10	@10	AS input "AUX_PR010"

ALARM_8P_2

ALARM_8P_2 is used to output "operating messages" for the job list:

Message	Message class	Event	Message text
SIG_1	Operating message – without acknowledgment	QMSG bit 8 "NEW"	$$BlockComment$$ @4%6s@ job New process cell ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ creator @9%s@
SIG_2		QMSG bit 9 "DEL"	$$BlockComment$$ @4%6s@ job Delete ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ creator @9%s@
SIG_3		QMSG bit 10 "ACTIVE"	$$BlockComment$$ @4%6s@ job Activate ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ creator @9%s@
SIG_4		None	$$BlockComment$$ A8P_2 SIG4
SIG_5		None	$$BlockComment$$ A8P_2 SIG5
SIG_6		None	$$BlockComment$$ A8P_2 SIG6
SIG_7		None	$$BlockComment$$ A8P_2 SIG7
SIG_8		None	$$BlockComment$$ A8P_2 SIG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%s@	INT or EXT or IT
SD_5	@5%d@	Parameter set number
SD_6	@6%g@	Setpoint
SD_7	@7%g@	Actual value
SD_8	@8%s@	Unit of setpoint/actual value (UOM)
SD_9	@9%s@	Creator
SD_10	-	

ALARM_8P_3

ALARM_8P_3 is used to output "operating messages" for job control:

Message	Message class	Event	Message text
SIG_1	Operating message – without acknowledgment	QMSG bit 0 "START"	$$BlockComment$$ @10%6s@ start ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@
SIG_20		QMSG bit 1 "HOLD"	$$BlockComment$$ @10%6s@ hold ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@
SIG_3		QMSG bit 2 "RESUM"	$$BlockComment$$ @10%6s@ resume ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@
SIG_4		QMSG bit 3 "GOEND"	$$BlockComment$$ @10%6s@ end ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@
SIG_5		QMSG bit 4 "ABORT"	$$BlockComment$$ @10%6s@ abort ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@
SIG_6		None	$$BlockComment$$ A8P_3 SIG6
SIG_7		None	$$BlockComment$$ A8P_3 SIG7
SIG_8		None	$$BlockComment$$ A8P_3 SIG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%s@	INT or EXT or IT
SD_5	@5%d@	Parameter set number
SD_6	@6%g@	Setpoint
SD_7	@7%g@	Actual value
SD_8	@8%s@	Unit of setpoint/actual value (UOM)
SD_9	@9%s@	Creator
SD_10	@10%s@	MAN or AUTO or IT

ALARM_8P_4

ALARM_8P_4 is used to output "operating messages" for job control:

Message	Message class	Event	Message text
SIG_1	Operating message – without acknowledgment	State Job active	$$BlockComment$$ job active ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ actual @7%g@@8%s@
SIG_2		State Job paused	$$BlockComment$$ job paused ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ actual @7%g@@8%s@
SIG_3		Status at end of job: Job completed	$$BlockComment$$ job completed ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ actual @7%g@@8%s@
SIG_4		Status at end of job: Job completed after "Go End" preselected	$$BlockComment$$ Job completed after "Go End" preselected ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ actual @7%g@@8%s@
SIG_5		Status at end of job: Job completed after "Abort" preselected	$$BlockComment$$ Job completed after "Abort" preselected ID@3%d@ parameter set @5%d@ setpoint @6%g@@8%s@ actual @7%g@@8%s@
SIG_6		None	$$BlockComment$$ A8P_4 SIG6
SIG_7	AS process control Message - error	Control command feedback monitoring, monitoring timeout	$$BlockComment$$ control command feedback error ID@3%d@ parameter set @5%d@
SIG_8	Operating message – without acknowledgment	QERROR bit 8 "RUN_EN"	$$BlockComment$$ no external job release ID@3%d@ parameter set @5%d@

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%s@	INT or EXT or IT
SD_5	@5%d@	Parameter set number
SD_6	@6%g@	Setpoint
SD_7	@7%g@	Actual value

Associated value	Format	Value
SD_8	@8%s@	Unit of setpoint/actual value (UOM)
SD_9	@9%s@	Creator
SD_10	@10%s@	MAN or AUTO or IT

ALARM_8P_5

ALARM_8P_5 is used to output "error messages" for job management:

Message	Message class	Event	Message text
SIG_1	AS process control Message - error	QERROR bit 24 "SEL_ID"	$$BlockComment$$ @4%6s@ Delete job not possible ID@2%d@ does not exist
SIG_2		QERROR bit 25 "NEW_ID"	$$BlockComment$$ @4%6s@ Job data cannot be applied, ID = 0
SIG_3		QERROR bit 26 "NEW_PSN"	$$BlockComment$$ @4%6s@ Job data could not be applied, no parameter set number was specified
SIG_4		QERROR bit 27 "NEW_TITLE"	$$BlockComment$$ @4%6s@ Job data could not be applied, no title was specified
SIG_5		QERROR bit 28 "NEW_REQ_TL"	$$BlockComment$$ @4%6s@ Autom. spec. / application of title failed upon creation of new job
SIG_6		QERROR bit 29 "NEW_REQ_ID"	Autom. spec. / application of ID failed upon creation of new job
SIG_7		None	$$BlockComment$$ A8P_5 SIG7
SIG_8		None	$$BlockComment$$ A8P_5 SIG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	Selected job ID
SD_3	-	
SD_4	@4%s@	INT or EXT or IT
SD_5	-	
SD_6	-	

Associated value	Format	Value
SD_7	-	
SD_8	-	
SD_9	-	
SD_10	-	

ALARM_8P_6

ALARM_8P_6 is used to output "error messages" for job control:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	QERROR bit 16 "REQ_QERR"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, error feedback "QERR" (ID@3%d@)
SIG_2		QERROR bit 17 "REQ_MON_ERR"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, error feedback "MON_ERR" (ID@3%d@)
SIG_3		QERROR bit 18 "REQ_QPARAMF"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, error feedback "QPARAMF" (ID@3%d@)
SIG_4		QERROR bit 19 "REQ_QLEN_ERR"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, error feedback "QLEN_ERR" (ID@3%d@)
SIG_5		QERROR bit 20 "REQ_QARCHERR"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, error feedback "QARCHERR" (ID@3%d@)
SIG_6		QERROR bit 21 "REQ_TIMEMON"	$$BlockComment$$ @4%6s@ Req. parameter @2%d@, No feedback (ID@3%d@)
SIG_7		QERROR bit 22 "COPY_QERR"	$$BlockComment$$ @4%6s@ Copy data record @2%d@ to job mailbox, error "QERR" (ID@5%d@/ACTIVE ID@7%d@)
SIG_8		QERROR bit 23 "COPY_TIMEMON"	$$BlockComment$$ @4%6s@ Copy data record @2%d@ to job mailbox, no feedback (ID@5%d@/ACTIVE ID@7%d@)

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	Parameter set number with "Create job"
SD_3	@3%d@	ID with "Create job"
SD_4	@4%s@	INT or EXT or IT
SD_5	@5%d@	ID with "Activate job"
SD_6	@6%d@	Parameter set number with "Activate job"
SD_7	@7%d@	Active ID with "Activate job"
SD_8	-	
SD_9	-	
SD_10	-	

ALARM_8P_7

ALARM_8P_7 is used to output "error messages" for job control:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER Error in version check
SIG_2		None	$$BlockComment$$ A8P_7 SIG2
SIG_3		None	$$BlockComment$$ A8P_7 SIG3
SIG_4		None	$$BlockComment$$ A8P_7 SIG4
SIG_5		None	$$BlockComment$$ A8P_7 SIG5
SIG_6		None	$$BlockComment$$ A8P_7 SIG6
SIG_7		None	$$BlockComment$$ A8P_7 SIG7
SIG_8		None	$$BlockComment$$ A8P_7 SIG8

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$ source
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	Parameter set number with "Create job"
SD_3	@3%d@	ID with "Create job"
SD_4	@4%s@	INT or EXT or IT
SD_5	@5%d@	ID with "Activate job"

Associated value	Format	Value
SD_6	@6%d@	Parameter set number with "Activate job"
SD_7	@7%d@	Active ID with "Activate job"
SD_8	-	
SD_9	-	
SD_10	-	

6.2 User archives

The parameter management user archives are used to access the parameter sets.

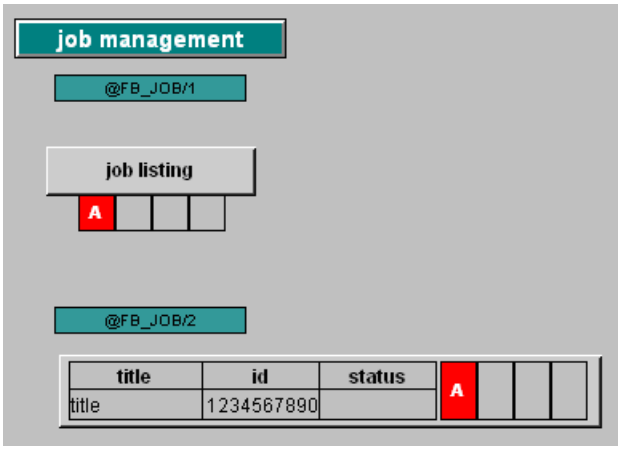
For more information, refer to the section Parameter management (Page 143).

6.3 Operator control and monitoring

6.3.1 Block icons

@TypicalsAPF.pdl

The "@TypicalsAPF.pdl" template picture contains templates for automatically generating block icons from the plant hierarchy:

Block	Template
FB_JOB	

@Template_APF.pdl

The "@Template_APF.pdl" template picture contains additional templates for configuring the block icons in the process pictures.

6.3.2 Picture objects

The operation and monitoring of job management is performed in a faceplate that uses the following pictures for the various views:

- @PG_FB_JOB_STANDARD.PDL
  This picture is the operator control and monitoring interface for job control. In this picture, you select "Cyclic" or "One time" and "Manual" or "Auto" mode.
- @PG_FB_JOB_PARAMETERS.PDL
  The picture is the operator control and monitoring interface for managing the job list. In this picture, you select the "Internal" or "External" interface for job input.
- @PCS7_ALARM.PDL
  This picture is the interface to the message system.
- @PL_FB_JOB.PDL
  The "@PL_FB_JOB.PDL" picture is the advanced faceplate view.

