SIEMENS

SIMATIC

PCS 7 Process Control System PCS 7 Advanced Process Library Readme V9.1 SP2 Update 6 (Online)

Readme

Cybersecurity information 1

Industrial Security 2

Overview 3

Notes on installation 4

New features and changes as compared to previous versions 5

Information on configuration and operation 6

Notes on the documentation 7

Change history of PCS 7 Advanced Process Library 8

Version: 2025-10 (Online)

V9.1 SP2 Update 6
A5E51838815-AG

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 persons are those who, because of their training and experience, are familiar with the installation, assembly, commissioning, operation, decommissioning and disassembly of the product and can recognize risks and avoid possible hazards.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the application described in the catalog and the associated usage information. 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 Aktiengesellschaft. 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 Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E51838815-AG
Ⓟ 11/2025 Subject to change

Copyright © Siemens 2025.
All rights reserved

Table of contents

1 Cybersecurity information 5

2 Industrial Security 7

3 Overview 9

4 Notes on installation 11

4.1 Scope of delivery 11

4.2 Hardware requirements 11

4.3 Software requirements 11

4.4 Installing the PCS 7 Advanced Process Library 11

4.5 Uninstallation 12

5 New features and changes as compared to previous versions 13

5.1 Version 9.1 13
5.1.1 What's new in Version 9.1? 13
5.1.2 Changes in version 9.1 16
5.1.3 Update information for version 9.1 17

5.2 Version 9.1 SP1 20
5.2.1 What's new in Version 9.1 SP1? 20
5.2.2 Changes in Version 9.1 SP1 22
5.2.3 Update information for Version 9.1 SP1 24

5.3 Version 9.1 SP2 27
5.3.1 What's new in Version 9.1 SP2? 27
5.3.2 Changes in Version 9.1 SP2 29
5.3.3 Update Information for Version 9.1 SP2 31

5.4 Version 9.1 SP2 Update 1 34
5.4.1 Changes in Version 9.1 SP2 Update 1 34
5.4.2 Update Information for Version 9.1 SP2 Update 1 36

5.5 Version 9.1 SP2 Update 2 38
5.5.1 Changes in Version 9.1 SP2 Update 2 38
5.5.2 Update information for Version 9.1 SP2 Update 2 39

5.6 Version 9.1 SP2 Update 3 39
5.6.1 Changes in Version 9.1 SP2 Update 3 39
5.6.2 Update information for Version 9.1 SP2 Update 3 40

5.7 Version 9.1 SP2 Update 4 41
5.7.1 Changes in Version 9.1 SP2 Update 4 41
5.7.2 Update information for Version 9.1 SP2 Update 4 41

5.8 Version 9.1 SP2 Update 5 42
5.8.1 Changes in Version 9.1 SP2 Update 5 42
5.8.2 Update information for Version 9.1 SP2 Update 5 43

5.9 Version 9.1 SP2 Update 6 44
5.9.1 Changes in Version 9.1 SP2 Update 6 44
5.9.2 Update information for Version 9.1 SP2 Update 6 44
6 Information on configuration and operation 47
7 Notes on the documentation 51
8 Change history of PCS 7 Advanced Process Library 57

Cybersecurity information

1

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
https://www.siemens.com/cybersecurity-industry.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
https://new.siemens.com/cert.

Industrial Security

2

This product is a part of SIMATIC PCS 7 and integrated into the overall plant security concept. Siemens strongly recommends setting up, operating, maintaining, and decommissioning the plant environment according to the SIMATIC PCS 7 Compendium Part F – Industrial Security: Technical documentation SIMATIC PCS 7 (https://support.industry.siemens.com/cs/ww/en/view/109801081).

Overview

3

Note

Read these notes carefully; they contain important information and additional details about the PCS 7 Advanced Process Library .

The statements in this readme file take precedence over all the PCS 7 manuals.

4 Notes on installation

4.1 Scope of delivery

The scope of delivery includes the following library:

- PCS 7 Advanced Process Library V9.1 SP2

4.2 Hardware requirements

The installation is subject to the conditions of at least SIMATIC PCS 7 V9.1 SP2.

4.3 Software requirements

The blocks of the library require at least SIMATIC PCS 7 V9.1 SP2.

4.4 Installing the PCS 7 Advanced Process Library

Exit all applications before you start setup.

Install the library via the PCS 7 system setup or start the Setup.exe program of the PCS 7 Advanced Process Library. You will receive all further information during the installation process.

The library is copied to the same drive on which the PCS 7 basic software is installed. Installation requires about 1.6 MB of free memory on the hard disk.

Following the installation, you can find the blocks of the PCS 7 Advanced Process Library and templates under "PCS 7 AP Library V9.1".

Practical examples can be found in the example project "APL_Example_*xx*", ("*xx*" stands for one of the installed language variants EU = European, CHS = Chinese, FE = Japanese.

To update your AS project, transfer the blocks to your project and then perform a block type import.

Note

If CPU 410-5H is not being used, the **"Type Change in Run" (TCiR)** function is not available for upgrading from ≤V9.1 to ≥V9.1 SP2 . In order to receive the delta compilation and delta load capability in your project, use the blocks from the previous version of the PCS 7 Advanced Process Library.
Note that in this case you can not use the new features and improvements described in the section "New features and changes compared to earlier versions".

4.5 Uninstallation

If you uninstall an older PCS 7 Advanced Process Library using Windows mechanisms, you must afterwards install the current/latest version of the PCS 7 Advanced Process Library again. It is not possible to uninstall via the product setup.

New features and changes as compared to previous versions

5

5.1 Version 9.1

5.1.1 What's new in Version 9.1?

- The blocks of the **Condition Monitoring Library (CML)** have been integrated into the APL. The F-blocks for the fail-safe partial stroke test application (`F_PST` and `F_PST_S`) are **not** included. They are freely available for download at the address <link will be specified at a later date>.
  The following blocks have been added: PumpMon, VlvMon, PrDrpMon, SteadyState, PST, PST_Calc, PST_OrQC, PST_Time.
- Two functions have been added to the **"Convert"** block family, namely `StruCmIn` (splits a structure comprising multiple values into discrete outputs) and `StruCmOut` (combines discrete inputs into a single structure).
- **Block-internal error states**, which are output to `ErrorNum` as error numbers, are displayed in the **standard view**.
  The status bits of the following blocks have been adapted for this purpose: ShrdResL, ShrdResS, MotL, MotS, MotSpdL, MotRevL, MotSpdCL, VlvL, VlvS, Vlv2WayL, VlvAnl, VlvMotL, VlvPosL, Intlk02, Intlk04, Intlk08, Intlk16, MonAnL, MonAns, MonDiL, MonDi08, OpAnl, OpAns, OpDi01, OpDi03, PIDConL, PIDConR, PIDConS, PIDStepL, FmCont, FmTemp, ModPreCon, ConPerMon, Ratio, MPC10x10, KalFilt, CountScL, CntOhSc, CountOh, TotalL, DoseL, SelA16In, SelAn16L, TimeTrig, SteadyState, PST.
- At the **channel block Pcs7AnOu**, `Feature.Bit5 =1` can be used to invert the direction of action of the `PV_In.Value` or `SimPV_In.Value` input parameter. This eliminates the need for a separate block for inverting, e.g. when using a fail-open valve.

- The **channel blocks** have been given a **"Channel active" identifier**. Identification is provided via the `DataXchg1` input of the upstream diagnostic block of the PCS 7 Basis Library and evaluated in the channel block.
  The following channel blocks evaluate this "Channel active" identifier: FbAnIn, FbAnOu, FbAnTot, FbDiIn, FbDiOu, FbDrive, FbEnMe, FbSwtMMS, FmCont, FmTemp, Pcs7AITC, Pcs7AnIn, Pcs7AnOu, Pcs7Cnt1, Pcs7Cnt2, Pcs7Cnt3, Pcs7DiCnt, Pcs7DiIn, Pcs7DiIT, Pcs7DiOu, Pcs7HaAI, Pcs7HaAO.
  The "Channel active" identifier is reset for channels that have been deactivated for HW Config. The channel blocks react to this with signal status 16#68 (uncertain, maintenance demanded) and `Bad = false`.

  Note

  Behavior change for channels deactivated in HW Config

  There has been a change in behavior in the channel blocks for channels deactivated in HW Config. Up to now, these reacted with signal status 16#00 (bad) and Bad=true. This behavior has now changed to signal status 16#68 (uncertain, maintenance demanded) and Bad=false.

  Behavior change when the "Create module driver" CFC function is not used or when 3rd-party diagnostics blocks are used

  Normally the DataXchg1 input of the channel blocks is automatically switched when a module driver is generated. DataXchg1 now also contains the new "Channel active" identifier. Therefore, make sure that the 3rd party diagnostics blocks provide this identifier, or, if necessary, configure the identifier for "Channel active" yourself: 2#xxxxxxxxxxxx1xxx (Byte0 Bit 3 = 1).

- The information on **interlocks, motor protection function, external error, and process control error** are now also available as additional outputs (`PermitOut, IntlockOut, ProtectOut,.., TripOut, FaultExtOut, CSF_Out`) for further processing in your user program.
  The following blocks have been extended for this purpose: FmCont, FmTemp, PIDConL, PIDConR, PIDConS, PIDStepL, DoseL, MotL, MotRevL, MotS, MotSpdCL, MotSpdL, Vlv2WayL, VlvAnL, VlvMotL, VlvPosL, VlvL, VlvS, MonAnL, MonAnS, MonDiL, MonDiS, OpDi01, OpDi03
- The information "**Internal simulation**" is now also available at the new `SimAct` output.
  The following blocks have been extended for this purpose: DoseL, FmCont, FmTemp, ModPreCon, PIDConL, PIDConR, PIDConS, PIDStepL, VlvPosL, VlvMotL, VlvAnL, VlvS, Vlv2WayL, VlvL, MonAnS, MonDiL, MonDiS, MonAnL, MPC10x10, TotalL, Ratio, MotL, MotRevL, MotSpdCL, MotSpdL, MotS, KalFilt.
- The **alarm limits** of the setpoint/actual value difference have been made available at the PIDConL, PIDStepL, PIDConR, MotSpdCL, VlvAnL, VlvPosL blocks as the **output** `ER_AH_LimOut, ER_AL_LimOut` . The alarm limits for overdosing and underdosing (relative to the dosing setpoint) have been made available for DoseL as the `ER_AH_TolOut, ER_AL_TolOut` output.
- The range limits and units are also available at the CountScL, MonAnL, MonAnS, MotSpdCL, OpAnL, OpAnS, Ratio, TotalL, PIDConL, PIDConS, PIDConR, PIDStepL, SteadyState, DoseL blocks as `xxxOpScaleOut` and `xxxUnitOut` outputs.