The following windows are also available:

- @FB_MessageBox.PDL
  The help window is used to display a message text in the standard faceplate view.
- @FB_String.PDL
  This picture is used to enter text in the faceplate.
- @FB_BinOp.PDL
  This picture is used in 2-step faceplate operation.

Note

The process pictures are pre-configured and do not need to be edited.

For detailed information on interface operator control and monitoring in process mode, see the APF Operator Manual.

6.3.3 Messages

Operator messages

The following messages can be output:

Message class	Event	Offset	Message text
Operator message	Apply button	0	INT job create new ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Delete button	1	INT job delete ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Activate job button	2	INT job activate ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Button	3	
	Button	4	
	Button	5	
	Start button	6	MANUAL start ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Hold button	7	MANUAL hold ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Resume button	8	MANUAL resume ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	End button	9	MANUAL end ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Abort button	10	MANUAL abort ID@4%d@ parameter set @9%s@ setpoint @5%g@@8%s@
	Button	11	
	Button	12	
	Button	13	
	Button	14	
	Auto button	15	Switchover to automatic mode
	Manual button	16	Switchover to manual mode
	Internal button	17	Switchover to internal mode
	External button	18	Switchover to external mode
	One time button	19	Switchover to one time mode
	Cyclic button	20	Switchover to cyclic mode
	Button	21	
	Button	22	

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%s@	BA_NA
Message associated value 2	@2%@	*Not used*
Message associated value 3	@3%@	*Not used*
Message associated value 4	@4%d@	ID

Message associated value	Format	Value
Message associated value 5	@5%g@	Setpoint
Message associated value 6	@6%@	*Not used*
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	UOM
Message associated value 9	@9%s@	Parameter set
Message associated value 10	@10%s@	Source (UNIT)

For additional information, refer to the PCS 7 online help.

Script messages

The following messages can be output:

Message class	Event	Offset	Message text
Operator message	Script	0	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%s@	*Not used*
Message associated value 2	@2%@	*Not used*
Message associated value 3	@3%d@	Block ID of the FB
Message associated value 4	@4%d@	REC ID
Message associated value 5	@5%d@	Error number
Message associated value 6	@6%@	*Not used*
Message associated value 7	@7%s@	*Not used*
Message associated value 8	@8%s@	*Not used*
Message associated value 9	@9%s@	*Not used*
Message associated value 10	@10%s@	

For additional information, refer to the PCS 7 online help.

Archive management 7

7.1 FB_XXXA (FB_XXXAI, FB_XXXAR)

7.1.1 Description

Object properties

The following table shows the object properties of FB_XXXAD:

Name	Number	Family	Module
FB_XXXA	FB1793	APF	Archive management

Area of application

The "FB_XXXA" block is used to archive data records from the AS. You can add new data records to the user archive and update or delete existing data records.

On the basis of the "FB_XXXA" block, the APF Engineering Tool generates the "FB_XXXAI" and "FB_XXXAR" blocks for displaying format-related numerical values.

Startup characteristics

The "FB_XXXA" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Data exchange with the "FB_UA_RM" communication block on the AS level is performed via the "UDT_UA_R_REQ_ST" (request) and "UDT_UA_R_ST" (feedback) structures.

The "ALARM_8P" structure is used to implement data communication with the OS message system.

The archiving data at the block output are available in the OS as WinCC tags.

7.1.2 Generation

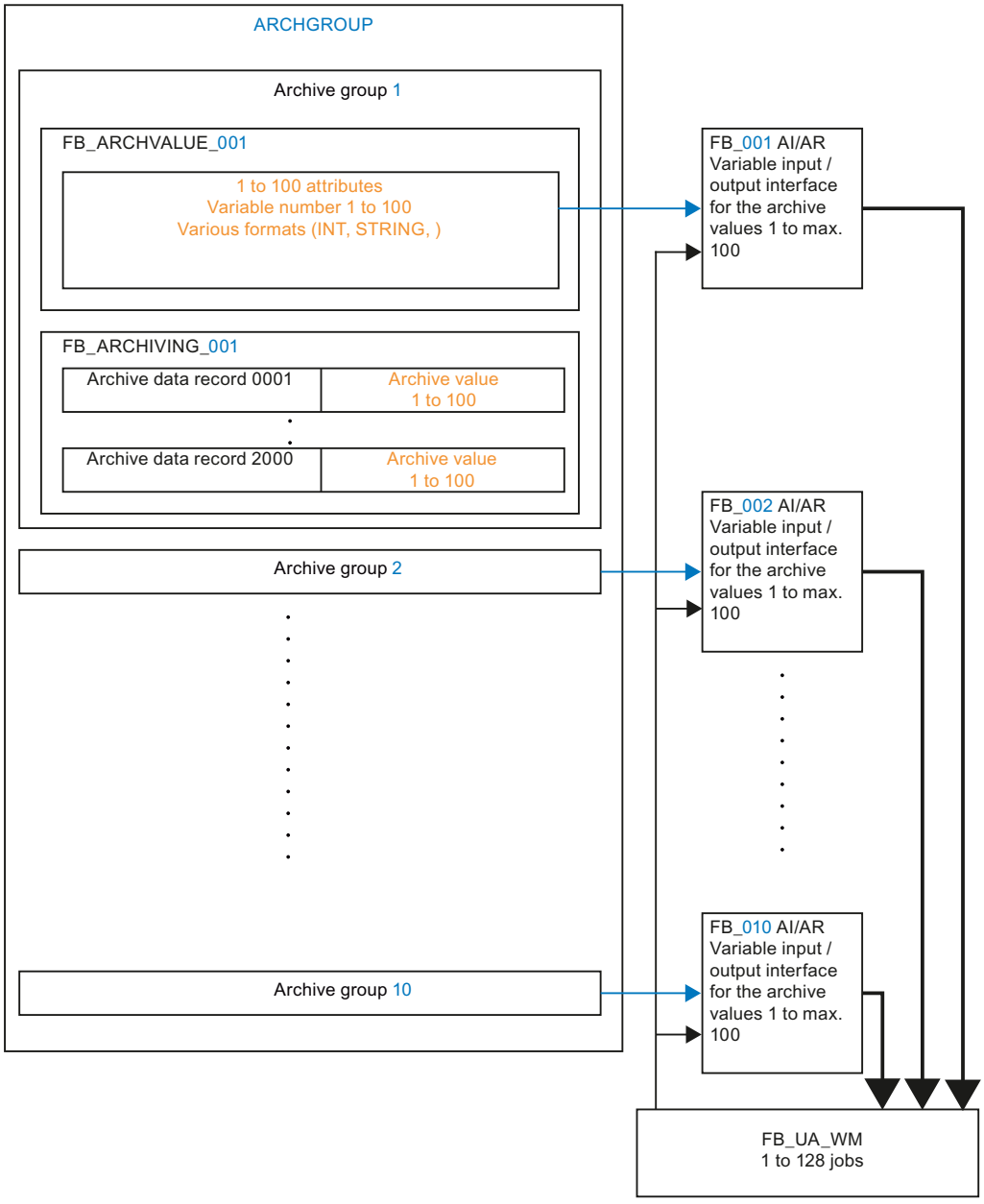
Using the APF Engineering Tool, the following blocks are generated on the basis of the "FB_XXXA" (FB1793) block for each archive group created:

Function block (FB)	Number	Function
FB_XXXAI	Specified by the "FB Number (I)" field of the APF engineering tool	Creates, updates, and deletes archive data records of format "INTEGER (32)" (for format-related numerical values)
FB_XXXAR	Specified by the "FB Number (R)" field of the APF engineering tool	Creates, updates, and deletes archive data records of format "REAL" (for format-related numerical values)

The "FB_XXXAI" and "FB_XXXAR" blocks differ in their display of format-related numerical values for FORMAT= 54 and 70. These values are displayed as INTEGER (32-bit) in FB_XXXAI and as REAL in FB_XXXAR. The two blocks write to the same archive (as DOUBLE 64-bit).

The input and output parameters of the function blocks depend on the structure of the "FB_ARCHIVING_XXX" user archive. The blocks are numbered consecutively in line with the user archives.

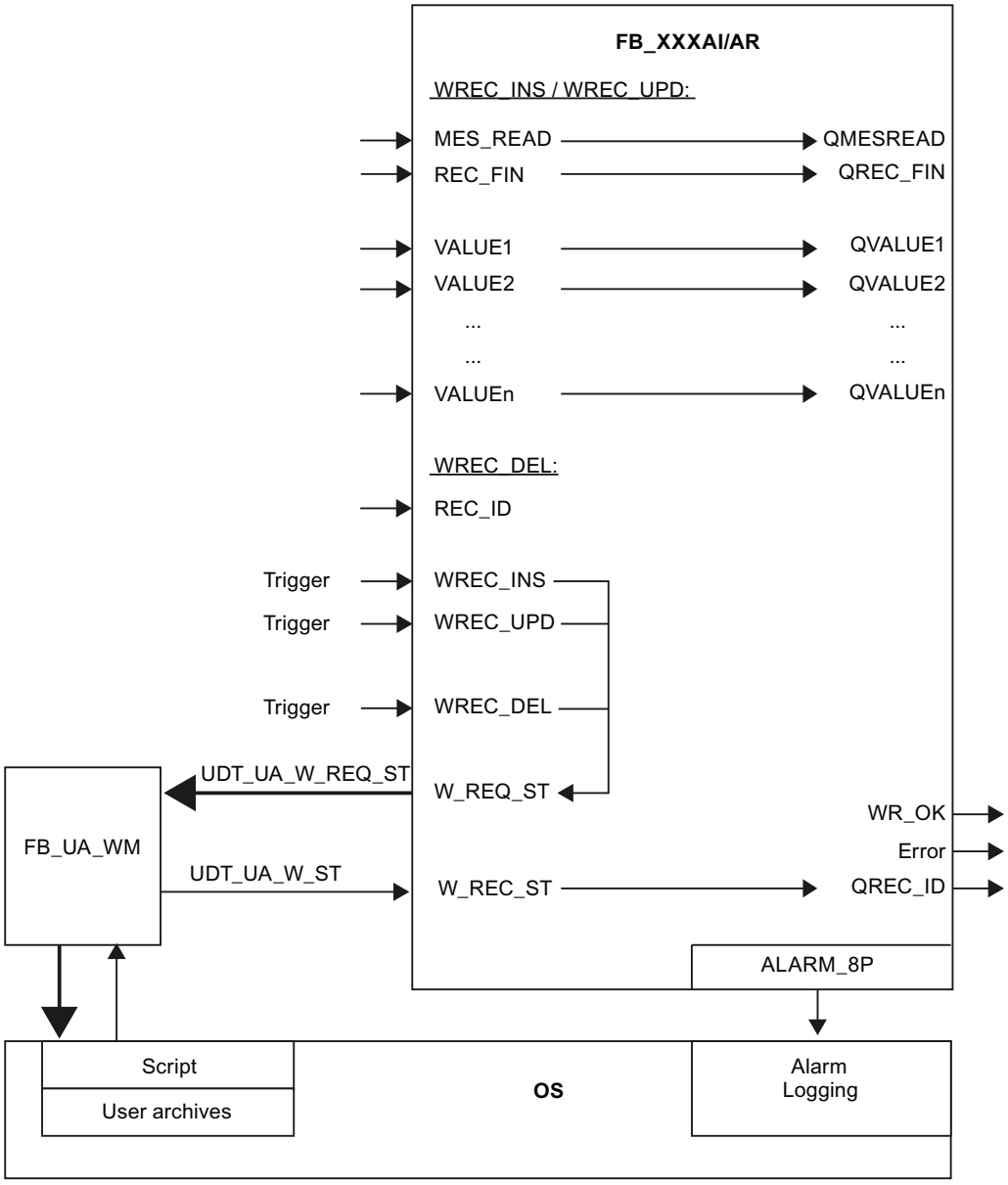
The figure below provides an overview of the "FB_XXXA" block structure:


ARCHGROUP
Archive group 1
FB_ARCHVALUE_001
1 to 100 attributes
Variable number 1 to 100
Various formats (INT, STRING,)
FB_ARCHIVING_001
Archive data record 0001
Archive value
1 to 100
Archive data record 2000
Archive value
1 to 100
Archive group 2
Archive group 10
FB_001 AI/AR
Variable input / output interface for the archive values 1 to max. 100
FB_002 AI/AR
Variable input / output interface for the archive values 1 to max. 100
FB_010 AI/AR
Variable input / output interface for the archive values 1 to max. 100
FB_UA_WM
1 to 128 jobs

7.1.3 Functions

Overview

The figure below gives an overview of how the "FB_XXXA" block works:



Adding data

Following a request by a positive edge at block input "WREC_INS", the system writes the block input values for archiving to the user archive on the basis of the archive name, archive number, etc. The values are simultaneously copied to corresponding block outputs to store the values written in the clipboard until the next write operation.

The value "REC_ID" must be unique for each archive for its entire life cycle.

In "Insert" mode, a new unique data record number REC_ID (REC_ID_MAX + 1) is automatically generated on the OS. This number is returned to the requesting block at the end of the write job by "FB_UA_WM" feedback messages. The data record number "REC_ID" is output at the "QREC_ID" block output for further processing. This "REC_ID" is needed for deleting and updating the archive data record.

The newly generated "REC_ID" is written to the value in the "REC_ID_MAX" column of the "FB_ARCHGROUP" archive once the new archive data record has been successfully created.

Updating data

Upon a request by a positive edge at the "WREC_UPD" input, the values of the relevant block inputs for archiving are written to the user archive on the basis of the archive data record ID (REC_ID), archive name, archive number, etc. The values are simultaneously copied to the corresponding block outputs to save the written values on the clipboard until the next write operation.

Deleting data

The options for deleting data are as follows:

- With MES and "MES_READ = 1"
  Following a request by a positive edge at the "WREC_DEL" input, the system uses the archive data record ID (REC_ID), archive name, archive number, etc. to set the "REC_PREPARED := 2" column in the relevant archive data record of the user archive and thus select it for an MES. The values of the corresponding outputs at the function block are deleted.
  If the MES evaluates this information, it sets the value "REC_READ := 1". A global time-triggered C function (FB_DeleteMarkedRecordsTrigger.pas) in WinCC checks all data records in the user archive with the filter condition "(REC_EXPORTED = 1) AND (REC_PREPARED = 2) AND (REC_READ = 1)" and deletes these records.
- Without MES and "MES_READ = 0"
  Following a request by a positive edge at the "WREC_DEL" input, the system uses the archive data record ID (REC_ID), archive name, archive number, etc. to delete the corresponding archive data record from the user archive. The values of the corresponding outputs at the function block are deleted.

Duration of request

The request inputs WREC_INS, WREC_UPD and WREC_DEL must remain set until the request has been processed in full, which means until the feedback message WR_OK for successful processing has been set or QGR_ERR if an error occurs. Resetting the input sooner aborts the request.

Repetition of jobs

The "COUNT" input value defines how often a job is to be repeated if an error occurs (COUNT = 0 -> no repetition). Until the specified number of job repetitions has been reached, the job is reactivated after a wait cycle.

An error message is output after the last repetition.

Exception: The job is not repeated after inserting (WREQ_INS).

7.1.4 I/Os

Input parameters

The following table shows the input parameters of the "FB_XXXA" block:

Parameter	Description	Type	Default	WinCC tag
SAMPLE_T	Sampling time [s]	REAL	1	-
RUNUPCYC	Number of startup cycles	INT	10	-
ID	ID of the block	INT	0	-
ARCH_NA	Archive name	STRING[16]	''	-
ARCH_NO	Archive number	DINT	0	-
REC_ID	Record number in the user archive	DINT	0	-
BA_NA	Batch name	STRING[32]	''	-
BA_ID	Batch ID	DWORD	0	-
COUNT	Number of repetitions if an error occurs; Exception: The job is not repeated after inserting (WREQ_INS).	INT	1	-
TIME_MON	Monitoring time for write job	REAL	30.0	-
W_REC_ST	Feedback structure from "FB_UA_WM"	UDT_UA_W_ST		-
WREC_INS	Trigger Insert data record	BOOL	0	-
WREC_DEL	Trigger Delete data record	BOOL	0	-
WREC_UPD	Trigger Update data record	BOOL	0	-
RESET	1 = Reset write interface	BOOL	0	-
MODE	Currently not used (operating mode of the block)	WORD	0	-
MES_READ	1 = Release for MES system for reading data 0 = No read release for MES system for reading data	BOOL	0	-
REC_FIN	1 = Recording / changes to data record completed 0 = Archive data record recording / change	BOOL	0	-
VALUE1	Example: Archive value 1	DINT	0	-
VALUE2	Example: Archive value 2	REAL	0.0	-
VALUE3	Example: Archive value 3	STRING[16]	''	-
VALUE4	Example: Archive value 4 part A	DWORD	16#08123123	-
VALUE4B	Example: Archive value 4 part B	DWORD	16#59590000	-
VALUE5	Example: Archive value 5	BOOL	0	-
VALUE6	Example: Archive value 6	DINT	0	-

Parameter	Description	Type	Default	WinCC tag
VALUE7	Example: Archive value 7	DINT	0	-
VALUE8	Example: Archive value 8	DINT	0	
VALUE9	Example: Archive value 9	DINT	0	-
MSG_EVID	ALARM_8P option: EV_ID	DWORD	0	-
"x": WinCC tag available; "-": WinCC tag not available;				

Output parameters

The following table shows the output parameters of the "FB_XXXA" block:

Parameter	Description	Type	Default	WinCC tag
QREC_W	Write request status 0 = Free 1 = Write request active 2 = Write request completed 3 = Write request completed with errors 4 = Write request aborted	INT	0	-
QRUN_UP	Status display startup active = 1	BOOL	0	-
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0	-
QRELEASED	1= Function block released for processing	BOOL	1	x
QGR_ERR	Function block group error/fault	BOOL	0	-
QERR_VER	1 = Version check error	BOOL	0	x
QPARAMF	1 = Parameter assignment error, ID=0 or ARCH_NA=' ' or ARCH_NO=0 or TIME_MON=0	BOOL	0	-
QERR	1 = Write error, data header: requested data and written data do not match	BOOL	0	-
QMONERR	1 = Write job monitoring timeout	BOOL	0	-
QRCIDERR	1 = Error "REC_ID = 0"	BOOL	0	-
QWM_ERR	FB_UA_WM: 1 = Write error, requested data and written data do not match	BOOL	0	-
QWM_AERR	FB_UA_RM: 1 = Archive error, invalid data in send buffer	BOOL	0	-
QWM_CERR	FB_UA_RM: 1 = Communication error, error message from "BSEND" coupling blocks	BOOL	0	-
QWM_PERR	FB_UA_WM: 1 = Parameter assignment error, duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	0	-
QINS_ERR	Error: 1 = Cannot insert,	BOOL	0	-
QUPD_ERR	Error: 1 = Cannot update, data record does not exist	BOOL	0	-
QDEL_ERR	Error: 1 = Cannot delete, data record does not exist	BOOL	0	-
QBADAERR	Error: 1 = Batch data invalid, during inserting / updating	BOOL	0	-
QREQ_ERR	Error: 1 = More than one action (insert/update/ delete) triggered in same cycle	BOOL	0	-