- At the **CntOhSc counter block**, the `CntHi1LimOut, CntHi2LimOut, HrsHi1LimOut, HrsHi2LimOut` outputs have been added for further interconnection of the limit values.
  The `OutAH_LimOut, OutWH_LimOut, OutTH_LimOut, OutTL_LimOut, OutWL_LimOut, OutAL_LimOut` outputs have been added to the **CountScL** block accordingly.
- The **analog auxiliary values** together with their units are also available as outputs (`UserAna1Out, UserAna2Out, UA1unitOut, UA2unitOut`) for further interconnection. This affects the following blocks: **MotL, MotRevL, MotSpdL, MotSpdCL, VlvL, VlvMotL, VlvPosL, VlvAnL, Vlv2wayL, DoseL, MonAnL, MonDiL, PIDConL, PIDConR, PIDStepL**.
- In order to increase the number of free bits for OS information, **all technological blocks** except the Small blocks now have a second `UserStatus2` input. This affects the following blocks: OpAnL, OpDi01, OpDi03, OpStations, OpTrig, MonAnL, MonDiL, MonDi08, ConperMon, FmCont, FmTemp, ModPreCon, PIDConL, PIDConR, PIDStepL, Ratio, MPC10X10, KalFilt, DoseL, MotL, MotRevL, MotSpdCL, MotSpdL, ShrdResL, Vlv2WayL, VlvMotL, VlvAnL, VlvPosL, Intlk02, Intlk04, Intlk08, Intlk16, Event, EventNck, EventTs, Event16Ts, SelA16In, SelAn16L, TimeTrig, VlvMon, PumpMon, PressPropMon.
- The behavior of the `xx_xx_Act` "**Limit active**" outputs with deactivated limit messages (`xx_xx_MsgEn =0`) can be configured via `Feature Bit` . If the limit message is deactivated and `Feature Bit = 1`, the output for "**Limit active**" is also reset.
  The new `Feature Bit` is available for the following blocks:
  `Feature Bit3:` **MonAnL, MonAnS, PIDConS, AV, CountOh, CountScL, TotalL, PumpMon**
  `Feature2 Bit12:` **PIDConL, PIDConR, PIDStepL, ConPerMon, DoseL, VlvPosL, MotSpdCL, FmCont, FmTemp, VlvAnL**.
- The **PIDConL, PIDConR, PIDStepL controller blocks** have been extended by the **"Display auxiliary values"** function. Two analog auxiliary values, namely `UserAna1, UserAna2` can now be displayed together with their `UA1unit, UA2unit` units in the **standard view**.
- The process input signals can now also be evaluated in inverted form at the **MonDiL, MonDiS, MonDi08 monitoring blocks**. This can be configured at the newly added `InvIn, InvIn1..8` inputs.
- With `Feature2 Bit13 = 1`, an error can also be reset at the MotL, MotRevL, MotSpdcL, MotSpdL, VlvMotL, VlvPosL, Vlv2WayL, VlvL **motor and valve blocks** in **local mode** (`LocalSetting = 1, 3`) via the **reset button** in the standard view.
- The analog **position feedback** Rbk at the **VlvPosL** block can now also be monitored in the **intermediate position** when **position control is switched off**.
  After a stop control, the manipulated variable is tracked for the time from `V_MonTiStatic` to the position feedback (`MV_Out = RbkOut`). The manipulated variable is then held and can be monitored.
- The Stop output has been added to the **VlvAnL** block. Stop is set when the analog valve has reached its position and no further control is pending.
- The **MotSpdCL** block has been extended to include a **directional interlock**.
- At the **DoseL** block, the automatically determined `DribbOut` **dribbling quantity** can be applied for a complete download using Feature2 Bit7 =1. The `OldDribbOut` input has been added for this purpose.

- The **TimeTrig** block has been extended to include an **external time input** and the **local time**. Local time is calculated using the `FC296 LOC_TIME_BAS` function from the PCS 7 Basis Library.
- At the **TimeTrig** block, the periodic trigger output has been extended to include a **minute output**.
- The **FlipFlop** block has been extended to include **JK functionality**.

5.1.2 Changes in version 9.1

- Suitable `SP_LiBy` and `SP_LiST` inputs were added in byte format to the **FbDiOu** block.
- The `SP_ChnST` output for reconnection to a technological block was added to the **FbDiOu** block.
- The signaling behavior for high-precision time stamping with toggled signal sequences has been improved at the **Pcs7DiIT** block.
- If the signaling is 0-active (`Feature.Bit29 =1`) and "Last saved value" is selected as startup behavior (`Feature.Bit0 =1`), the limit messages are no longer triggered after a complete download of the user program. This has been improved for the following blocks: **AV, CntOhSc, ConPerMon, CountOh, CountScL, DoseL, FmCont, FmTemp, MonAnL, MotSpdCL, PIDConL, PIDConR, PIDStepL, TotalL, VlvMon, VlvAnL, VlvPosL.**
- The function "**Bumpless switchover to automatic mode for valves, motors, and dosers**" at the **MotL**, **MotRevL**, **MotSpdCL**, **MotSpdL**, **VlvL**, **Vlv2WayL**, **VlvMotL**, **VlvPosL**, **VlvAnL**, **DoseL** blocks also takes into account the implausible values of the automatic commands with `Feature2.Bit3` and a bad status with `Feature2.Bit10`.
- Parameter default values have changed at the following blocks:
  MotSpdCL: `SP_Int 0.0`
  VlvPosL: `V_MonTiDynamic 30.0`
  Check these parameter values after an upgrade.
- The function of the `Stop` output has changed at the **VlvPosL** and **VlvMotL** blocks. The calculations of the `Status3.Bits24,25` have been adapted for this purpose.
  Previously, `Stop =1`: Motor is stopped and the valve is in an intermediate position.
  New behavior `Stop =1`: Motor is stopped regardless of the valve position.
- An outgoing motor trip error (`Trip =0->1`) at the **VlvPosL, VlvMotL** blocks must first be reset before it can move to the "Closed" or "Opened" idle position due to a monitoring error of the feedback signals.
- With **VlvAnL**, an interlock now has higher priority than tracking via `MV_TrkOn =1` with `MV_TrkRbk =0`. This block therefore now behaves like all controller blocks and the **VlvPosL**.
- With `SafePos =2` (stop), the manipulated variable `MV` of the `RbkOut` position feedback is tracked at the **VlvAnL** block. **VlvAnL** thus now behaves like VlvPosL.
- The manipulated variable limit outputs `MV_HiAct, MV_LoAct` at the **VlvAnL** block are now set when the manipulated variable limit is reached.

- With `Feature2.Bit10 = 1` (*bad quality of automatic commands taken into account ...*) at the **VlvAnL** block, a bad signal status at the `OpenAut.ST, CloseAut.ST, MV_Ext.ST` inputs only becomes effective after switching to **automatic mode** and not in **manual mode**. With bad signal status, the block sets the idle position configured at `SafePos`.
- At the **VlvPosL** block, the `OpenForce`, `CloseForce`, and `StopForce` inputs have a higher priority than `MV_ForOn`. An error number at the `ErrorNum` output is not output if `MV_ForOn` or one of the other force inputs is set.
- At the **VlvPosL** block, the activation enable for position control and direction-dependent interlocking has been improved.
- The manipulated variable limit outputs `MV_HiAct, MV_LoAct` at the **VlvPosL** block are now also set if the manipulated variable limits are reached when the position control is switched off.
- The **controller blocks PIDConL, PIDConR, PIDStepL, PIDConS, FmCont, FmTemp** can now continue to be operated even if a #NaN process value is present. The block acts as a sink, thus ensuring that no #NaN values are passed on to outputs.
- Behavior during setpoint tracking `SP_TrkPV = 1` has been improved at the **PIDConL, PIDConR, PIDStepL, PIDConS** blocks.
- At the **PIDStepL** block, messages for the position feedback `RbkWH_Act, RbkWL_Act` are no longer output for the "**Step controller without position feedback**" parameter assignment `WithRbk = 0`.
- The switchable inputs `OpenLi, StopLi, CloseLi` can be activated at the **PIDStepL** block via the `CtrlLiOp` parameter and `Feature.Bit7` in manual mode. This means that the `OpenLi, StopLi, CloseLi` inputs can now be used independently of `ModLiOp` and therefore independently of the operating mode switchover.
- At the **TimeTrig** block, it is now also possible to change the delay time of the single trigger while the single trigger is running. This means that the single trigger behaves in the same way as the periodic trigger.
- At the **FlowCorr** block, the `Flow.ST` status "Bad, device related" `16#68` is output if one of the relevant inputs for the formation of `Flow.ST` is not defined.

5.1.3 Update information for version 9.1

The following applies for the software update from version 9.0 SP 3 >=Update 4 to version 9.1:

1.	Make text interconnections	**Yes**
2.	Compile AS	**Complete compilation**
3.	Download AS program	**Complete download**
4.	AS STOP necessary	**Yes (AS STOP is not necessary when using a CPU 410-5H with TCiR)**
5.	Compile OS	**Complete compilation**

Note

Updating the blocks from the PCS 7 Advanced Process Library with TCiR

"Type import - Changed default value" dialog box

- If information on changed internal local default values in the static area is displayed in the dialog box during type import, these changes are not relevant. A complete download in STOP mode is not necessary.
- If information on changed default values is displayed in the dialog box at block I/Os during type import, these new default values must be applied in the "Update block types" dialog.

List of modified blocks

Name	Number	Block version	Interface change (AS STOP is not necessary when using a CPU 410-5H with TCiR)	New block contact for SFC type	Code change
AssetM	FB1840	9.5	Yes	No	No
AV	FB1903	5.5	Yes	No	Yes
CntOhSc	FB1803	5.5	Yes	No	Yes
ConPerMon	FB1805	5.5	Yes	No	Yes
CountScL	FB1806	5.5	Yes	No	Yes
CountOh	FB1864	5.5	Yes	No	Yes
DoseL	FB1809	5.5	Yes	No	Yes
Event	FB1811	5.5	Yes	No	Yes
Event16Ts	FB1887	5.5	Yes	No	Yes
EventNck	FB1904	5.5	Yes	No	Yes
EventTs	FB1812	5.5	Yes	No	Yes
FbAnIn	FB1813	9.5	Yes	No	Yes
FbAnOu	FB1814	9.5	Yes	No	Yes
FbAnTot	FB1817	9.5	Yes	No	Yes
FbDiIn	FB1815	9.5	Yes	No	Yes
FbDiOu	FB1816	9.5	Yes	No	Yes
FbDrive	FB1905	9.5	Yes	No	Yes
FbEnMe	FB1908	9.5	Yes	No	Yes
FbSwtMMS	FB1907	9.5	Yes	No	Yes
FlipFlop	FC389	5.5	Yes	No	Yes
FlowCorr	FB1916	9.5	Yes	No	Yes
FmCont	FB1818	9.5	Yes	No	Yes
FmTemp	FB1819	9.5	Yes	No	Yes
Integral	FB1823	5.5	Yes	No	No
Intlk02	FB1824	5.5	Yes	No	Yes
Intlk04	FB1825	5.5	Yes	No	Yes

Intlk08	FB1826	5.5	Yes	No	Yes
Intlk16	FB1827	5.5	Yes	No	Yes
KalFilt	FB1925	9.5	Yes	No	Yes
LPOptim	FB1844	5.5	Yes	No	No
ModPreCon	FB1843	5.5	Yes	No	Yes
MonAnL	FB1845	5.5	Yes	No	Yes
MonAnS	FB1912	5.5	Yes	No	Yes
MonDi08	FB1847	5.5	Yes	No	Yes
MonDiL	FB1848	5.5	Yes	No	Yes
MonDiS	FB1913	5.5	Yes	No	Yes
MotL	FB1850	5.5	Yes	Yes	Yes
MotRevL	FB1851	5.5	Yes	Yes	Yes
MotS	FB1910	5.5	Yes	No	Yes
MotSpdCL	FB1854	5.5	Yes	Yes	Yes
MotSpdL	FB1856	5.5	Yes	Yes	Yes
MPC10x10	FB1920	5.5	Yes	No	Yes
MPC_OpFB	FB1922	5.5	Yes	No	No
OpAnL	FB1865	5.5	Yes	No	Yes
OpAnS	FB1915	5.5	Yes	No	Yes
OpDi01	FB1866	5.5	Yes	No	Yes
OpDi03	FB1867	5.5	Yes	No	Yes
OpStations	FB1901	5.5	Yes	No	No
OpTrig	FB1868	5.5	Yes	No	Yes
Pcs7AITC	FB1933	9.5	Yes	No	Yes
Pcs7AnIn	FB1869	9.5	Yes	No	Yes
Pcs7AnOu	FB1870	9.5	Yes	No	Yes
Pcs7Cnt1	FB1833	9.5	Yes	No	Yes
Pcs7Cnt2	FB1834	9.5	Yes	No	Yes
Pcs7Cnt3	FB1835	9.5	Yes	No	Yes
Pcs7DiCn	FB1935	9.5	Yes	No	Yes
Pcs7DiIn	FB1871	9.5	Yes	No	Yes
Pcs7DiIT	FB1872	9.5	Yes	No	Yes
Pcs7DiOu	FB1873	9.5	Yes	No	Yes
Pcs7HaAI	FB1931	9.5	Yes	No	Yes
Pcs7HaAO	FB1932	9.5	Yes	No	Yes
PIDConL	FB1874	5.5	Yes	No	Yes
PIDConR	FB1875	5.5	Yes	No	Yes
PIDConS	FB1830	5.5	Yes	No	Yes
PIDKernR	FB1877	5.5	Yes	No	Yes
PIDStepL	FB1878	5.5	Yes	Yes	Yes
Polygon	FB1881	5.5	Yes	No	Yes
RateLim	FB1882	5.5	Yes	No	Yes
Ratio	FB1883	5.5	Yes	No	No
SelA16In	FB1888	5.5	Yes	No	Yes

SelAn16L	FB1889	5.5	Yes	No	Yes
Snd_AnaVal	FB1893	9.5	Yes	No	No
Snd_DigVal	FB1891	9.5	Yes	No	No
ShrdResL	FB1917	5.5	Yes	No	Yes
ShrdResS	FB1914	5.5	Yes	No	Yes
TimerP	FB1810	5.5	Yes	No	Yes
TimeTrig	FB1802	5.5	Yes	No	Yes
TotalL	FB1906	5.5	Yes	No	Yes
Vlv2WayL	FB1897	5.5	Yes	No	Yes
VlvAnL	FB1896	5.5	Yes	No	Yes
VlvL	FB1899	5.5	Yes	Yes	Yes
VlvMotL	FB1900	5.5	Yes	Yes	Yes
VlvPosL	FB1918	5.5	Yes	Yes	Yes
VlvS	FB1911	5.5	Yes	No	Yes

Note

- Blocks for which only attributes (except S7_contact), message texts, and/or version number were changed, are not listed.
- If the "**New block contact for SFC type**" column has a "**Yes**", this means that either the block type has a new parameter with "S7_contact=true" or the attribute S7_contact auf "true" has been set at an existing parameter. In both cases this results in an extension of the block I/Os and thus an interface change at the SFC type.
- If "Yes" is listed in the "Interface change" column and "No" in the "Code change" column, this indicates that either the interfaces have been prepared for a future function or the order of the static variables (VAR area) has been optimized.

5.2 Version 9.1 SP1

5.2.1 What's new in Version 9.1 SP1?

This section describes the new features compared to the previous version.

- **Resynchronization:**
  For certain scenarios, the following blocks can be synchronized to available and plausible feedback signals via feature bits. Positions and controls of the blocks are tracked in the process.
  - After CPU startup
  Feature Bit17: **OpAnL, OpDi01, OpDi03**
  Feature2 Bit17: **MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL**
  - After exiting "Out of service" mode
  Feature Bit18: **OpAnL, OpDi01, OpDi03**
  Feature2 Bit18: **MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL**
  - After exiting simulation
  Feature2 Bit18: **MotL**, **MotRevL**, **MotSpdCL**, **MotSpdL**, **VlvAnL**, **VlvL**, **VlvMotL**, **VlvPosL**
  - After system error CSF has cleared
  Feature2 Bit19: **MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL**
  - After 0->1 trigger of the input parameter StateTrkOn
  Feature2 Bit20: **MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL**
- **Switch-off enable:**
  The following blocks now also support a switch-off enable (PermitSafePos) in addition to the switch-on enable (Permit). If the switch-off enable is not present, switching off in the direction of the safety position is prevented.
  MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL.
- The **MotL, MotRevL, MotSpdCL** , and **MotSpdL motor blocks** have the following new functions:
  - The blocks can now also evaluate a **stop confirmation** (`FbkStop`) in addition to the run feedback. In addition, operation **without feedback signals** is now also possible, as is already the case with the valve blocks.
  - In the event of a **monitoring error**, it is possible to **retain the control**. As with the valve blocks, the parameters are set via the `MonSafePos` input.
- Additional **control for "Stop" or "Close"**
  The **MotL, MotRevL, MotSpdCL** and **MotSpdL** motor blocks can now also output a **stop command**. The **VlvL** valve block now supports a **close command** in addition to the open command.
- Valves with a large end position range can now be implemented at the **VlvAnL** block using Feature2 Bit15, whereby the `MV-Rbk` difference is also monitored:
  `Feature2 Bit15 =0` (default): No monitoring of the `MV-Rbk` difference in the end position ranges.
  `Feature2 Bit15 =1`: With monitoring of the `MV-Rbk` difference in the end position ranges.

- **Delayed setting of the** `Bad`output
  All **channel blocks** except **FmCont**, **FmTemp**, **Pcs7HaAi**, and **Pcs7HaAO** now have the option of setting the "`Bad`" output with a delay.
- Certain block types have been expanded to include the `BlkTypeInf` and `BlkTypeData` inputs. `BlkTypeInf` contains the **Block type number (**`TypNr`**)** and **block type version (**`VersId`**).**

WARNING
These additional inputs can neither be configured nor switched, and must not be changed under any circumstances. They are for internal use only and any attempt to manipulate them may lead to malfunctions.

The following new features are only available as of PCS 7 V9.1 SP 1.

- New **Pcs7ReIn** and **Pcs7Cnt4** channel blocks for signal acquisition of the following technology modules:
  - 6DL1138-6EA00-0EH1 Fast MultiIO and Counter
  - 6DL1138-6EC00-0EH1 Vibration Protection

5.2.2 Changes in Version 9.1 SP1

- The `S7_contact` parameter attribute has been set for the `ManModLi` input. The `ManModLi` I/O is thus available in the list of **block contacts** of an **SFC type**.
  Affected blocks: **DoseL, FmCont, FmTemp, ModPreCon, MotL, MotRevL, MotS, MotSpdCL, MotSpdL, MPC10x10, PIDConL, PIDConR, PIDStepL, Vlv2WayL, VlvAnL, VlvL, VlvMotL, VlvPosL, VlvS**.

  Note

  Please note that this extends the block contacts on existing SFC types and may result in an interface change at the SFC type. If you want to avoid this interface change, you can carry out the appropriate attribute synchronization. To do this, select "Destination (old)" for the ManModLi I/O during the attribute synchronization. The S7_contact=true attribute is then not adopted and the SFC type block interface is not extended by the ManModLi block contact.

- The limits are also visible in the **Limit view** in the "**Out of service**" mode.
  Affected blocks: **CntOhSc, ConPerMon, CountOh, CountScL, DoseL, FmCont, FmTemp, MonAnL, MonAnS, MotL, MotRevL, MotSpdCL, MotSpdL, PIDConL, PIDConR, PIDConS, PIDStepL, TotalL, VlvAnL, VlvMotL, VlvPosL**.

- Resettable monitoring errors, faults, or interlocks can now also be reset via the RstLi input in **local mode** `LocalSetting =1, =3` when `Feature2 Bit13 =1` is set. This affects the blocks **MotL, MotRevL, MotSpdCL, MotSpdL, Vlv2WayL, VlvL, VlvMotL, VlvPosL**.
- For the **MotL, MotRevL, MotSpdCL, MotSpdL, VlvAnL, VlvL, VlvMotL, VlvPosL** blocks, the **local mode with tracking of the control outputs (LocalSetting =2, =4)** has been adapted for non-existent feedback signals or implausible (discrepant) feedback signals NoFbkXXX =1 (e.g.: FbkOpen =1 AND FbkClose =1)**:**
  - The **MotL, MotRevL, MotSpdCL, MotSpdL, VlvL, VlvMotL** blocks output static monitoring errors in the event of discrepant feedback signals. After the monitoring counter has elapsed, the control for `MonSafePos =1` goes into the idle state. The control does not change with **VlvMotL**.
  - The **VlvAnL, VlvPosL** blocks output static monitoring errors if there are discrepant feedback signals (e.g.: FbkOpen =1 AND FbkClose =1). After the monitoring counter has elapsed, the analog control also tracks the Rbk feedback with `MonSafePos =1`.
  - When the feedback signal changes, the **VlvMotL** block tracks the control that belongs to a non-existent feedback signal with a delay of `V_MonTiDynamic` seconds. For example, if the `NoFbkClose =1` closed feedback is not present, opening `FbkOpen =1` is immediately tracked with `Open =1`, while closing `FbkOpen =0` is tracked with a delay of the valve run time `V_MonTiDynamic`.
- Parameter comments have been added to the blocks **StrgToBy**, **StrctCom**, **StrctDeC**, **StruCmIn**, and **StrCmOu**.
- The comments of the `SelFp1, SelFp2` inputs at the **Event, Event16Ts, EventTs, EventNck** blocks have been set to RESERVED.
- Channel block **Pcs7AnOu**: The SIMATIC PDM Maintenance State `MS=16#7xxx xxxx` now only results in Bad for HART channels.
- The following improvements have been made to the **VlvAnL** block:
  - The **manipulated variable difference** (`MV-Rbk`) can also be monitored in the end position ranges by configuring `Feature2 Bit15 =1`. To avoid monitoring errors, adjust the `MV_HiLim, MV_LoLim` manipulated variable limits in such a way that the position feedback `Rbk` can also reliably reach the end positions and that no impermissibly large (`MV-Rbk`) differences can occur in the end positions.
  - Reliable detection of `PosReached` and **avoidance of monitoring errors during operation without digital feedback** (`NoFbkOpen` and `NoFbkClose=1`).
  - **SafePos calculation** in status word 2 for `SafePos=2` (Stop) revised.
  - Improvement of the feedback monitoring with `MV_RateOn =1` **gradient limit**.
  - For use cases with **small intermediate ranges** where `PosDiOpen` and `PosDiClose` are close together, the `PosReached` calculation has been improved.
- The following improvements have been made to the **VlvMotL** block:
  - Improvement of the feedback monitoring and the reset response in the end positions.
  - Improvement of the automatic preview display for the standard view and preview.
- At the **MotSpdCL** block, the delay factors are hidden in the parameter view as well with `ER_AL_EN =0` and `ER_HL_EN =0`.

- At the **DoseL** block, the "**Invalid signal**" display in the standard view has been improved and calculation of the `ErrorNum` output has been adjusted in the event of `SP_LiOp=1, SP_IntLi=1 & SP_ExtLi=1`.
- **SplRange** block: The check for negative infinite REAL numbers (#-INF) at the `In` input now leads to setting error number 30 at the `ErrorNum` output. If there is an invalid REAL number at the input `In` (#NaN or #INF), no further calculation occurs. Instead, the last values at the output are retained and the error number `ErrorNum=30` is output.
- **TimerP** block: After changing from an invalid `Ti < SampleTime` value to a valid `Ti >= SampleTime` value, the outputs are only adopted after a `0->1` change at the `In` I/O.
- At the **SelD02In** block, the comments of the `Sel_In2` and `In2Selected` parameters have been changed.
- The parameter comment from `In` was added to the **StrctDeC** block. This means the **GainSched** block created via `CFC` was also updated.

5.2.3 Update information for Version 9.1 SP1

The following applies to the software update from version 9.1 to version 9.1 SP 1:

1.	Make text interconnections	**Yes**
2.	Compile AS	**Complete compilation**
3.	Download AS program	**Complete download**
4.	AS STOP necessary	**Yes (AS STOP is not necessary when using a CPU 410-5H with TCiR)**
5.	Compile OS	**Complete compilation**

Note

Updating the blocks from the PCS 7 Advanced Process Library with TCiR

"Type import - Changed default value" dialog box

- If information on changed internal local default values in the static area is displayed in the dialog box during type import, these changes are not relevant. A complete download in STOP mode is not necessary.
- If information on changed default values is displayed in the dialog box at block I/Os during type import, these new default values must be applied in the "Update block types" dialog.