Parameter	Description	Type	Default	WinCC tag
QIN_PROC	Error: 1 = Action not possible because an action is already active (insert/update/delete)	BOOL	0	-
QREQ_WM	Write job active	BOOL	0	-
WR_OK	Data have been written	BOOL	0	-
W_REQ_ST	Request structure to FB_UA_WM	UDT_UA_W_REQ_ST		-
QREQ_ID	Write job: ID	INT	0	-
QMODE	Write job: Mode	WORD	0	-
QARCH_NA	Write job: Archive name	STRING[16]	''	-
QARCH_NO	Write job: Archive number	DINT	0	-
QREC_ID	Write job: Data record number in user archive	DINT	0	-
QNO_REC	Write job: Number of data records	DINT	0	-
QCOLUMN	Write job: Column number	DINT	0	-
QNO_COL	Write job: Number of columns	DINT	0	-
QWM_ID	Write job: ID of the FB_UA_WM function block	INT	0	-
QBOOL	Write job: Binary auxiliary data of the feedback structure	WORD	0	-
QMESREAD	Release for the MES system for reading data	DINT	0	-
QREC_FIN	Recording / changes to data record completed	BOOL	0	-
QBA_NA	Batch name	STRING[32]	''	-
QBA_ID	Batch ID	DWORD	0	-
QVALUE1	Example: Display archive value 1	DINT	0	-
QVALUE2	Example: Display archive value 2	REAL	0.0	-
QVALUE3	Example: Display archive value 3	STRING[16]	''	-
QVALUE4	Example: Display archive value 4 part A	DWORD	16#08123123	-
QVALUE4B	Example: Display archive value 4 part B	DWORD	16#59590000	-
QVALUE5	Example: Display archive value 5	BOOL	0	-
QVALUE6	Example: Display archive value 6	DINT	0	-
QVALUE7	Example: Display archive value 7	DINT	0	-
QVALUE8	Example: Display archive value 8	DINT	0	-
QVALUE9	Example: Display archive value 9	DINT	0	-
ALARM_ST	ALARM structure for connection of an external message block	UDT_ALARM_ST		-
MSG_STAT	ALARM_8P: STATUS	WORD	0	-
MSG_ACK	ALARM_8P: ACK_STATE	WORD	0	-
QMSG_SUP	ALARM_8P: 1=Message suppression active	BOOL	BOOL	-
QMSG_ERR	ALARM_8P: ERROR	BOOL	BOOL	-
"x": WinCC tag available; "-": WinCC tag not available				

In/out parameters

The following table shows the in/out parameters of the "FB_XXXA" block:

Parameter	Description	Type	Default	WinCC tag
APF_AS	APF function block version in AS	WORD	0	x
APF_OS	APF function block version in OS	WORD	0	x
GEN_AS	Generation information in AS	DWORD	0	x
GEN_OS	Generation information in OS	DWORD	0	x
"x": WinCC tag available; "-": WinCC tag not available				

7.1.5 Error handling

Block error

Errors occurring in the block are signaled via an integrated "ALARM_8P" block and displayed at the digital outputs of the block. They can be analyzed by the requesting block on the basis of the output structure. The error bits are available at the block as output signals. In this case, the "WR_OK" output is not set.

The following error messages can be generated:

Output structure	Output	Meaning
	QPARAMF	Parameter assignment error; ID, ARCH_NO or TIME_MON <= 0, or ARCH_NA="
	QERR	Data error REQ_ID in request and receive structure not the same
MON_ERR	QMONERR	Runtime error; Monitoring timeout
	QRCIDERR	Specification error; REC_ID <= 0
	QWM_ERR	Error from FB_UA_W: Data error, job data in WinCC communication interface do not correspond to original job data
	QWM_AERR	Error from FB_UA_W: Archive error; job data in the WinCC communication interface reset by the OS, or script error occurred in OS
	QWM_CERR	Error from FB_UA_W: Communication error; Error message from "BSEND" coupling blocks
	QWM_PERR	Error from FB_UA_W: Parameter assignment error; incorrect ID_1, R_ID_1 or ID_2, R_ID_2 or duplicate REQ_ID
	QERR_VER	Error in the version information

Archive errors

In contrast to normal troubleshooting, a special error output and the setting of the "WR_OK" output indicate those errors occurring during operation which are not related to communication between FB_XXXA / FB_UA_WM / OS script.

Error bit	Output	Meaning
Bit 0	QINS_ERR	Error: Cannot insert
Bit 1	QUPD_ERR	Error: Cannot update, data record does not exist
Bit 2	QDEL_ERR	Error: Cannot delete, data record does not exist
Bit 3	QBADAERR	Error: Batch data invalid, during inserting / updating (only with user archive FB_ARCHIVING_XXX Type 1) The following block inputs must be usefully assigned: BA_ID (> 0) BA_NA (!= empty string) TIME_START (correct date / time) TIME_END (correct date / time)

7.1.6 Messaging

The "FB_XXXA" block uses the "ALARM_8P" block to generate messages. Following CPU startup, messages are suppressed (QMSG_SUP = 1) until the "RUNUPCYC" cycles have elapsed.

The version error will additionally be reported through the ALARM_8P (SIG_6).

The following messages can be output:

Message	Message class	Event	Message text
SIG_1	AS process control Message - fault	QINS_ERR or QUPD_ERR or QDEL_ERR or QBADAERR	$$BlockComment$$ error @7%d@ in archiving function INS/UPD/DEL ID @4%d@ REC_ID @5%d@
SIG_2	AS process control Message - fault	Write job monitoring timeout	$$BlockComment$$ QMON_ERR runtime error write job ID @4%d@ REC_ID @5%d@
SIG_3	AS process control Message - fault	requested data and read data do not match	$$BlockComment$$ QERR data error write job ID @4%d@ REC_ID @5%d@
SIG_4	AS process control Message - error	Parameter assignment error, ID=0 or ARCH_NA=' ' or ARCH_NO=0 or TIME_MON=0	$$BlockComment$$ QPARAMF parameter assignment error ID @4%d@ REC_ID @5%d@
SIG_5	AS process control Message - error	REC_ID <= 0 during deletion / update	$$BlockComment$$ REC_ID error ID @4%d@ REC_ID <= 0
SIG_6	AS process control Message - fault	Error in the version information	$$BlockComment$$ QERR_VER error in version check

Message	Message class	Event	Message text
SIG_7	AS process control Message - fault	More than one trigger in a cycle (WREC_INS / WREC_DEL / WREC_UPD);	$$BlockComment$$ QREQ_ERR, more than one trigger of function active ID @4%d@ REC_ID @5%d@
SIG_8	AS process control Message - fault	Trigger (WREC_INS / WREC_DEL / WREC_UPD) not valid: WREC_INS, WREC_DEL or WREC_UPD action still active;	$$BlockComment$$ QIN_PROC, a write job is already active ID @4%d

The following keywords are used:

- $$BlockComment$$: instance-specific block comment.
- $$HID$$: Origin
- $$AREA$$: OS_AREA
- @1%s@: Batch ID

The following associated values are used:

Associated value	Format	Value
SD_1	@1%s@	BA_NA
SD_2	@2%d@	STEP_NO
SD_3	@3%d@	BA_ID
SD_4	@4%d@	ID
SD_5	@5%d@	REC_ID
SD_6	-	
SD_7	@7%d@	Error code
SD_8	-	
SD_9	-	
SD_10	-	

7.2 C functions for export and backup

7.2.1 Overview

C functions are available for export and data backup processes. These functions perform export and data backup using specific transfer parameters. Processes are triggered with global C functions which can be configured using the APF Engineering Tool.

The following C functions are used as a template and copied and adapted by the APF Engineering Tool:

- "FB_ARExportTrigger0.pas"
  Triggers export of the data according to the specified method
- "FB_ARBackupTriggerEN0.pas"
  Triggers data backup for the engineering/configuration data
- "FB_ARBackupTriggerRT0.pas"
  Triggers data backup for runtime data

All C functions are adapted by the APF Engineering Tool for each archive group. For example, archive group 3 is allocated the corresponding C functions "FB_ARExportTrigger3.pas" and "FB_ARBackupTriggerRT3.pas". These functions are then executed at the intervals set for the relevant archive group.

There is also a global C action "FB_DeleteMarkedRecordsTrigger.pas" which is executed hourly and, if there is an MES system, deletes all data records marked as deleted from all archives.

We recommend that you configure the individual functions using the APF Engineering Tool.

7.2.2 Data export

Two different methods are available for data export. Three flags are available in the archive for these methods:

- REC_READ: Data record read from connected MES or similar system
- REC_EXPORTED: Data record exported by internal C export function to a csv file
- REC_FINISHED: Recording / changing data record by AS completed

Using the engineering tool, you also specify whether a connected MES system or similar system exists and reads data records ("mes_used").

Each archive has a corresponding C function (template: "FB_ARExportTrigger0.pas") that is time-triggered and has the following functionalities.

The various export methods are distinguished as follows, and are defined separately for each archive in the "EXPORT_METHOD" column of the "FB_ARCHGROUP" archive (numbers 1… 6 correspond to the export method):

1. Data export with deletion of data and MES connection (check MES value in the ini file)
   - Data are exported by the time-triggered C function if:
     REC_READ = 1 (only queried if the MES flag in the ini file is activated)
     REC_EXPORTED = 0
     REC_FINISHED = 1
   - Following successful export, the data records exported are deleted from the archive.
   - Data are deleted if the corresponding archive is too full following the export. This should be done using the method detailed in "Deleting data at maximum archive capacity".
2. Data export with deletion of data without an MES connection (check MES value in the ini file)
   - Data are exported by the time-triggered C function if:
     REC_EXPORTED = 0
     REC_FINISHED = 1
   - Following successful export, the data records exported are deleted from the archive. If the archive still contains data records which were previously exported, these are also deleted.
   - Data are deleted if the corresponding archive is too full following the export. This should be done using the method detailed in "Deleting data at maximum archive capacity".
3. Data export without deletion of data and with an MES connection (check MES value in the ini file)
   - Data are exported by the time-triggered C function if:
     REC_READ = 1 (only queried if the MES flag in the ini file is activated)
     REC_EXPORTED = 0
     REC_FINISHED = 1
   - Following successful export, the following flags in the archive are set:
     REC_EXPORTED := 1
   - Data are only deleted if the corresponding archive is full. This is done using the method detailed in "Deleting data at maximum archive capacity".
4. Data export without deletion of data and without an MES connection (check MES value in the ini file)
   - Data are exported by the time-triggered C function if:
     REC_EXPORTED = 0
     REC_FINISHED = 1
   - Following successful export, the following flags in the archive are set:
     REC_EXPORTED := 1
   - Data are only deleted if the corresponding archive is full. This is done using the method detailed in "Deleting data at maximum archive capacity".