List of modified blocks

Name	Number	Block version	Interface change (AS STOP is not necessary when using a CPU 410-5H with TCiR)	New block contact for SFC type	Code change
CntOhSc	FB1803	5.6	No	No	Yes
ConPerMon	FB1805	5.6	No	No	Yes
CountScL	FB1806	5.6	No	No	Yes
CountOh	FB1864	5.6	No	No	Yes
DoseL	FB1809	5.6	No	No	Yes
Event	FB1811	5.6	No	No	Yes
Event16Ts	FB1887	5.6	No	No	Yes
EventNck	FB1904	5.6	No	No	Yes
EventTs	FB1812	5.6	No	No	Yes
FbAnIn	FB1813	9.6	No	No	Yes
FbAnOu	FB1814	9.6	No	No	Yes
FbAnTot	FB1817	9.6	No	No	Yes
FbDiIn	FB1815	9.6	No	No	Yes
FbDiOu	FB1816	9.6	No	No	Yes
FbDrive	FB1905	9.6	No	No	Yes
FbEnMe	FB1908	9.6	No	No	Yes
FbSwtMMS	FB1907	9.6	No	No	Yes
FmCont	FB1818	9.6	No	No	Yes
FmTemp	FB1819	9.6	No	No	Yes
GainSche	FB1820	9.6	No	No	Yes
Intlk02	FB1824	5.6	No	No	Yes
Intlk04	FB1825	5.6	No	No	Yes
Intlk08	FB1826	5.6	No	No	Yes
Intlk16	FB1827	5.6	No	No	Yes
KalFilt	FB1925	9.5	No	No	Yes
ModPreCon	FB1843	5.6	No	No	Yes
MonAnL	FB1845	5.6	No	No	Yes
MonAnS	FB1912	5.6	No	No	Yes
MonDi08	FB1847	5.6	No	No	Yes
MonDiL	FB1848	5.6	No	No	Yes
MonDiS	FB1913	5.6	No	No	Yes
MotL	FB1850	5.6	Yes	No	Yes
MotRevL	FB1851	5.6	Yes	No	Yes
MotS	FB1910	5.6	No	No	Yes
MotSpdCL	FB1854	5.6	Yes	No	Yes
MotSpdL	FB1856	5.6	Yes	No	Yes
MPC10x10	FB1920	5.6	No	No	Yes

OpAnL	FB1865	5.6	No	No	Yes
OpAnS	FB1915	5.6	No	No	Yes
OpDi01	FB1866	5.6	No	No	Yes
OpDi03	FB1867	5.6	No	No	Yes
OpStations	FB1901	5.6	No	No	Yes
OpTrig	FB1868	5.6	No	No	Yes
Pcs7AITC	FB1933	9.6	No	No	Yes
Pcs7AnIn	FB1869	9.6	Yes	No	Yes
Pcs7AnOu	FB1870	9.6	No	No	Yes
Pcs7Cnt1	FB1833	9.6	No	No	Yes
Pcs7Cnt2	FB1834	9.6	No	No	Yes
Pcs7Cnt3	FB1835	9.6	No	No	Yes
Pcs7Cnt4	FB1836	9.6	No	No	New block
Pcs7DiCn	FB1935	9.6	No	No	Yes
Pcs7DiIn	FB1871	9.6	No	No	Yes
Pcs7DiIT	FB1872	9.6	No	No	Yes
Pcs7DiOu	FB1873	9.6	No	No	Yes
Pcs7HaAI	FB1931	9.6	No	No	Yes
Pcs7HaAO	FB1932	9.6	No	No	Yes
Pcs7ReIn	FB1837	9.6	No	No	New block
PIDCoefR	FC366	5.6	No	No	Yes
PIDConL	FB1874	5.6	No	No	Yes
PIDConR	FB1875	5.6	No	No	Yes
PIDConS	FB1830	5.6	No	No	Yes
PIDStepL	FB1878	5.6	No	No	Yes
PrDrpMon	FB1952	5.6	Yes	No	Yes
PumpMon	FB1950	5.6	Yes	No	Yes
PST	FB1956	5.6	No	No	Yes
Ratio	FB1883	5.6	No	No	Yes
SelA16In	FB1888	5.6	No	No	Yes
SelAn16L	FB1889	5.6	No	No	Yes
SelD02In	FC391	9.6	No	No	Yes
ShrdResL	FB1917	5.6	No	No	Yes
ShrdResS	FB1914	5.6	No	No	Yes
SplRange	FC372	5.6	No	No	Yes
SteadyState	FB1955	5.6	No	No	Yes
StrctCom	FC393	5.6	No	No	Yes
StrctDeC	FC396	5.6	No	No	Yes
StrctDeC	FC396	5.6	No	No	Yes
StruCmIn	FC446	5.6	No	No	Yes
StruCmOu	FC447	5.6	No	No	Yes
TimerP	FB1810	5.6	Yes	No	Yes
TimeTrig	FB1802	5.6	No	No	Yes
TotalL	FB1906	5.6	No	No	Yes

Vlv2WayL	FB1897	5.6	No	No	Yes
VlvAnL	FB1896	5.6	Yes	No	Yes
VlvL	FB1899	5.6	Yes	No	Yes
VlvMon	FB1951	5.6	Yes	No	Yes
VlvMotL	FB1900	5.6	Yes	No	Yes
VlvPosL	FB1918	5.6	Yes	No	Yes
VlvS	FB1911	5.6	No	No	Yes

Note

- Blocks for which only attributes (except S7_contact), message texts, and/or version number were changed, are not listed.
- If the "**New block contact for SFC type**" column has a "**Yes**", this means that either the block type has a new parameter with "S7_contact=true" or the attribute S7_contact auf "true" has been set to an existing parameter. In both cases this results in an extension of the block I/Os and thus an interface change at the SFC type.
- If "Yes" is listed in the "Interface change" column and "No" in the "Code change" column, this indicates that either the interfaces have been prepared for a future function or the order of the static variables (VAR area) has been optimized.

5.3 Version 9.1 SP2

5.3.1 What's new in Version 9.1 SP2?

This section describes the new features compared to the previous version.

- For Modbus communication via ET 200SP CM PtP (6ES7137-6AA00-0BA0 FW V1.0.3) a new **MBComm** channel block and new communication blocks **MBWrReg**, **MBWrCoil**, **MBRdReg**, **MBRdCoil, MBRcvReg, MBPdReg** are available.
  These blocks use function blocks of Modbus RTU Library V1.3, whose blocks have been adopted in the number range of the APL according to the following scheme:

Modbus RTU Library V1.3		**APL FB number**
Modbus_Comm_Load	FB640	FB1996
Modbus_Master	FB641	FB1997
Modbus_Slave	FB642	FB1998
Send_Config	FB611	FB1991
Receive_Config	FB612	FB1992
Send_P2P	FB613	FB1993
Receive_P2P	FB614	FB1994
Receive_Reset	FB617	FB1995

  You can find information about Modbus RTU Library V1.3 in https://support.industry.siemens.com/cs/document/75226762/

  Note

  Adapting the Modbus configuration

  The change of an existing Modbus communication setting via the `Configure` input at the **MBComm block** only takes effect after **AS-STOP** or pulling and plugging the CM.

- The**FbAnOu** channel block supports the substitute value specification for the actuator position feedback `Rbk`. `Feature.Bit29` can be used to activate the substitute value specification in the event of bad actuator position feedback.
  Bit = 0: If there is invalid actuator position feedback at the `Rbk` input, no `SubsRbk` substitute value is output.
  Bit = 1: If there is invalid actuator position feedback at the `Rbk` input, the substitute value `SubsRbk` is output. The signal status of the output value is set to "Manipulated value" (16#60).
- The **Pcs7AnIn** channel block supports the updated version of **NAMUR recommendation NE 43** from July 26, 2021. For this purpose, the safety margin defined in the NAMUR recommendation can be specified at the channel module for the upper and lower current ranges at the `HiSafteyMrg` and `LoSafetyMrg` inputs. The presetting of the safety limits is in accordance with the NAMUR recommendation:
  ```
  HiSafteyMrg = 0.2
  LoSafetyMrg = 0.1
  ```
  This results in a **change in behavior** with regard to the calculation of the **failure identifier** for **4..20 mA signals**.
  Previously, a signal failure (output `Bad=1`) was detected at `PV > 21.0 mA` or `PV < 3.6 mA`.
  According to the new NE 43, signal failure identification is carried out as follows depending on the new safety limits:
  - `PV ≥ (21.0 mA - HiSafetyMrg)` or `PV ≤ (3.6 mA + LoSafetyMrg)`

5.3.2 Changes in Version 9.1 SP2

- The version was adapted at the **BlkTypeInfo** parameter on the following blocks: **CntOhSc, ConPerMon, CountOh, CountScL, DoseL, FmCont, FmTemp, Intlk02..16, KalFilt, ModPreCon, MonAnL, MonAnS, MonDi08, MonDiL, MonDiS, MotL, MotRevL, MotS, MotSpdCL, MotSpdL, MPC10x10, OpAnL, OpAnS, OpDi01, OpDi03, OpStations, OpTrig, PIDConL, PIDConR, PIDConS, PIDStepL, PressDropMon, PST, PumpMon, Ratio, SelA16In, SelAn16L, ShrdResL, ShrdResS, SteadyState, TimeTrig, TotalL, Vlv2WayL, VlvAnL, VlvL, VlvMon, VlvMotL, VlvPosL, VlvS**
- The behavior for direction-dependent interlock `Feature2 Bit16 =1` on the **MotRevL, ModSpdL, MotSpdCL, VlvMotL, VlvPosL, VlvAnL** blocks was extended with `Feature2 Bit14`. When attempting to change to a locked motor direction of rotation / speed or valve position, the drive behaves as follows:
  `Bit =0`: The drive remains in the present direction of rotation / speed / valve position.
  `Bit =1`: The drive changes to the Stop position.
- The configurable reset response of the **MotRevL, ModSpdL, VlvMotL, VlvPosL** blocks has been improved when the "Protection" or "Motor protection" interlock via `Feature Bit19` is pending.
- The reset response via the `XxxLocal` command inputs for "Motor protection" in local mode `LocalMode =1/3` has been adapted on the **VlvMotL, VlvPosL** blocks:
  Resetting via the `XxxLocal` command inputs for "Motor protection" is performed via a 0->1 trigger at the `StopLocal` input.
- The reset response via the **VlvMotL, VlvPosL** automatic commands at the `Feature Bit9 =1` blocks has been adapted:
  A reset is only performed from the idle state with the corresponding automatic command `OpenAut, CloseAut` that leads out of the idle state.
- At the **VlvMotL, VlvPosL** blocks, bumpless switchover to automatic mode `Feature Bit17 =1` of invalid automatic commands `Feature2 Bit3` and to automatic commands with bad status `Feature2 Bit10` has been extended.
- When using the "Sealing the valve" function `Feature.Bit8=1` at the **VlvMotL, VlvPosL** blocks, resetting of the monitoring error is allowed in the intermediate state i.e. `FbkClose = 1 & TorqClose = 1` at **VlvMotL** or `Rbk < PosDiClose , FbkClose = 1 & TorqClose = 1` at **VlvPosL** is allowed. After the reset, `FbkClose` monitoring is resumed and an open/close command can be issued during static monitoring, allowing continued operation of the valve (especially in the direction of the closed state).
- Improved control response when deactivating the Stop command `OS_Perm.Bit4 =0` has been improved at the **VlvMotL** block.
- The display of the feedback monitoring has been improved at the **VlvMotL** block.

- On the **VlvMotL** block, the valve position is also tracked in the "Out of service" mode `OosAct =1` when "Motor stop"`SafePos =2` is parameterized at rest. As a result, the `FbkOpenOut, FbkCloseOut` outputs are also updated when not in operation. For this purpose, the mini symbol for the resting position is updated in the standard view and the "Feedback open" and "Feedback close" displays in the preview. In the case of feedback error FbkOpenOut =1, FbkOpenOut=1, the last valid position is displayed on the mini symbol, which is also used for further control.
- The tracking of the analog manipulated variable when the position control is switched off has been improved on the **VlvPosL** block. If the approach to the idle state `SafePos =0/1` is blocked by the torque monitoring `TorqOpen, TorqClose` when an error occurs, the analog manipulated variable is tracked to the readback value.
- Sealing `Feature.Bit8 =1` has been improved at the **VlvPosL** block when the position control is switched off.
- At the **VlvAnL** block with auxiliary valve `Feature.Bit5 =1`, the auxiliary valve can also be controlled with `Feature.Bit6 =1`.
- At the **VlvAnL** block, tracking `MV` to `Rbk` when an error occurs with `SafePos =2` is limited to the manipulated variable limits `MV_HiLim/MV_LoLim`.
- Monitoring of the feedback signals has been improved on the **Vlv2WayL** block. After a monitoring error for maintaining the position (static), monitoring is no longer started when changing to the idle state.
- The switchover behavior Fast Slow Fast with configured switchover time `SwOverTime` has been changed at the **MotSpdL** block. It is possible to switch back to Fast without having to wait for the switchover time.
- The **OpAnS** block is no longer limited to `SP_OpScale` and thus corresponds to the documented functionality for Small blocks.
- At the **TimeTrig** block, the `SummerTime` output indicates whether daylight saving time is active for the currently set time (external, AS, local). Local time is used if the external time is switched off `Feature Bit6 =0` and the local time is switched on `LocalTime =1`.