5. Data management as circular archive
   - Data in the archive are not archived externally and are only available during runtime.
   - If the archive is full (data records in the archive >= FULL_NO_OF_REC), the C function "FB_ARExportTrigger0.pas" deletes (without exporting the data) the n data records (n := relevant cell in "DEL_NO_OF_REC" column in the "FB_ARCHGROUP" archive) with the smallest REC_ID (the n oldest data records).
6. Deleting of data with maximum archive size
   If the archive is full (data records in the archive >= FULL_NO_OF_REC), the C function "FB_ARExportTrigger0.pas" tries (without exporting the data) to delete the following n data records one after the other (n := relevant cell in "DEL_NO_OF_REC" column in the "FB_ARCHGROUP" archive):
   - Step 1: x_1 data records (x_1<=n) with the following properties:
     1. The smallest REC_IDs
     2. Flag REC_FINISHED = 1
     3. If applicable: Flag REC_READ = 1 (check MES value in the ini file)
     4. Flag REC_EXPORTED = 1
   - Step 2: Only performed if you are using an MES. If step 1 does not cover enough data records, the following x_2 data records (x_2 <= n – data records deleted in step 1 x_2) are also deleted:
     1. The smallest REC_IDs
     2. Flag REC_FINISHED = 1
     3. Flag REC_READ = 1
   - Step 3: If the previous attempts do not cover enough data records, x_3 data records (x_3 <= n – data records deleted in step 1 x_1 - in data records deleted in step 2 x_2) are also deleted:
     1. The smallest REC_IDs
     2. Flag REC_FINISHED = 1
   - Step 4: If there are still not enough data records, x_4 data records (x_4 <= n – data records deleted in step 1 x_1 - data records deleted in step 2 x_2- data records deleted in step 3 x_3) are also deleted:
     1. The smallest REC_IDs

Access using an MES or similar system (without deleting the data records read)

Once a data record has been successfully read, the MES system must set the following flags in the archive:

REC_READ := 1

Access from AS

Once a data record has been successfully inserted / updated, the AS must set the following flags in the archive:

REC_READ := 0

REC_EXPORTED = 0

Once a data record has been successfully inserted / updated, the AS can set the following flags in the archive:

REC_FINISHED = 1

Naming convention for the export files / backup files

- Export methods 1-4:
  <Year><Month><Day>_<Time>_<Archive name>
  Example: 091224_183059_Archiving_004
- Data backup:
  sav_<Year><Month><Day>_<Time>_<Archive name>
  Example: sav_091224_183059_Archiving_004

Export file redundancy

There are two backup paths (engineering tool) and two export paths (specified separately in the group archive for each group). The exported export / backup files (csv format) are copied to both paths.

OS server redundancy

In a redundant server pair, data backup is only executed from the master currently active. The two backup paths mean that redundant data backup is guaranteed in any case.

Error during export

Error messages are output if errors occur during export. If an export method "With deletion" has been selected, data are only deleted if at least one of the two export files (if both export paths are being used) has been correctly saved at the export path.

If data has only been successfully copied to one of the two export paths, the export file is also saved in the "Engineeringtools\ua_tmp" subfolder of the project directory.

If errors occur during copying to both export paths, the temporary export file in the project directory is deleted and error messages are generated. The archive to be exported is unchanged. It is reset to its original state before export.

7.2.3 Data backup

While export is specifically for the archive management user archives, the engineering tool can generate corresponding backup scripts for all APF modules. These scripts save the content of one or more archives in csv files without deleting the data records from the archive.

The data backup is described in the section Data backup of the user archives (Page 247).

7.3 User archives

7.3.1 FB_ ARCHGROUP

The archive groups are defined in this user archive. The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of the group	INTEGER 32
NAME	Name of the group	STRING 32
DESCR	Description of the storage location group	STRING 128
TXTLIB	ID for multilingual management in the text library	INTEGER 32
TYPE	Type of archive group 1 = Batch-based with fixed structure in some cases 2 = Freely definable	INTEGER 32
REC_ID_MAX	Currently highest "REC_ID" in archive	INTEGER 32
EXPORT_METHOD	Export method	INTEGER 32
MAX_NO_OF_REC	Maximum number of data records possible	INTEGER 32
FULL_NO_OF_REC	Number of data records as of which deletion is triggered.	INTEGER 32
DEL_NO_OF_REC	Number of data records to be deleted when archive is full (FULL_NO_OF_REC).	INTEGER 32
EXP_PATH	First path for saving export file	STRING 255
EXP_PATH2	Second path for saving export file	STRING 255

7.3.2 FB_ARCHVALUE_XXX

The user archives "FB_ARCHVALUE_001" to "FB_ARCHVALUE_XXX" are allocated to archive groups 1 to XXX of the "FB_ARCHGROUP" archive.

Up to 100 archives are defined in each of the "FB_ARCHVALUE_XXX" user archives. The user archives are structured as follows:

Element	Description	Format
POS	Position in the visualization order	INTEGER 32
NAME	Parameter name	STRING 32
UOM	Unit of parameter	STRING 16
FORMAT	Format of parameter	INTEGER 32
ENUM_MAT	Name of coding/enumeration	STRING 16
HL	High limit of operator control for numerical values	DOUBLE

Element	Description	Format
LL	Low limit of operator control for numerical values	DOUBLE
DESCR	Description of the parameter	STRING 128
TXTLIB	ID for multilingual management in the text library	INTEGER 32
FCT_BLOCK_IO_NAME	Name of the input in the corresponding function block	STRING 20
VISIBLE	Display in process picture	INTEGER 32

Format coding

The following table shows the format coding in "INTEGER":

Integer value	Format
0	INTEGER 32
-	-
1	REAL without decimal places
2	REAL with 1 decimal place
3	REAL with 2 decimal place
4	REAL with 3 decimal place
5	REAL with 4 decimal place
-	-
10	STRING 8 characters
11	STRING 16 characters
12	STRING 32 characters
13	STRING 64 characters
14	STRING 128 characters
15	STRING 14 characters (for PLM Dairy Library)
-	-
20xx	DATE
21xx	TIME
22xx	DATE AND TIME
-	-
30	BOOL
40	ENUMERATION -> STRING 16 characters in the archive. ENUM_MAT column contains name of S7 enumeration
50	MATERIAL -> STRING 64 characters in archive. ENUM_MAT column contains names of MaterialClassMaterialType number combination (e.g. 001003 = MatCl.001 and Mat.Type 003)
-	-
52	MATERIALLOT (release check) -> STRING 64 characters in archive
54	CONTENTS FORMAT (format-related value, cannot be modified)
-	-
60	32BITS (material class releases)

Integer value	Format
61	32BITS (material type releases)
62	32BITS (static procedure enables from the Dairy Library)
63	32BITS (dynamic procedure enables from the Dairy Library)
-	-
70	CONTENTS FORMAT (format-related value, can be modified)
-	-
71	STORAGE STATE -> STRING 32 characters in archive
72	MATERIAL STATE -> STRING 32 characters in archive
73	JOB STATE -> STRING 32 characters in archive
74	AENUMERATION (archive enumeration) -> STRING 128 characters in archive. ENUM_MAT column then contains the name of the archive.
-	-
83	ANY MATERIAL, regardless of material class/type

DATE, TIME and DATE AND TIME format

Two additional values are available for the formats DATE, TIME and DATE AND TIME:

Integer value	Format
01	German date format (DD.MM.YYYY) + time (hh:mm:ss)
02	German date format (DD.MM.YYYY) + time (hh:mm)
-	-
11	U.S. date format (MM/DD/YYYY) + time (hh:mm:ss)
12	U.S. date format (MM/DD/YYYY) + time (hh:mm)
-	-
21	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm:ss)
22	ISO 8601 date format (YYYY/MM/DD) + time (hh:mm)
-	-
31	British date format (DD/MM/YYYY) + time (hh:mm:ss)
32	British date format (DD/MM/YYYY) + time (hh:mm)

32BITS format

For formats 60-63, the APF Engineering Tool always creates all four releases. If material class releases for storage locations are archived, all 128 material class releases are automatically created (4 cells, each 32-bit).

STORAGE/MATERIAL/JOB STATE format

The APF Engineering Tool creates the following user archives, where various job, storage and material states are saved:

- FB_JOB STATUS
- FB_STORAGE_STATUS
- FB_MATERIAL_STATUS

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID	INTEGER 32
NAME	Name	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

AENUMERATION format

An integer is saved in the archive, and a text up to 128 characters long for this integer can be read from a corresponding archive.

In the APF Engineering Tool, you can specify a name in the "ENUM_MAT" column for an archive which does not yet exist. This archive is then automatically generated and filled by the APF Engineering Tool. This archive is structured as follows:

Element	Description	Format
REC_ID	Internal ID	IINTEGER 32
NAME	Name	STRING 128
TXTLIB	ID for multilingual management in the text library	INTEGER 32

Text formats

In contrast to other APF modules, the text instead of an ID is saved in the archive for all formats which are to represent a text. A material, for example, is saved in the archive as text rather than an ID as in parameter management.

The same applies for the following formats:

- 40 ENUMERATION -> STRING 16
- 50 MATERIAL -> STRING 64
- 52 MATERIALLOT -> STRING 64
- 71 STORAGE STATE -> STRING 32
- 72 MATERIAL STATE -> STRING 32
- 73 JOB STATE -> STRING 32
- 74 AENUMERATION -> STRING 128

7.3.3 FB_ARCHIVING_XXX

Up to 2000 archive data records and their values are saved in each of the "FB_ARCHIVING_XXX" user archives. There are two types of archives when using the engineering tool.