- For the **TimerP** block, the reset response for `Mode=2` (start timer with switch-on delay) has been improved with an upgrade via **TCiR**. For this, the defaults of an internal tag and thus also the **interface time stamp** was changed.

  Note

  If you are already using block version 5.6 from the V9.1.1 library, it is not absolutely necessary to upgrade the TimerP block to block version 5.7 from V9.1.2.

- Channel blocks
  - Setting invalid process values PV_LoAct, PV_HiAct and Bad at the range limits for symmetrical physical value ranges has been improved at the **Pcs7AnIn** block.
  - At the **FbAnOu** block, if the raw value and `Feature.Bit31 =1` are invalid, the last valid value of `RCasOutLi, Rbk und PosD` is held and the value status is set to manipulated variable 16#60.
  - At the **FbAnOu** block with `Feature.Bit 28=0, Feature.Bit31 =1`, the `RbkLi` output is held if there is a bad signal status at the `Rbk.ST, PosD, RCasOut` parameters.
  - At the **Pcs7AnOu** block, the behavior at the `Bad`, `PV_ChnST` outputs is adapted for `SimOn =1` and thus corresponds to the behavior of **Pcs7AnIn, Pcs7DiIn** and **Pcs7DiOu**.

5.3.3 Update Information for Version 9.1 SP2

The following applies to the software update from version 9.1 SP1 to version 9.1 SP2:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Complete compilation**
3.	Download AS program	**Complete download**
4.	AS STOP necessary	**Yes (AS STOP is not necessary if a CPU 410-5H with TCiR is used)**
5.	Compile OS	**Complete compilation**

Note

Updating the blocks from the PCS 7 Advanced Process Library with TCiR

"Type import - Changed default value" dialog box

- If information on changed internal local default values in the static area is displayed during type import in the dialog box, these changes are not relevant. A complete download in STOP mode is not necessary.
- If information on changed default values is displayed at block I/Os during type import in the dialog box, these new default values must be applied in the "Update Block Types" dialog.

List of changed blocks

Name	Number	Block version	Interface change (AS STOP is not necessary if a CPU 410-5H with TCiR is used)	New block contact for SFC type	Code change
CntOhSc	FB1803	5.7	No	No	Yes
ConPerMon	FB1805	5.7	No	No	Yes
CountScL	FB1806	5.7	No	No	Yes
CountOh	FB1864	5.7	No	No	Yes
DoseL	FB1809	5.7	No	No	Yes
FbAnIn	FB1813	9.7	No	No	No
FbAnOu	FB1814	9.7	No	No	Yes
FbAnTot	FB1817	9.7	No	No	No
FbDiIn	FB1815	9.7	No	No	No
FbDiOu	FB1816	9.7	No	No	No
FbDrive	FB1905	9.7	No	No	No
FbEnMe	FB1908	9.7	No	No	No
FbSwtMMS	FB1907	9.7	No	No	No
FmCont	FB1818	9.7	No	No	Yes
FmTemp	FB1819	9.7	No	No	Yes
Intlk02	FB1824	5.7	No	No	Yes
Intlk04	FB1825	5.7	No	No	Yes
Intlk08	FB1826	5.7	No	No	Yes
Intlk16	FB1827	5.7	No	No	Yes
KalFilt	FB1925	9.7	No	No	Yes
MBCOMLOA	FB1996	1.0	No	No	New block
MBComm	FB1941	9.7	No	No	New block
MBMASTER	FB1997	1.0	No	No	New block
MBPdReg	FB1947	9.7	No	No	New block
MBRcvReg	FB1946	9.7	No	No	New block
MBRdCoil	FB1942	9.7	No	No	New block
MBRdReg	FB1943	9.7	No	No	New block
MBSLAVE	FB1998	1.0	No	No	New block
MBWrCoil	FB1944	9.7	No	No	New block
MBWrReg	FB1945	9.7	No	No	New block
ModPreCon	FB1843	5.7	No	No	Yes
MonAnL	FB1845	5.7	No	No	Yes
MonAnS	FB1912	5.7	No	No	Yes
MonDi08	FB1847	5.7	No	No	Yes
MonDiL	FB1848	5.7	No	No	Yes
MonDiS	FB1913	5.7	No	No	Yes
MotL	FB1850	5.7	No	No	Yes

MotRevL	FB1851	5.7	No	No	Yes
MotS	FB1910	5.7	No	No	Yes
MotSpdCL	FB1854	5.7	No	No	Yes
MotSpdL	FB1857	5.7	No	No	Yes
MPC10x10	FB1920	5.7	No	No	Yes
OpAnL	FB1865	5.7	No	No	Yes
OpAnS	FB1915	5.7	No	No	Yes
OpDi01	FB1866	5.7	No	No	Yes
OpDi03	FB1867	5.7	No	No	Yes
OpStations	FB1901	5.7	No	No	Yes
OpTrig	FB1868	5.7	No	No	Yes
Pcs7AITC	FB1933	9.7	No	No	No
Pcs7AnIn	FB1869	9.7	No	No	Yes
Pcs7AnOu	FB1870	9.7	No	No	Yes
Pcs7Cnt1	FB1833	9.7	No	No	No
Pcs7Cnt2	FB1834	9.7	No	No	No
Pcs7Cnt3	FB1835	9.7	No	No	No
Pcs7Cnt4	FB1836	9.7	No	No	New block
Pcs7DiCnt	FB1935	9.7	No	No	No
Pcs7DiIn	FB1871	9.7	No	No	No
Pcs7DiIT	FB1872	9.7	No	No	No
Pcs7DiOu	FB1873	9.7	No	No	No
Pcs7HaAI	FB1931	9.7	No	No	No
Pcs7HaAO	FB1932	9.7	No	No	No
Pcs7ReIn	FB1837	9.7	No	No	New block
PIDConL	FB1874	5.7	No	No	Yes
PIDConR	FB1875	5.7	No	No	Yes
PIDConS	FB1830	5.7	No	No	Yes
PIDStepL	FB1878	5.7	No	No	Yes
PrDrpMon	FB1952	5.7	No	No	Yes
PumpMon	FB1950	5.7	No	No	Yes
PST	FB1956	5.7	No	No	Yes
Ratio	FB1883	5.7	No	No	Yes
RcvConf	FB1992	1.0	No	No	New block
RcvP2P	FB1994	1.0	No	No	New block
RcvRst	FB1995	1.0	No	No	New block
SelA16In	FB1888	5.7	No	No	Yes
SelAn16L	FB1889	5.7	No	No	Yes
SendConf	FB1991	1.0	No	No	New block
SendP2P	FB1993	1.0	No	No	New block
ShrdResL	FB1917	5.7	No	No	Yes
ShrdResS	FB1914	5.7	No	No	Yes
SteadyState	FB1955	5.7	No	No	Yes
TimerP	FB1810	5.7	Ja	No	Ja

TimeTrig	FB1802	5.7	No	No	Yes
TotalL	FB1906	5.7	No	No	Yes
Vlv2WayL	FB1897	5.7	No	No	Yes
VlvAnL	FB1896	5.7	No	No	Yes
VlvL	FB1899	5.7	No	No	Yes
VlvMon	FB1951	5.7	No	No	Yes
VlvMotL	FB1900	5.7	No	No	Yes
VlvPosL	FB1918	5.7	No	No	Yes
VlvS	FB1911	5.7	No	No	Yes

Note

- Blocks for which only attributes (except S7_contact), message texts, and/or version number were changed, are not listed.
- If the column "**New block contact for SFC type**" has a "**Yes**", this means that either the block type has a new parameter with "S7_contact=true" or an existing parameter has been set to the attribute S7_contact auf "true" . In both cases this results in an extension of the block I/Os and thus an interface change at the SFC type.
- If there is a "No" in the "Interface change" column and a "No" in the "Code change" column, the version has been raised in the channel blocks.
- The TCiR behavior was improved for TimerP, this required adjusting an interface. The change is only relevant for users of a CPU 410-5H who want to upgrade from a version <= V9.1 to the version V9.1 SP2 using TCiR. For users who do not use CPU 410-5H and have already upgraded to V9.1 SP1, the change is not relevant and the TimerP block does not have to be upgraded in this case.

5.4 Version 9.1 SP2 Update 1

5.4.1 Changes in Version 9.1 SP2 Update 1

- In the PCS 7 measuring point browser, unused limit values are no longer displayed in the "Suppressed messages" filter. To this end, the following blocks have been modified: **ConPerMon, CountOh, CountScL, DoseL, FmCont, FmTemp, MonAnL, MonAnS, MonDiL, MotL, MotRevL, MotSpdCL, MotSpdL, PIDConL, PIDConR, PIDConS, PIDStepL, PumpMon, TotalL, VlvAnL, VlvMotL, VlvPosL**.
- During the delay time `DelTiBad`, the last valid process output value is retained regardless of the `Feature Bits 28, 29, 30`. To this end, the following blocks have been modified: **FbAnIn, FbAnTot, FbDiIn, Pcs7AITC, Pcs7AnIn, Pcs7Cnt1, Pcs7Cnt2, Pcs7Cnt3, Pcs7Cnt4, Pcs7DiCnt, Pcs7DiIn, Pcs7DiIT, Pcs7ReIn**.
- At the **MotSpdL** block, switching in automatic mode to Speed 2 `Spd2Aut` with direction-dependent interlocking `Feature2 Bit16 =1` has been improved.

- At the **MotSpdL** block, switching to Speed 2 during switch operation and direction-dependent interlocking `Feature2 Bit16 =1` has been improved.
- At the **VlvAnL** block, direct switching between open and closed to a locked position has been improved in manual mode with `Feature2 Bit16 =1` and `Feature2 Bit14 =1`.
- The control response in automatic mode has been improved at the **VlvPosL** block, with `Feature2 Bit16 =1` when switching directly in a locked direction.

Note

With the introduction of `Feature2 Bit14` in APL 9.1.2, the standard behavior of the motor valve **VlvPosL** with position control switched on was also adapted.

PREVIOUSLY:
When the command direction is **changed to a locked direction**, the motor valve moves to the stop position when position control is active. The manipulated variable is updated `MV_Out = RbkOut`.
NEW:
With `Feature2 Bit14 =0` (default setting), the actuator continues to run with the current manipulated variable output.

NOTICE
As a result of harmonization, there is a change in behavior for the VlvPosL. **If you want to retain the previous behavior of the block, you must set the** `Feature2 Bit14` **to** `1` **when upgrading.**

- The **control response** with deactivated Stop command `OS_Perm.Bit4 =0` has been improved at the **VlvPosL** block.
- With **DoseL** in automatic mode with `Feature Bit4 =1` (switch operation), a continuation of dosing is now only achieved by a negative edge change 1->0 at the `PauseAut` input. This prevents, for example, unintentional continuation after underdosing `Feature Bit12 =0`.

NOTICE
Due to this improvement, the interconnection logic may have to be adjusted. Check the interconnection of the input parameter `PauseAut` during switch operation to ensure that a 1->0 edge change is now necessary for continuation.

- At the **PIDConL, PIDConS, PIDConR, PIDStepL, FmCont, FmTemp** blocks, a parameterization `DiffGain <1` or `DiffGain >10` does not lead to an error indication at the output `ErrorNum =62` when the D-component (`DiffSel =0, TD =0`) is switched off.

- At the **PIDConL, PIDConS und PIDStepL** blocks, the behavior in the event of **#NaN values** occurring in automatic mode has been improved. **- The output of #NaN values at the process value output `MV` is now prevented more reliably.**
  - #NaN values at ordinary REAL value inputs, such as at the `NormPV` or `Gain` input, no longer lead
  to the saving of the #NaN values at the `P_Part, I_Part, D_Part` outputs or at the written-back `Man` input.

  Note

  Basically, only process values (these are values that are typically acquired via a channel block or derived from such) are checked for valid REAL values during program runtime. In the "Error handling" section, you can use the error number table to determine which process values are subject to such a check.

- At the **OpDi03** block, the `Out1, Out2, Out3` outputs are reset in "Out of service" mode and are applied after exiting "Out of service" mode.
- Channel blocks
  - Correction at the block **FbEnMe** when calculating the output parameters `PL1Li.Value, PL2Li.Value, PL3Li.Value, PTotLi.Value, AcPwrLi.Value, RePwrLi.Value, ApPwrLi.Value`
  - Extension to the **Pcs7ReIn** block to support additional mode values

5.4.2 Update Information for Version 9.1 SP2 Update 1

The following applies to the software update from Version 9.1 SP 2 to Version 9.1 SP 2 Update 1:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Not necessary**

List of changed blocks

Name	Number	Block version	Interface change	Code change
ConPerMon	FB1805	5.7	No	Yes
CountOh	FB1864	5.7	No	Yes
CountScL	FB1806	5.7	No	Yes

DoseL	FB1809	5.7	No	Yes
FbAnIn	FB1813	9.7	No	Yes
FbAnTot	FB1817	9.7	No	Yes
FbDiIn	FB1815	9.7	No	Yes
FbEnMe	FB1908	9.7	No	Yes
FmCont	FB1818	9.7	No	Yes
FmTemp	FB1819	9.7	No	Yes
MonAnL	FB1845	5.7	No	Yes
MonAnS	FB1912	5.7	No	Yes
MonDiL	FB1848	5.7	No	Yes
MotL	FB1850	5.7	No	Yes
MotRevL	FB1851	5.7	No	Yes
MotSpdCL	FB1854	5.7	No	Yes
MotSpdL	FB1857	5.7	No	Yes
OpDi03	FB1867	5.7	No	Yes
Pcs7AITC	FB1933	9.7	No	Yes
Pcs7AnIn	FB1869	9.7	No	Yes
Pcs7Cnt1	FB1833	9.7	No	Yes
Pcs7Cnt2	FB1834	9.7	No	Yes
Pcs7Cnt3	FB1835	9.7	No	Yes
Pcs7Cnt4	FB1836	9.7	No	Yes
Pcs7DiCnt	FB1935	9.7	No	Yes
Pcs7DiIn	FB1871	9.7	No	Yes
Pcs7DiIT	FB1872	9.7	No	Yes
Pcs7ReIn	FB1837	9.7	No	Yes
PIDConL	FB1874	5.7	No	Yes
PIDConR	FB1875	5.7	No	Yes
PIDConS	FB1830	5.7	No	Yes
PIDStepL	FB1878	5.7	No	Yes
PumpMon	FB1950	5.7	No	Yes
TotalL	FB1906	5.7	No	Yes
VlvAnL	FB1896	5.7	No	Yes
VlvMotL	FB1900	5.7	No	Yes
VlvPosL	FB1918	5.7	No	Yes

Note

The comment [Release 09.01.02.01] has been inserted in all changed blocks.

5.5 Version 9.1 SP2 Update 2

5.5.1 Changes in Version 9.1 SP2 Update 2

- The direction-independent interlock `(Feature2 Bit16 =0)` has been corrected at the **VlvAnL** block without auxiliary valve `(Feature Bit5 =0)`. The valve goes into neutral position even if there is a pending OpenAut =1 or CloseAut =1 command.
- The direction-dependent interlock `(Feature2 Bit16 =1)` has been corrected at the **VlvAnL** block with consideration of `Feature2 Bit14`.
- At the **VlvAnL** block, the `ER_AH_Act, ER_AL_Act` outputs for monitoring the difference between manipulated variable and position feedback are set regardless of whether the position feedbacks `RbkOut, FbkOpenOut, FbkCloseOut` are still in the neutral position.
- The control behavior with switched-off feedback signal monitoring has been improved at the **MotL, MotRevL, MotSpdCL, MotSpdL, MotSpd3L, VlvMotL** and **VlvPosL** blocks.
- At the **VlvPosL** block, the monitoring behavior of the valve feedbacks with control in the minimum pulse duration phase `Breaktime` has been improved.
- At the **VlvMotL, VlvPosL** blocks, interlock protection and errors can be reset again via the automatic commands (e.g. OpenAut, CloseAut) and `Feature Bit9`.
- The command for starting the drive from the neutral position can take place at the **MotL, MotRevL, MotSpdL, MotSpdCL, VlvMotL** and **VlvPosL** blocks, even during the restart inhibit. After the restart inhibit has elapsed, they become active at the outputs for controlling the drive.
- The **Pcs7AnIn** channel block has been expanded by the following modes for support of the module ET 200SP HA AI 8xU/I/TC/4xRTD 2-/3-/4-wire HA (6DL1134-6AF00-0PH1):
  - 16#08A2 Cu50 (GOST climate range TC = 428) thermistor, linear, 4-wire
  - 16#08A3 Cu100 (GOST climate range TC = 428) thermistor, linear, 4-wire
  - 16#09A2 Cu50 (GOST climate range TC = 428) thermistor, linear, 3-wire
  - 16#09A3 Cu100 (GOST climate range TC = 428) thermistor, linear, 3-wire
  - 16#0F9F Cu50 (GOST standard range TC = 428) thermistor, linear, 2-wire
  - 16#0FA1 Cu100 (GOST standard range TC = 428) thermistor, linear, 2-wire
  - 16#0FA2 Cu50 (GOST climate range TC = 428) thermistor, linear, 2-wire
  - 16#0FA3 Cu100 (GOST climate range TC = 428) thermistor, linear, 2-wire

  Note

  At least PCS 7 Basis Library V9.1.2.2 is required to use these new modes.
- Improvement at the **PCS7AITC** channel block to prevent toggling of the Bad output parameter with Feature.Bit29=1 (substitute value)
- The link to the Japanese online help has been updated.

5.5.2 Update information for Version 9.1 SP2 Update 2

The following applies to the software update from Version 9.1 SP 2 Update 1 to Version 9.1 SP 2 Update 2:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Not necessary**

List of changed blocks

Name	Number	Block version	Interface change	Code change
Pcs7AnIn	FB1869	9.7	No	Yes
Pcs7AITC	FB1933	9.7	No	Yes
VlvAnL	FB1896	5.7	No	Yes
VlvMotL	FB1900	5.7	No	Yes
VlvPosL	FB1918	5.7	No	Yes

Note

The comment [Release 09.01.02.02] has been inserted in all changed blocks.

5.6 Version 9.1 SP2 Update 3

5.6.1 Changes in Version 9.1 SP2 Update 3

- Improvements for the channel block **Pcs7AITC**
  - to prevent toggling of the `Bad` output parameter with `Feature.Bit29`=1 (substitute value) in REAL mode
  - to set the PV_LoAct and PV_HiAct output parameters depending on fixed Scale values in the WORD and REAL mode
- The REAL limit values of `PV_In` are now detected at **Pcs7ReIn** and, when reached, output with `Bad` =1.

- Robustness measure at **PCS7AnIn** so that an incorrect process value does not result even when a different measuring range than `4..20mA NamurOff`=1 is used.
- In the VlvAnl, the monitoring of the feedback in the intermediate and end position is limited by the Feature bit in the `LocalSetting` =2/4
  `Feature2.Bit24` = 0 Feedback monitoring with existing behavior (default)
  `Feature2.Bit24` = 1 Feedback monitoring only occurs when (FbkOpen = 1 AND FbkClose = 1) OR (FbkOpen = 1 AND RBK < PosDiClose) OR (FBKClose = 1 AND RBK> PosDiOpen)
- After the switchover to local operation LocalMode =2/4, the control signal is now reset at **VlvMotL** without motor feedback `Feature Bit12` =1 and without valve feedback `NoFbkOpen` =1 or `NoFbkClose` =1.
- **Event16Ts** has been improved to display the batch name for all 16 messages.
- **CntOhSc**: Improved calculation of TimeHour on AS restart when `Feature.Bit5` = 1

5.6.2 Update information for Version 9.1 SP2 Update 3

The following applies to the software update from Version 9.1 SP 2 Update 2 to Version 9.1 SP 2 Update 3:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Not necessary**

List of changed blocks

Name	Number	Block version	Interface change	Code change
Pcs7ReIn	FB1837	9.7	No	Yes
Pcs7AnIn	FB1869	9.7	No	Yes
Pcs7AITC	FB1933	9.7	No	Yes
VlvAnL	FB1896	5.7	No	Yes
VlvMotL	FB1900	5.7	No	Yes
Event16Ts	FB1887	5.0	No	Yes
CntOhSc	FB1803	5.7	No	Yes

Note

The comment [Release 09.01.02.03] has been inserted in all changed blocks.

5.7 Version 9.1 SP2 Update 4

5.7.1 Changes in Version 9.1 SP2 Update 4

- At the DoseL block, the Error Number 12 (Configuration error) generation has been improved when the alarms are disabled.
- In MotL, MotRevL, MotSpdCL, MotSpdL, MotSpd3L, VlvMotL blocks the start command given during Idle Time will be executed after the `IdleTime` has expired.
- At MonDiL & MonDiS block the input InvIn is considered when switching to simulation mode for bump less switchover and at the MonDiL block `SimInLi` & `BypIn` can be inverted with InvIn when `Feature.Bit26` = 1.
- For Interlock blocks (Intlk02, Intlk04, Intlk08, Intlk16), if Feature.Bit4 is active, Interlock get higher priority than bypass.
- In Channel blocks if a Quality Information (QI) from the ET 200SP HA module is provided, the input parameter ProImQB is interconnected with the corresponding symbol and the input parameter SelQB is set to 1 (otherwise 0).
- In PCS7AiTC Block, bad PV input is passed to the `PV_out` only after the Delay time has expired

5.7.2 Update information for Version 9.1 SP2 Update 4

The following applies to the software update from Version 9.1 SP 2 Update 3 to Version 9.1 SP 2 Update 4:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Not necessary**

List of changed blocks

Name	Number	Block version	Interface change	Code change
DoseL	FB1089	7.1	No	Yes
FbAnOu	FB1814	9.7	No	Yes
Intlk02	FB1824	5.7	No	Yes
Intlk04	FB1825	5.7	No	Yes
Intlk08	FB1826	5.7	No	Yes
Intlk16	FB1827	5.7	No	Yes
MonDiL	FB1848	5.7	No	Yes
MonDiS	FB1913	5.7	No	Yes
Pcs7AITC	FB1933	9.7	No	Yes
Pcs7ReIn	FB1837	9.7	No	Yes

Note

The comment [Release 09.01.02.04] has been inserted in all changed blocks.

5.8 Version 9.1 SP2 Update 5

5.8.1 Changes in Version 9.1 SP2 Update 5

- In **MotL, MotRevL, MotSpdCL, MotSpdL** and **VlvL** blocks, `Feature2.Bit21` is extended to cover two and single plausible feedback conditions in Local Mode 2/4.
- In Drives family blocks the resetting of the non-feedback errors does not require respective feedback to the control command when the monitoring of the feedback functionality is disabled.
- With the interlock blocks **Intlk02, Intlk04, Intlk08** and **Intlk16**, `Feature.Bit3` can be used to specify that a bypassed measuring point leads directly to bypassing the associated interlock input. The `BypEn` input at the interlock block is routed accordingly and no longer has to be parameterized manually.
- Improvement of **FbDrive** in cases in which the drive is running in "Local Mode"
- Channel block **FbDiou**: At the output parameter `Bad`, channel errors are displayed with 1. The channel error is generated from the signal states of `RCasOutST, RCasInLiST, RbkST, SP_LiST`. Error is also generated due to bad mode runtime information (or wrong mode information) and module error.

- Channel block **Pcs7AnOu**: In case of bad input signal and `Feature.Bit31=1` (hold last value), the block now also freezes `PV_ChnST.Value`. `PV_ChnST.ST` then changes to 16#60.
- Channel block **Pcs7DiOu**: In case of bad input signal and `Feature.Bit31=1` (hold last value), the block now also sets `PV_ChnST.ST` = 16#60.