Depending on the type of archive group defined with the APF Engineering Tool and saved in the "FB_ARCHIVGROUP" user archive, the "FB_ARCHIVING_XXX" can be used as follows:

- Batch-based archiving
- Archiving freely-definable data

Batch-based archiving

The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of archive data record	INTEGER 32
REC_READ	Job data read (MES)	BOOL
REC_EXPORTED	Job data exported (C function > csv file)	BOOL
REC_FINISHED	Job completed	BOOL
REC_PREPARED	Read enable (MES)	INTEGER 32
BA_NA	Batch name	STRING 32
BA_ID	Batch ID	INTEGER 32
TIME_START	Start time of the job	DATE
TIME_END	End time of the job	DATE
SOURCE	Source	STRING 64
SOURCE_ID	ID of the source	INTEGER 32
DEST	Name of the destination	STRING 64
DEST_ID	ID of the destination	INTEGER 32
MAT_NAME	Material	STRING 64
MAT_ID	ID of the material	INTEGER 32
MLOT_NAME_SRC	Material lot source	STRING 64
MLOT_ID_SRC	ID of the material lot source	INTEGER 32
MLOT_NAME_DST	Material lot destination	STRING 64
MLOT_ID_DST	ID of the material lot destination	INTEGER 32
MLOT_NAME_DST_OLD	Previous material lot	STRING 64
MLOT_ID_DST_OLD	ID of previous material lot	INTEGER 32
SP	Overall setpoint	DOUBLE
PV	Actual value (MLOT-based)	DOUBLE
UOM	Unit of the setpoint/actual value	STRING 16
PRC_NA	Procedure name	STRING 32
PARAMSET	Parameter set	STRING 64
PARAMSET_ID	ID of the parameter set	INTEGER 32

Element	Description	Format
CHANGEDATE	Date of the last change to parameter set	DATE
STATUS	Status designation	STRING 64
STATUS_ID	ID of the status	INTEGER 32
VALUE_<ioname> …. VALUE_<ioname>	First value to be archived …. Last value to be archived	Depends on "FB_ARCHVAL-UE_XXX" archive

Archiving freely-definable data records

The archive has the following structure:

Element	Description	Format
REC_ID	Internal ID of archive data record	INTEGER 32
REC_READ	Job data read (MES)	BOOL
REC_EXPORTED	Job data exported (C function > csv file)	BOOL
REC_FINISHED	Job completed	BOOL
REC_PREPARED	Read enable (MES) 0 = MES may not read 1 = MES may read 2 = Released for deleting	INTEGER 32
VALUE_<ioname> …. VALUE_<ioname>	First value to be archived …. Last value to be archived	Depends on "FB_ARCHVAL-UE_XXX" archive

The archive values are assigned the column name "VALUE_" with the "FCT_BLOCK_IO_NAME" of the FB_ARCHVALUE_XXX table.

Note

Forming the actual value column name (a maximum of 20 characters) from "VALUE_" and "<ioname>" limits the actual useful length of the IO name to 14 characters.

7.3.4 FB_JOB_STATUS, FB_STORAGE_STATUS, FB_MATERIAL_STATUS

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID	INTEGER 32
NAME	Name	STRING 32
TXTLIB	ID for multilingual management in the text library	INTEGER 32

7.3.5 AENUMERATION (name in ENUM_MAT)

The archives are structured as follows:

Element	Description	Format
REC_ID	Internal ID	INTEGER 32
NAME	Name	STRING 128
TXTLIB	ID for multilingual management in the text library	INTEGER 32

7.4 Operator control and monitoring

There are no picture objects assigned to the "FB_XXXMA" block. The block is operated on the OS using the configured APF modules.

Messages

The following messages can be output:

Message class	Event	Message text
Operator message	New data record successfully generated with FB_XXXAR / FB_XXXAI	@7%s@: Data record @3%d@ created
Operator message	New data record successfully modified with FB_XXXAR / FB_XXXAI	@7%s@: Data record @3%d@ modified
Operator message	New data record successfully deleted with FB_XXXAR / FB_XXXAI (no MES)	@7%s@: Data record @3%d@ deleted
Operator message	New data record successfully selected for deletion with FB_XXXAR / FB_XXXAI (MES)	@7%s@: Data record @3%d@ marked for deletion
OS process control message - fault	Attempt to generate existing data record again with FB_XXXAR / FB_XXXAI	@7%s@: Data record @3%s@ already exists - cannot be created again! (FB_UA_WM.ID: @4%d@ FB_000AR.ID: @5%d@)
OS process control message - fault	Attempt to delete non-existent data record with FB_XXXAR / FB_XXXAI	@7%s@: Data record @3%s@ does not exist - cannot be deleted! (FB_UA_WM.ID: @4%d@ FB_000AR.ID: @5%d@)
OS process control message - fault	Attempt to modify non-existent data record with FB_XXXAR / FB_XXXAI	@7%s@: Data record @3%s@ does not exist - cannot be modified! (FB_UA_WM.ID: @4%d@ FB_000AR.ID: @5%d@)
OS process control message - fault	Failed to back up an archive group	@7%s@: Error backing up the archive
OS process control message - fault	An error occurred during cyclic deletion of the selected archive data records	@7%s@: Error during cyclic deletion
OS process control message - fault	Failed to export an archive group	@7%s@: Error exporting archive data
OS process control message - fault	An archive group with export method 1 or 2 reached the set archive size and data records had to be deleted.	@7%s@: Caution! Data unexpectedly had to be deleted!
OS process control message	Cyclic export action: The export file could not be created.	@7%s@: Export file (@2%s@) could not be exported

Message class	Event	Message text
OS process control message	Cyclic export action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the export file
OS process control message	Cyclic export action: Directory 1 could not be created	@7%s@: Export directory 1 does not exist / could not be created
OS process control message	Cyclic export action: Directory 2 could not be created	@7%s@: Export directory 2 does not exist / could not be created
OS process control message	Cyclic export action: Copy failed	@7%s@: Export file 1 (@2%s@) could not be copied
OS process control message	Cyclic export action: Copy failed	@7%s@: Export file 2 (@2%s@) could not be copied
OS process control message	Cyclic export action: Directory in ini file is empty	@7%s@: Configured export directory 1 is corrupt
OS process control message	Cyclic export action: Directory in ini file is empty	@7%s@: Configured export directory 2 is corrupt
OS process control message	Cyclic export action: Call of external export program failed	@7%s@: Initialization of the export program failed
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ error code: @5%d@
OS process control error	OS scripts	FB ID: @3%d@ REC ID: @4%d@ invalid value for date @6%s@
OS process control error	Cyclic backup action: The export file could not be created.	@7%s@: Backup file (@2%s@) could not be exported
OS process control error	Cyclic backup action: The readout of the project path was incorrect.	@7%s@: Project path could not be read. Cannot copy the backup file
OS process control error	Cyclic backup action: Directory 1 could not be created	@7%s@: Backup directory 1 does not exist / could not be created
OS process control error	Cyclic backup action: Directory 2 could not be created	@7%s@: Backup directory 2 does not exist / could not be created
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 1 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Copy failed	@7%s@: Backup file 2 (@2%s@) could not be copied
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 1 is corrupt
OS process control error	Cyclic backup action: Directory in ini file is empty	@7%s@: Configured backup directory 2 is corrupt
OS process control error	Cyclic backup action: Call the external backup program failed	@7%s@: Initialization of the backup program failed

Message associated values

The table below shows the message associated values:

Message associated value	Format	Value
Message associated value 1	@1%s@	Batch name
Message associated value 2	@2%s@	Data record / export file / backup file
Message associated value 3	@3%d@	Batch ID

Message associated value	Format	Value
Message associated value 4	@4%d@	ID of the FB_UA_WM block
Message associated value 5	@5%d@	ID of the FB_XXXAR / FB_XXXAI block
Message associated value 6	@6%s@	Archive name / archive group + data record ID
Message associated value 7	@7%s@	Logged-on user
Message associated value 8	@8%s@	Not used
Message associated value 9	@9%s@	Not used
Message associated value 10	@10%s@	Not used

In order to discharge the message system load, C function export and deletion processes only produce error messages. Only archiving triggered by the block (paste, update, delete) is logged with a message even when successful.

An error message is also triggered if deletion was required with export methods 1 or 2.

For additional information, refer to the PCS 7 online help.

Best before / UTC time supply

8

8.1 FB_BBD

8.1.1 Description

Object properties

The following table shows the object properties of the "FB_BBD" block:

Name	Number	Family	Module
FB_BBD	FB1794	APF	Best before

Area of application

The block is used to generate and provide the date of birth (DOB) and best before date (BBD), and to supply the UTC time, local time (LOT) and modified Julian date (MJD).

Note

The AS time system must be set to UTC time. The block operates in a UTC time range of January 01, 1990 to December 31, 2039.

Startup characteristics

The "FB_BBD" block is called from the startup OB upon CPU startup. This call is configured during CFC configuration by the CFC editor. Internal data and outputs are reset upon startup.

Program execution starts once the "RUNUPCYC" cycles have elapsed.

During startup, the output "QRUN_UP" = 1.

Time characteristics

The block must be called from a cyclic interrupt OB. The CFC Editor enters the sampling time of the block in the "SAMPLE_T" parameter.

Communication

Communication on the AS level is performed via the block inputs/outputs.

There is no data communication with the OS.

8.1.2 Functions

Overview

The block executes the following functions:

- Generating and providing the date of birth (DOB)
- Generating and providing the best before date (BBD)
- Outputting UTC time, local time (LOT) and the modified Julian date (MJD)

The date of birth and best before data are provided at the block outputs either cyclically or on request, depending on the mode.

MODE (WORD format)

Bit	Symbol	Meaning
0	NOT_EU_DST	0 = European daylight saving time - time of change is calculated 1 = Switch to daylight saving time with the "DST_ON" input
1	UTC_EXT	0 = UTC time of AS read internally in block 1 = External (UTC_DT and UTC_DTB)
2		Not used
3		Not used
4	TRIGGER	0 = DOB and BBD provided cyclically 1 = DOB and BBD output triggered (with the "GENERATE" input)
5		Not used
6		Not used
7		Not used
8	SUPP_SEC	Suppression of seconds
9	SUPP_MIN	Suppression of minutes
10	SUPP_HOUR	Suppression of hours
11		Not used
12	RND_SEC	Seconds rounded up
13	RND_MIN	Minutes rounded up
14	RND_HOUR	Hours rounded up
15		Not used

Time source and UTC time

One of the two following options is used as the time source:

- The internal time (read in the block from the AS).
- A time specified externally at the UTC_DT and UTC_DTB inputs (January 01, 1990 to December 31, 2039).

Bit 1 of the mode word is used for selection. The UTC time is displayed at the QUTC_DT and QUTC_DTB outputs of the block

Local time

The local time is displayed at the QLOT_DT and QLOT_DTB outputs of the block. The local time is the UTC time adjusted for daylight saving time and time zone.

The divergence between local time and UTC time due to the time zone is specified at the "LOT_DEV" input (in minutes, +/- 720). The default value of the input is "60". This represents the time zone "UTC + 01:00" (Amsterdam, Berlin, Bern, etc.).

The divergence due to daylight saving time is specified at the "DST_DEV" input (in minutes, +/- 120). The default value of the input is "60".

Switch to daylight saving time

For Europe, the clock change time is calculated in the block. The time of the switch to daylight saving time is the last weekend (Sat/Sun) in March; the clock is set forward at 2:00 am on Sunday. The end of daylight saving time is the last weekend (Sat/Sun) in October; the clock is set back at 3:00 am on Sunday.