5.8.2 Update information for Version 9.1 SP2 Update 5

The following applies to the software update from Version 9.1 SP 2 Update 4 to Version 9.1 SP 2 Update 5:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Compile changes**

List of changed blocks

Name	Number	Bock Version	Interface change	Code change
Intlk02	FB1824	5.7	No	Yes
Intlk04	FB1825	5.7	No	Yes
Intlk08	FB1826	5.7	No	Yes
Intlk16	FB1827	5.7	No	Yes
MonAnL	FB1845	5.7	No	Yes
MonDiL	FB1848	5.7	No	Yes
MotL	FB1850	5.7	No	Yes
MotRevL	FB1851	5.7	No	Yes
MotS	FB1910	5.7	No	Yes
MotSpdCL	FB1854	5.7	No	Yes
MotSpdL	FB1856	5.7	No	Yes
Vlv2WayL	FB1897	5.7	No	Yes
VlvAnL	FB1896	5.7	No	Yes
VlvL	FB1899	5.7	No	Yes
VlvMotL	FB1900	5.7	No	Yes
VlvPosL	FB1918	5.7	No	Yes
VlvS	FB1911	5.7	No	Yes
PIDConL	FB1874	5.7	No	Yes
PIDConR	FB1875	5.7	No	Yes
PIDStepL	FB1878	5.7	No	Yes

TotalL	FB1906	5.7	No	Yes
FmCont	FB1818	9.7	No	Yes
FmTemp	FB1819	9.7	No	Yes
Pcs7AITC	FB1933	9.7	No	Yes
Pcs7AnOu	FB1870	9.7	No	Yes
Pcs7DiOu	FB1873	9.7	No	Yes

Note

The comment [Release 09.01.02.05] has been inserted in all changed blocks.

5.9 Version 9.1 SP2 Update 6

5.9.1 Changes in Version 9.1 SP2 Update 6

- For channel block **FbDiOu**, Feature Bit 28 has been introduced. If `Feature.Bit28` is true, then Rbk and RCasOut has bad value
- For channel block **FbDrive,** Calculation of control word 2 "`PZDOut2`" has been corrected now if low limit for the scaling of the process value "`SP_LiScale.low`" is not zero.
- For **PIDConL/PIDConR** and **PIDStepL** blocks, the operator permission to change the setpoint is no more granted when Forced Tracking function is active.
- The multi language-specific files have been consolidated into a single file "Multilingual_ReadMe_OSS.html" for improved efficiency.
- For **MotSpdCL** block, now mode change to Out Of Service will not disable Gradient limit (`SP_RateOn`) if it was already activated in previous mode.
- For **VlvAnL** block, when Gradient limiting of the manipulated variable is activated with gradient high or low limit value as zero and if manipulated variable tracking / forced tracking is active then tracking functionality remains intact.

5.9.2 Update information for Version 9.1 SP2 Update 6

The following applies to the software update from Version 9.1 SP 2 Update 5 to Version 9.1 SP 2 Update 6:

1.	Make Textual Interconnections	**Yes**
2.	Compile AS	**Compile changes**
3.	Download AS program	**Download changes**
4.	AS STOP necessary	**No**
5.	Compile OS	**Compile changes**

List of changed blocks

Name	Number	Block Version	Interface change	Code change
MotSpdCL	FB1854	5.7	No	Yes
PIDConL	FB1874	5.7	No	Yes
PIDConR	FB1875	5.7	No	Yes
PIDStepL	FB1878	5.7	No	Yes
FbDiOu	FB1816	9.7	No	Yes
FbDrive	FB1905	9.7	No	Yes
VlvAnL	FB1896	5.7	No	Yes

Note

The comment [Release 09.01.02.06] has been inserted in all changed blocks.

Information on configuration and operation 6

- The **blocks in the library** have been designed exclusively for **S7-400** automation systems and are not suitable for the configuration of failsafe applications.
- For **AS blocks** that include **ALARM_8P blocks**:
  The BOOL data type is allowed as a **message associated value** (not allowed are: Bit array), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER. Copy a STRING data type into an ARRAY OF BYTE . Structures must be interconnected with the message associated values via a converter block; otherwise, these message associated values are always displayed with "0".

  Note

  When an associated value is interconnected for the first time or the value of the data type is changed, this takes effect after a CPU restart.

- The **templates** contain CFC charts with ready-made technological functions. These charts include the blocks and interconnections that are required as default for each of the functions. You can continue to use these charts as a template for individual automation solutions in the future. The templates are described in detail in the online help.
- You are not permitted to set the **minimum cycle time** of OB1 to a value other than 0. For **CPU 410 PA SMART** , the minimum cycle time of OB1 is permanently set to 200 ms.
- Note that the **signal status** of the individual block I/Os is not evaluated by the **SFC type**, since the SFC type does not provide any features of the Struct type.
- **Interconnecting inputs and outputs with APL structures to inputs and outputs of blocks without structures.**
  This CFC function is available from PCS 7 V7.1 SP1 and enables the **interconnection of APL block parameters with blocks of other libraries without converter block**. This simplifies the engineering. However, this function may result in an interruption of the Flow signal status.
  Example:
  A BOOL output with status information can be connected directly to a BOOL input without status information and vice versa. If the signal status information is to be maintained, the signal status must be routed around the "non-APL logic" using converter blocks.
- The user-defined data types (**UDT**) from 1 to 50 are reserved for APL.
- The **FbAnIn**, **FbAnOu**, **FbDiIn** and **FbDiOu** channel blocks support **FF field devices** with analog or digital input and output values.
  The first implementation stage of using FF field devices behind **Profibus Slave AB7000** (introduced for PCS 7 V7.0) is no longer supported.
- The use of different channel block types for the same drive or frequency converter device type is not permitted.
- If the **EventTS/Event16TS** block is interconnected with the `EventTsIn` input in a technological block, the **EventTs/Event16TS** and the technological block must be installed **in the same** cyclic interrupt OB of the **same** CPU.

- The use of different channel block types for the same drive or frequency converter device type is not permitted.
- **Feature "Separate evaluation of excluded and simulated interlock signals"**
  If the feature is enabled using `Feature2.Bit2 = 1`, the signal may **not** be inverted per CFC for an interconnection to the inputs, otherwise the **bypass display** cannot be refreshed for system related reasons.
  This affects the following interlock inputs:
  Intlk02, Intlk04, Intlk08, Intlk16: `In01`, `In02`, ...
  DoseL, MotL, MotRevL, MotSpdL, MotSpdCL, VlvL, Vlv2WayL, VlvAnL, VlvMotL, VlvPosL: `Permit`, `Interlock`, `Protect`
  MotRevL, MotSpdCL: `Permit`, `Interlock`, `Protect`, `PermRev`, `IntlRev`, `ProtRev`
  MotS, VlvS, OpDi01, OpDi03, PIDConL, PIDConR: `Interlock`
  For the **inversion of the interlock signals**, the inputs `InvIn01`, `InvIn02`... can be used at the interlock blocks.
- **LAG** block
  A change to the `SampleTime` first takes effect after a change to the delay time `LagTime` or after resetting the block (`Reset` = 1).
- Before using the **PCS 7 example project Advanced Process Control** (APL_Example_xx, xx refers to the language version), it recommended to update the function blocks of the PCS 7 Advanced Process Library and the basic data contained within it using the OS Project Editor.
- Channel block **Pcs7DiOu**:
  When using the digital outputs of modules with the order numbers **6DL1 133-6EW00-0PH1** and **6DL1 132-6HD50-0PK0** and configuring substitute values in the HW configuration, the configured substitute value must be entered at the `StartVal` input parameter at the associated channel blocks **Pcs7DiOu** in the CFC Editor and `Feature Bit 0` must be set to the value "1" for the substitute value to be applied as the startup value.
- Channel block **Pcs7Cnt1**:
  For the acyclic data traffic to an **FM350-2 counter module** to function correctly, at least one **Pcs7Cnt1** channel block must be configured in the user program for each FM350-2 counter module configured in the HW configuration.
- Channel blocks **FbDrive**, **FbSwtMMs**:
  `Feature Bit 9` (as of CFC version 9.0.1.2) is used to select whether the CFC installs the standard diagnostic blocks of the PCS 7 Basis Library or the diagnostic blocks of an AddOn Library for "Generate module drivers".
  0 = Diagnostic blocks from PCS 7 Basis Library, 1 = Diagnostic blocks from an AddOn Library.
- **MonDiL, MonDiS** blocks:
  A change of the `Color` parameter does not take effect until after a picture change.
- **Channel blocks** - **Input** `Feature`**:**
  To prevent duplicate configurations, specific Feature bits are automatically configured by the **PCS 7 driver generator** during CFC compiling with the "**Generate module drivers**" function based on the settings in the hardware configuration. For this reason, the `Feature` parameter **must not** be interconnected. Otherwise, any existing interconnections could be deleted.

- When configuring `Feature2 Bit3=1` **"Control priority for an invalid and implausible input command"** on the **MotRevL** and **MotSpdCL** motor blocks, note that in automatic and local mode (`LocalSetting` =1/3) a direct change of direction is also possible with `Feature Bit7 =0`. If a command is given in the opposite direction, the motor first goes to stop and thereby returns to a plausible state. Then, the motor changes to the opposite direction. The configured enable time `IdleTime` is maintained in any case.
- For the controllers **PIDConL**, **PIDConR** and the motor block **MotSpdCL**, the function **"Delay alarm for system deviation with setpoint steps"** (`ER_AL_DFac, ER_AH_DFac >0`) only works for the limit in the direction of the setpoint step. If the control difference exceeds the opposite limit when the process value overshoots or undershoots, the delay time at the `ER_A_DCOut` output is reset and the corresponding `ER_A_DC` or `ER_AH_DC.` on-delay goes into effect.
- The `Gain` parameter must not be set at `>100` at the **PIDConR** block. Using the scaling factors `NormPV, NormMV`, it is possible to set the effective controller gain `GainEff` so that a controller gain at `Gain >100` is not necessary.
- The **PST** and **PST_Calc** blocks and for the **partial stroke test** uses the following IEC functions AD_DT_TM (FC1), CONCAT (FC2), DT_DATE (FC6) DT_TOD (FC8), I_STRING (FC16, NE_DT (FC28) and RIGHT (FC32) from the STEP 7 Standard Library. Therefore, before using the two blocks **PST** and **PST_Calc** , the **reserved area for other applications** must be adapted in the **CFC settings**. Details can be found in the manual in section "**Compile and load logic of the partial stroke test**".
- If the **time stamp of the I/O** is to be used with `TimeStampOn = 1` set for an **EventTS instance**, it must be ensured that `Feature.Bit5 = 0` for **all** connected **Pcs7DiIT** channel blocks. Time stamping will otherwise not work reliably for this EventTS instance.
- During placement at one of the end positions, no P_Stop stop pulse is output at the **VlvPosL** block.
- If the **position control is switched on** at the **VlvPosL** block, the block tries to move to the specified position. As a result, the control signals Open, Close, P_Open, P_Stop may change even during the **resynchronization**.
- The **manipulated variable difference** (`MV-Rbk`) can now also be monitored at the **VlvAnL** block in the end position ranges by configuring `Feature2 Bit15 =1`. To avoid monitoring errors, adjust the manipulated variable limits `MV_HiLim, MV_LoLim` so that the position feedback `Rbk` can also reliably reach the end positions and that no impermissibly large (`MV-Rbk`) differences can occur in the end positions.
- When the auxiliary valve is closed, the **position feedback** `Rbk` is not monitored for the limit values `RbkWH, RbkWL` at the **VlvAnL** block. The `RbkWH_Act, RbkWH_Act` outputs are reset.
- If the operating authorization of the stop command `OS_Perm Bit4 =0` is reset, a motor that has been started from the end position can only be stopped by the opposite command in the standard view at the **VlvMotL** block if the corresponding feedback signal has been reset `FbkOpen =0, FbkClose =0`.

Note

During configuration, you need to ensure that a started motor cannot be stopped via the manual commands in the standard view if the `FbkOpen =1, FbkClose =1` feedback signals are defective in the event of `OS_Perm Bit4 =0`.

- For the interlock blocks **Intlk02, Intlk04, Intlk08, Intlk16**, the following must be taken into account when assigning parameters for the bypass enable `BypEn`:

  Note

  If an input `Inx` of the interlock block is connected to the output `Out` of an upstream interlock block, the bypass enable bit of the corresponding input must not be set `BypEn.Inx =0`, because this could exclude interlock-relevant inputs of the upstream interlock block.
  The bypass enable bit `BypEn.Inx =1` should only be set for interlock inputs that are directly connected to a measuring point, e.g. MonAnL.
  The parameter `BypEn.Inx` is only effective in conjunction with `Feature.Bit2 =1`.