For regions outside Europe, the clock change time can be set at the "DST_ON" input. Bit 0 of the mode word is used for selection.

Modified Julian date

The block calculates the modified Julian date on the basis of the UTC time and displays it at the "QMJD" output.

The resolution of the value is around 15 minutes. Higher resolution is available at the "QMJD_DAY" and "QMJD_TOD" outputs. "QMJD_DAY" contains the integer places and the "QMJD_TOD" output the decimal places of the modified Julian date

Date of birth (DOB) and best before date (BBD)

The date of birth and best before data are provided at the block outputs either cyclically or on request, depending on the mode. Bit 4 (TRIGGER) of the mode word is used to set the mode.

With triggered provision of the DOB and BBD, a positive edge at the "GENERATE" input generates the date of birth and best before date and outputs these dates at the QDOB_DT/ QDOB_DTB and QBBD_DT/QBBD_DTB outputs for further processing. The provision of the data is indicated at the "PROVIDED" output ("PROVIDED" = 1). The "PROVIDED" output is rest when the trigger is removed.

Provision is also indicated at the "PROVIDED" output with cyclic provision of the DOB and BBD.

The shelf life is specified at the SL_HOUR and SL_DAY inputs (SL = Shelf Life); the two values are added together.

The setting range for "SL_HOUR" is between 0 and 9999 hours and the range for "SL_DAY" between 0 and 6000 days.

Rounding DOB and BBD to full minutes, hours and days

The date of birth / best before data can be rounded up to full minutes, hours or days. The rounding function is executed before the function for suppressing seconds, minutes and hours.

- Rounding up seconds to full minutes:
  The seconds are set to "0" and the minute value is increased by 1. Bit 12 (RND_SEC) of the mode word is used for selection.
- Rounding up minutes to full hours:
  The second and minute values are set to "0" and the hour value is increased by 1. Bit 13 (RND_MIN) of the mode word is used for selection.
- Rounding up hours to full days:
  The second, minute and hour values are set to "0" and the day value is increased by 1. Bit 14 (RND_HOUR) of the mode word is used for selection.

Suppressing seconds, minutes and hours in the DOB and BBD

The seconds, minutes, and/or hours in the date of birth / best before date can be suppressed (set to = "0"). Bit 8 (SUPP_SEC), bit 9 (SUPP_MIN) and bit 10 (SUPP_HOUR) of the mode word are used for selection. Suppression is executed after any rounding processes.

8.1.3 I/Os

Input parameters

The following table shows the input parameters of the "FB_BBD" block:

Parameter	Description	Type	Default
SAMPLE_T	Sampling time [s]	REAL	1
RUNUPCYC	Number of startup cycles	INT	10
UTC_DT	External UTC time part A	DWORD	16#08123123
UTC_DTB	External UTC time part B	DWORD	16#59590000
LOT_DEV	Difference between local time and UTC [min]	INT	60
DST_DEV	Difference between daylight saving time and local time [min]	INT	60
SL_HOUR	Best before date in hours (0 – 9999)	DINT	0
SL_DAY	Best before date in days (0 – 6000)	DINT	0
GENERATE	1 = Generate DOB / BBD	BOOL	1
DST_ON	1 = Daylight saving time active	BOOL	30.0
MODE	Mode of the block	WORD	0

Output parameters

The following table shows the output parameters of the "FB_BBD" block:

Parameter	Description	Type	Default
QRUN_UP	Status display startup active = 1	BOOL	0
QNOT_OK	Block flag "OK" not set, error = 1	BOOL	0
QGR_ERR	Function block group error/fault	BOOL	0
QPARAMF	1 = Parameter assignment error,	BOOL	0
QERR	1 = Function error	BOOL	0
PROVIDED	1 = DOB / BBD generated and provided	BOOL	0
QDST_ON	Daylight saving time is active = 1	BOOL	0
QUTC_DT	Display of UTC time part A	DWORD	16#08123123
QUTC_DTB	Display of UTC time part B	DWORD	16#59590000
QLOT_DT	Display of local time part A	DWORD	16#08123123
QLOT_DTB	Display of local time part B	DWORD	16#59590000
QDOB_DT	Date of production part A	DWORD	16#08123123
QDOB_DTB	Date of production part B	DWORD	16#59590000
QBBD_DT	Best before date part A	DWORD	16#08123123
QBBD_DTB	Best before date part B	DWORD	16#59590000
QMJD	Modified Julian date	REAL	0.0
QMJD_DAY	Modified Julian date integer places (whole days)	REAL	0.0
QMJD_TOD	Modified Julian date decimal places (fractions of days)	REAL	0.0

8.1.4 Error handling

The errors detected in the block are signaled by block outputs:

Output	Meaning
QPARAMF	1 = parameter assignment error SL_HOUR < 0 or > 9999 SL_DAY < 0 or > 6000 LOT_DEV < -720 or > 720 DST_DEV < -120 or > 120
QERR	1 = Function error UTC time: Year < 1990 or > 2039 UTC time: Month < 1 or > 12 UTC time: Day < 1 or > 31 UTC time: Hours < 0 or > 23 UTC time: Minutes < 0 or > 59

If a "QPARAMF" or "QERR" occurs, DOB and BBD are not calculated and the "PROVIDED" output remains "0".

8.2 User archives

There is no access to user archives.

8.3 Operator control and monitoring

There are no picture objects assigned to the "FB_BBD" block.

Data structures and data formats

9

9.1 UDT_UA_R_ST

The "UDT_UA_R_ST" structure is used to provide the read data from the "FB_UA_RM" communication block.

The structure is as follows:

Parameter	Description	Type	Default
REQ_ID	Repetition of request data: ID of the requesting block	INT	0
ARCH_NA	Repetition of request data: Archive name	STRING 16	''
ARCH_NO	Repetition of request data: Archive number	DINT	L#0
REC_ID	Repetition of request data: Data record number in user archive	DINT	L#0
NO_REC	Repetition of request data: Number of rows to be read	DINT	L#0
COLUMN	Repetition of request data: Column number at which reading commences	DINT	L#0
NO_COL	Repetition of request data: Number of columns to be read	DINT	L#0
OFFSET	Offset, application-specific value	DINT	L#0
CALC_LEN	Data length calculated in the OS script, application-specific value	DINT	L#0
GEN_OS	Generation version reported back	DWORD	DW#16#0
DATA	Mailbox pointer for receive data	STRUCT	
• SYNC		BYTE	B#16#0
• TYPE		BYTE	B#16#0
• LENGTH		WORD	W#16#0
• DB_NO		WORD	W#16#0
• P		DWORD	DW#16#0
NEW_DATA	Read job feedback 1 = New data available 0 = No new data available	BOOL	FALSE
QERR	1 = Read error requested data and read data do not match	BOOL	FALSE
QARCHERR	1 = Archive error Invalid data in receive buffer A script error has occurred in the OS	BOOL	FALSE
QCOM_ERR	1 = Communication error Error message from "BRCV" coupling blocks	BOOL	FALSE

Parameter	Description	Type	Default
QPARAMF	1 = parameter assignment error duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	FALSE
Q1 to Q11	Feedback from status bit 1 - 11 on read job Application-specific value, written by OS script	BOOL	FALSE

9.2 UDT_UA_R_REQ_ST

The "UDT_UA_R_REQ_ST" structure is used to select data and trigger reading of the data.

The structure is as follows:

Parameter	Description	Type	Default
REQ_ID	ID of the requesting block (must be unique)	INT	0
ARCH_NA	Archive name	STRING 16	''
ARCH_NO	Archive number	DINT	L#0
REC_ID	Data record number in user archive	DINT	L#0
NO_REC	Number of rows to be read	DINT	L#0
COLUMN	Column number at which reading commences; 0 = Entire row	DINT	L#0
NO_COL	Number of columns to be read	DINT	L#0
GEN_AS	Generation version	DWORD	DW#16#0
PARA_1	Associated value 1 for read job, application-specific value	DINT	L#0
PARA_2	Associated value 2 for read job, application-specific value	DINT	L#0
PARA_3	Associated value 3 for read job, application-specific value	DINT	L#0
PARA_4	Associated value 4 for read job, application-specific value	REAL	0.0
PARA_5	Associated value 5 for read job, application-specific value	REAL	0.0
PARA_6	Associated value 6 for read job, application-specific value	REAL	0.0
DATA	Mailbox pointer for application-specific auxiliary data cell	STRUCT	
• SYNC		BYTE	B#16#0
• TYPE		BYTE	B#16#0
• LENGTH		WORD	W#16#0
• DB_NO		WORD	W#16#0
• P		DWORD	DW#16#0

Parameter	Description		Type	Default
WM_ID	ID of the FB_UA_WM, The ID is used in parameter management to recalculate the parameters.		INT	0
DATA_EN	Activation of the application-specific auxiliary data cell 0 = Not active 1 = Active, The data are entered in the communication interface.		BOOL	FALSE
REC_REQ	Triggers the read job 1 = Activate read job 0 = Deactivate read job		BOOL	FALSE
MON_ERR	1 = Job monitoring timeout (monitoring error); Not evaluated in the program.		BOOL	FALSE
REQ_TYP	Unique ID of a block type. Examples of block type IDs:		BYTE	B#16#0
	FB_UA_RM	1		
	FB_UA_WM	2		
	FB_XXXMA	3		
	FB_MLOT	4		
	FB_STLOC	5		
	FB_STMAT	6		
	FB_STOR	7		
	FB_ST_AC	8		
	FB_SACT	9		
	FB_XXXPA	10		
	FB_JOB	11		
	FB_000DP	12		
	FB_000PG	13		
	FB_000PH	14		
	FB_DJOB	15		
	FB_DS	16		
	FB_DP	17		
	FB_MP	18		
	FB_XXXAD	19		
	FB_BBD	20		
	FB_PA_ST	21		

9.3 UDT_UA_W_ST

This structure is used for feedback to the triggering block.

The structure is as follows:

Parameter	Description	Type	Default
REQ_ID	Repetition of request data: ID of the requesting block	INT	0
MODE	Repetition of request data: Operating mode	WORD	W#16#0
ARCH_NA	Repetition of request data: Archive name	STRING 16	''
ARCH_NO	Repetition of request data: Archive number	DINT	L#0
REC_ID	Repetition of request data: Data record number in user archive	DINT	L#0
NO_REC	Repetition of request data: Number of rows to be written	DINT	L#0
COLUMN	Repetition of request data: Column number at which writing commences	DINT	L#0
NO_COL	Repetition of request data: Number of columns to be written	DINT	L#0
WM_ID	ID of the FB_UA_WM,	INT	0
GEN_OS	Generation version reported back	DWORD	0
QERR	1 = Error Job data in WinCC communication interface do not correspond to original job data	BOOL	FALSE
QARCHERR	1 = Archive error Job data in the WinCC communication interface reset by the OS	BOOL	FALSE
QCOM_ERR	1 = Communication error Error message from "BSEND" coupling blocks	BOOL	FALSE
QPARAMF	1 = parameter assignment error duplicate REQ_ID or incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	FALSE
QWR_OK	Write job feedback 1 = Data written to archive 0 = Write job still running or no write request yet	BOOL	FALSE
Q1 to Q11	Feedback from status bit 1 - 11 on write job Application-specific value, written by OS script	BOOL	FALSE

9.4 UDT_UA_W_REQ_ST

This structure is used to select data and trigger writing of the data.

The structure is as follows:

Parameter	Description	Type	Default
REQ_ID	ID of the requesting block The ID must be unique.	INT	0
MODE	Mode, evaluated by OS script: e.g.: 1 = Create new data record 2 = Overwrite existing data record	WORD	W#16#0
ARCH_NA	Archive name	STRING 16	''
ARCH_NO	Archive number	DINT	L#0
REC_ID	Data record number in user archive	DINT	L#0
NO_REC	Number of rows to be written	DINT	L#0
COLUMN	Column number at which writing commences; 0 = Entire row	DINT	L#0
NO_COL	Number of columns to be written	DINT	L#0
GEN_AS	Generation version	DWORD	DW#16#0
PARA_1	Associated value 1 for write job, application-specific value	DINT	L#0
PARA_2	Associated value 2 for write job, application-specific value	DINT	L#0
PARA_3	Associated value 3 for write job, application-specific value	DINT	L#0
PARA_4	Associated value 4 for write job, application-specific value	REAL	0.0
PARA_5	Associated value 5 for write job, application-specific value	REAL	0.0
PARA_6	Associated value 6 for write job, application-specific value	REAL	0.0
DATA	Mailbox pointer for send data	STRUCT	
• SYNC		BYTE	B#16#0
• TYPE		BYTE	B#16#0
• LENGTH		WORD	W#16#0
• DB_NO		WORD	W#16#0
• P		DWORD	DW#16#0
REC_REQ	Trigger for write job 1 = Activate write job 0 = Deactivate write job	BOOL	FALSE
MON_ERR	1 = Job monitoring timeout (monitoring error); Not evaluated in the program.	BOOL	FALSE

Parameter	Description		Type	Default
REQ_TYP	Unique ID of a block type. Examples of block type IDs:		BYTE	B#16#0
	FB_UA_RM	1		
	FB_UA_WM	2		
	FB_XXXMA	3		
	FB_MLOT	4		
	FB_STLOC	5		
	FB_STMAT	6		
	FB_STOR	7		
	FB_ST_AC	8		
	FB_SACT	9		
	FB_XXXPA	10		
	FB_JOB	11		
	FB_000DP	12		
	FB_000PG	13		
	FB_000PH	14		
	FB_DJOB	15		
	FB_DS	16		
	FB_DP	17		
	FB_MP	18		
	FB_XXXAD	19		
	FB_BBD	20		
	FB_PA_ST	21		

9.5 UDT_ALARM_ST

This structure is used to provide message data for external message processing.

The structure is as follows:

Parameter	Description	Type	Default
SIG_1	Message signal 1	BOOL	FALSE
SIG_2	Message signal 2	BOOL	FALSE
SIG_3	Message signal 3	BOOL	FALSE
SIG_4	Message signal 4	BOOL	FALSE
SIG_5	Message signal 5	BOOL	FALSE
SIG_6	Message signal 6	BOOL	FALSE
SIG_7	Message signal 7	BOOL	FALSE
SIG_8	Message signal 8	BOOL	FALSE
SD_1	Associated value 1	UDT_ANY_ST	-
SD_2	Associated value 2	UDT_ANY_ST	-
SD_3	Associated value 3	UDT_ANY_ST	-
SD_4	Associated value 4	UDT_ANY_ST	-

Parameter	Description	Type	Default
SD_5	Associated value 5	UDT_ANY_ST	-
SD_6	Associated value 6	UDT_ANY_ST	-
SD_7	Associated value 7	UDT_ANY_ST	-
SD_8	Associated value 8	UDT_ANY_ST	-
SD_9	Associated value 9	UDT_ANY_ST	-
SD_10	Associated value 10	UDT_ANY_ST	-

9.6 UDT_PA_REQ_ST

This structure is used to transfer data and trigger reading of the parameter set data at the "FB_XXXPA" block (parameter management).

The structure is as follows:

Parameter	Description	Type	Default
REC_ID	Data record number in user archive (parameter set ID)	DINT	L#0
ACTIVEID	ID of the active job	INT	0
REQ	Request to read parameter set from archive and provide data at FB_XXXPA	BOOL	FALSE
RESET	Acknowledge error information	BOOL	FALSE
QRELEASED	1 = Function block released for processing	BOOL	FALSE
UNIT	Name of plant unit	STRING[24]	''
ARCH_NA	Archive name	STRING[16]	''
ARCH_NO	Archive number	DINT	0.0
BA_NA	Batch name	STRING[32]	''
BA_ID	Batch ID	DWORD	DW#16#0
SETPOINT	Setpoint	REAL	0.0

9.7 UDT_PA_CON_ST

This structure is used for feedback in response to the parameter set data request from the "FB_XXXPA" block (parameter management).

The structure is as follows:

Parameter	Description	Type	Default
NEW_DATA	Read job feedback 1 = New data available 0 = No new data available	BOOL	FALSE
QERR	1 = Read error - Requested data and read data do not match	BOOL	FALSE
QMON_ERR	1 = Job monitoring timeout (monitoring error);	BOOL	FALSE

Parameter	Description	Type	Default
QPARAMF	1 = parameter assignment error - Duplicate REQ_ID - Incorrect ID_1, R_ID_1 or ID_2, R_ID_2	BOOL	FALSE
QLEN_ERR	1 = Read error - The data length of the requested data does not match that of the read data	BOOL	FALSE
QARCHERR	1 = Archive error - Invalid data in receive buffer	BOOL	FALSE
QRELEASED	1 = Function block released for processing	BOOL	FALSE
QERR_VER	1 = Version error	BOOL	FALSE
ACTIVEID	ID of the active job	DWORD	0

Additional information 10

10.1 Configuration file

General system settings and module-specific system settings are saved in the "FB_Configuration.ini" file for all APF modules that access user archives. This file is located in the "\library\APF" subfolder of the project directory and must be available on and identical across all project computers.

Note

The "FB_Configuration.ini" file is configured automatically with the APF Engineering Tool and should not be edited manually.

The latest file status is copied to all project computers using the APF Engineering Tool and the APF Integrator Tool. You do not need to copy the file manually.

10.2 Data backup of the user archives

10.2.1 General information

Basic data backup process

In the engineering tool (see the "Engineering manual" for a description), you can define triggers for the user archives for each APF module to trigger data backup in csv files. The cycle time is set in the engineering tool.

To ensure maximum security, two independent backup paths can be defined. These are written simultaneously.

In a redundant server pair, data backup is only executed from the master currently active.

Differentiating data for backup

APF differentiates between the following:

- Configuration/engineering archives (e.g. material classes and parameter set groups). There is only one trigger per APF module.
- Runtime archives (e.g. material and parameter sets). Separate triggers are available for each archive.

Examples of triggers

The engineering tool generates the following three triggers for material master data when two material classes are defined:

- FB_MMBackupTriggerEN.pas
  Backs up the configuration/engineering archives FB_MATCLASS, FB_MATTYPE_001, FB_MATTYPE_002, FB_MATATTRIB_001, FB_MATATTRIB_002.
- FB_MMBackupTriggerRT1.pas
  Backs up the runtime archive FB_MATERIAL_001.
- FB_MMBackupTriggerRT2.pas
  Backs up the runtime archive FB_MATERIAL_002.

10.2.2 Storage at the backup path

Backups in subfolders are generated by module and daily for each module.

Name of module subfolders

- Material management: FB_MAT
- Material lot management: FB_MATLOT
- Storage location management: FB_STOR
- Parameter set management: FB_PARAM
- Archive management: FB_ARCHIVING

Name of day-based subfolders

<Year><Month><Day>, e.g. 20130115 for January 15, 2013

Note

To avoid problems with the day-based subfolders, you should not run backups for the individual archives at midnight (or midnight + / - 10 minutes), as a backup which has actually been triggered for multiple archives together may otherwise be copied to various different subfolders.

Naming convention for the backup files

sav_<Year><Month><Day>_<Time>_<Archive name>,

e.g., sav_130115_183059_Archiving_004

10.2.3 Data backup procedure

Data backup has been implemented in an external program to minimize disruption to normal OS server operation. The program exports the content of the archive to a csv file and then copies this file to the two backup paths.

The exported csv file is always stored in the project directory in the "Engineeringtools\ua_tmp" folder. From this location, the file can then be copied to the two backup directories. If you are only using one backup directory, this directory must be specified in the engineering tool under path 1; path 2 must be empty.

The temporary export file is deleted once the files have been copied to the backup directory. If errors occur in copying the files, the temporary export file is not be deleted, which means you can copy it manually.

Note

Because a large number of archives and associated backup actions can place a heavy load on the server, you should make sure that the various backup actions are not scheduled at the same time. Specify the backup times in minutes, for example, to avoid triggering multiple backup actions at the same time.

10.3 User archives

APF uses user archives for data management.

The engineering tool automatically generates the necessary archives:

- Some upon initial startup
- Others upon initial processing of an APF module
- The remaining required archives when blocks are generated

If possible, you should not operate the user archive manually during runtime, as this can result in blockages and inconsistent data.

The WinCC help has information on user archives regarding:

- Performance data (e.g. number of data cells)
- Keywords (which must not be used as cell names)