- **Inputs that are written back by the function block algorithm** are marked by an * in the manual. Note the following.

  Note

  Input parameters to which the function block algorithm writes back or which are intended for operation via the faceplate should never be interconnected, as otherwise the correct block behavior can no longer be ensured.

Notes on the documentation

7

- The terms or message classes Alarm, Warning, Tolerance or the corresponding codes in icons and colors used in the documentation are not valid when using **user-configured message classes**.
  These terms and colors depend on the project-specific setting.
  The **user-configured message classes** are only supported by block icons from the template pictures **@PCS7TypicalsAPLV9.PDL** or template **@TemplateAPLV9.PDL**
- The following has to be replaced in the section "**Controlling a device infeed or inverter enable**" of the "**MotSpdCL** functions".
  - OLD:
    When the function is activated, the motor can also be started directly. When starting forwards or backwards the device infeed or inverter enable is also activated. The frequency converter automatically determines the correct sequence. The motor can now be stopped and restarted without having to switch off the device infeed or inverter enable. The activation of the device infeed or inverter enable for starting the motor can be performed in the automatic, manual and local operating modes with local setting 1 or 3, or by forcing.
  - NEW:
    When the "device infeed" function is activated, the motor can also be started directly: The device infeed is also activated when starting in forward or reverse. The frequency converter automatically determines the correct sequence. The activation of the device infeed for starting the motor can be performed in the automatic, manual and local operating modes with local setting 1 or 3, or by forcing.
- The following must be replaced in the "**Redundancy**" section of the "**Description of Pcs7AnIn**" chapter.
  - OLD:
    If both signals deliver a "bad" quality code, the substitute value is passed on to the process with a "bad" quality code
  - NEW:
    If both signals deliver a "bad" quality code, output `Bad` *= 1 is set and the last valid value, a substitute value or the invalid value takes effect depending on the configuration of the* `Feature` *parameter.*
- The following information is missing in the **Area of application for Pcs7AnIn** section of "**Description of Pcs7AnIn**" chapter:
  Note:
  You can use the block to record temperatures in °C and °F. Kelvin is not supported.
- The following paragraph must be amended in the "**Configuration**" section of the "**Description of Pcs7AnIn**" chapter.
  If you use the block for **temperature measurement** and the `Mode` input has been connected by the driver generator, the unit at the `PV_InUnit` input is automatically set to the code 1001 (°C) or 1002 (°F) by the block depending on the hardware configuration.
- In the **I/Os of Pcs7AnIn** chapter, there is no note for the `PV_InUnit` input parameter that this input can be written back by the block algorithm.

- The following must be replaced in the "**Redundancy**" section of the "**Description of Pcs7AnIn"** chapter.
  - OLD:
    If both signals deliver a "bad" quality code and an "E/E redundancy warning" (`Mode` *= 16#x5xxxxx) is issued, 50 % of the value of the application program is passed on to the process.*
    If both signals deliver a "bad" quality code, the last process value is retained.
  - NEW:
    If both signals deliver "good" quality code and an "E/E redundancy warning" `(Mode` *= 16#x5xxxxx) is issued, 50 % of the* `PV_In` *value takes effect at each of the outputs* `PV_Out` *and* `PV_OutSlv`*.*
    If both signals deliver "bad" quality code, 100 % of the `PV_In` *value takes effect at each of the outputs* `PV_Out` *and* `PV_OutSlv`*.*
- "**Description of TimeTrig**"
  - The `DailyOn`, `WeeklyOn`, `MonthlyOn` and `MinuteOn` inputs mentioned in the text are not available on the block for the selection of the **periodic trigger**. The selection of the periodic trigger is made with the `PerMode` input.
    `PerMode=0` No periodic trigger
    `PerMode=1` Daily periodic trigger
    `PerMode=2` Weekly periodic trigger
    `PerMode=3` Monthly periodic trigger
    `PerMode=4` Minute periodic trigger, the trigger is always released at the same minute of an hour.
- Chapter "Status word of **FbSwtMMS"**
  The status word assignment is as follows:

```
Bit  0 FdkRev
Bit  1 FdkOff
Bit  2 FdkFwd
Bit  3 OverlAct
Bit  4 LockTmAct
Bit  5 FdkAuto
Bit  6 ErrAct
Bit  7 WarnAct
Bit  8 Status8
Bit  9 Status9
Bit 10 ManSpcO5
Bit 11 ManSpcO6
Bit 12 ManSpcO1
Bit 13 ManSpcO2
Bit 14 ManSpcO3
Bit 15 ManSpcO4
```

- In the chapter "**Configurable response using the Feature I/O**" - "**Calculation of the time stamp**", the **note** relating to `Feature bit5` must be replaced.
  - OLD:
    Changes are made in the **Pcs7DiIT** *block and prevent invalid error messages regarding time stamp.*
  - NEW:
    With `Feature.BitB5 = 0` *the* `TimeStampOn` *input at the connected* **EventTS** *instance must be configured with "1" (use I/O time stamp).*

- The following sections must be added or replaced in the "**Time stamp**" section:
  "**Time stamp**" section:
  The two blocks **EventTS** and **Event16TS** can be used for processing of time stamped signals. If the **EventTS** block is mentioned in this chapter, the **Event16TS** block is always meant accordingly.
  "**Forming the time information**" section: The section must be replaced as follows:

  – OLD:
    The time information is generated by one of the following methods and is specified at the block by means of the input parameter TimeStampOn:
    `TimeStampOn = 0`: Use time stamp of the CPU (default)
    `TimeStampOn = 1`: Use time stamp of the I/O.

  – NEW:
    The time information is generated in one of the following two ways and is specified at the EventTS block via the `TimeStampOn` input parameter for **all** channels of an **EventTS** instance:
    `TimeStampOn = 0`: Use time stamp of the CPU (default)
    `TimeStampOn = 1`: Use time stamp of the I/O.

  Time stamping in the EventTS block - Use time stamp of the CPU
  Connect the binary output parameter of another block (e.g. **Pcs7DiIn**) to an `Inx` message input (x = 1 ... 8) of the **EventTS**, (**Event16TS**, x = 1 ... 16) block.
  The message time stamp is formed in the EventTS from the CPU time. If this mode is to be selected for a signal, you need to disable time stamping for the corresponding channel in the HW configuration. If you use the **Pcs7DiIT** block instead of the **Pcs7DiIn** channel block for signal acquisition in your user program and if you have connected this block to one of the `Inx` inputs with the EventTS, you also need to ensure that `Feature.Bit5` at **Pcs7DiIT** is configured with "1" (CPU time stamp).

  If the **EventTS** block detects a change of the signal state at a signal input, it reads the CPU time and uses this as time stamp. Only the signal changes that are slower than the cycle time of the block can be detected.
  High-precision time stamp in the process I/O - Use time stamps of the I/O
  You have configured the hardware of your system for high-precision time stamping as explained in the "**PCS 7 - High-Precision Time Stamping Function Manual**". Signal changes are recognized in the I/O and provided with the time stamp of the I/O. This data is available at the output parameter `TS_Out` of the **Pcs7DiIT** block. To generate a time-stamped message, connect the `TS_Out` output parameter of **Pcs7DiIT** to an `InTSx` (x = 1 ... 8) message input of the **EventTS**, (**Event16TS**, x = 1 ... 16) block. Ensure that `Feature.Bit5=0` (I/O time stamp) is at the **Pcs7DiIT** block.

  The high-precision time stamp is independent of the cycle time of the blocks. The actual time resolution for two different status changes depends on your plant configuration and the hardware you are using.

- In the chapter "**Functions of VlvPosL**", the following note must be inserted in the section "**Motor valve control without position control**":
  After a stop control, the manipulated variable is tracked for the position feedback (`MV_Out = RbkOut`) for the time from `V_MonTiStatic`. The manipulated variable is then held and can be monitored to ensure that the position is maintained.

- In the section "**Motor with PROFIdrive Drive Profile telegram 1 and 20 (Namur)**", the following sentence is missing after the first section:
  The **MotSpdCL** block is prepared for **local mode** with `LocalMode=4`. For other `LocalMode` settings, interconnections must be changed in the template `Drive`.
- In the "**Interlocks**" section, the "**Interlocks for blocks**" paragraph must be replaced by the description of the **switch-on enable** as follows:
  - OLD:
    Switch-on enable ("Permission"): The switch-on enable allows the motor to be switched in the enabled direction. The switch-on inhibit is indicated by a padlock in the default view. The motor cannot switch in the enabled direction.
  - NEW:
    Switch-on enable ("Permission"): The switch-on enable (Permit=1) makes it possible to leave the neutral position of the block.
    The switch-on inhibit (`Permit=0`) is always indicated by a padlock in the default view. The display is independent of whether the switch-on inhibit is effective or not. This gives the operator the possibility to determine whether an actuator could be switched on again after it has been switched off or whether this would be prevented by an existing switch-on inhibit.
- In the "**Bypass signals**" section at the end of the paragraph "If the bypass function...is set." and in the section Functions of Intlkxx under "Bypass", the following note must be inserted:

Note

If an input `Inx` of the interlock block is connected to the output `Out` of an upstream interlock block, the bypass enable bit of the corresponding input must not be set `BypEn.Inx =0`, because this could exclude interlock-relevant inputs of the upstream interlock block.
The bypass enable bit `BypEn.Inx =1` should only be set for interlock inputs that are directly connected to a measuring point, e.g. MonAnL.
The parameter `BypEn.Inx` is only effective in conjunction with `Feature.Bit2 =1`.

- In the section "Functions of **PIDConS**", an open bracket must be removed in the formula under "PID algorithm".
  - OLD:
    `MV = GainEff · (1 + (1 / (TI · s) + (TD · s) / (1 + TD / DiffGain · s)) · ER`
  - NEW:
    `MV = GainEff · (1 + 1/(TI·s) + (TD·s)/(1+TD/DiffGain·s)) · ER`
- In the chapter "**Descriptions of PIDConL**", the description for `Bit25` and `Bit28`.
  `Status1 Bit25` "`MV_TrkOn.Value`"
  `Status1 Bit28` "Not used" is missing in the **status word** `Status1`.
- In the chapter "**Descriptions of PIDConR**", the description for `Bit25` and `Bit31`.
  `Status1 Bit25` "`MV_TrkOn.Value`"
  `Status1 Bit31` "Bypass information to be displayed in the block icon" is missing in the **status word** `Status1`.

- The note contained in the "**Description of STRep**" section must be replaced as follows:

  Note

  The `QSTATE` output of the STRep block has the attribute `S7_enum: = 'Operating State'`. This assignment can be changed for user applications.

  Note that
  the "Smart Alarm Hiding" function expects the "Operating State" enumeration.

- In chapters "**Output substitute value if raw value is invalid**" and "**Outputting last valid value if raw value is invalid**" related to `Feature.Bit29` and `Feature.Bit30`.
  - Old
    Bit = 1.....
    If there is an invalid raw value or the device is in the initialization phase, the output parameter Bad = 1 is set automatically
    Prioritizing the Feature bits for channel blocks:
  - New
    Bit = 1.....
    Prioritizing the Feature bits for channel blocks:
- In the Chapter "**Intlkxx functions**", Section "**Bypass**" of all Interlock blocks (**Intlk02, Intlk04, Intlk08** & **Intlk16**)
  - OLD
    Special situation: If all input parameters are bypassed, the output value is defined using the DefaultOut parameter.
  - NEW
    Special situation: If all input parameters are bypassed, the output value is defined using the DefaultOut parameter. Signal status of Out will be Simulation (16#60) irrespective of FeatureBit2

Change history of PCS 7 Advanced Process Library

8

Changes since delivery release PCS 7 Advanced Process Library Readme (online)

Version	Edition	Change
2021-01-21 (Online)	01/2021	Delivery status PCS 7 Advanced Process Library V9.1
2021-11-11 (Online)	11/2021	Delivery status PCS 7 Advanced Process Library V9.1 SP1
2022-06-21 (Online)	06/2022	Delivery status PCS 7 Advanced Process Library V9.1 SP2
2022-11-21 (ONLINE)	11/2022	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 1
2023-04-28 (ONLINE)	04/2023	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 2
2023-08-25 (ONLINE)	08/2023	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 3
2024-02-29 (ONLINE)	02/2024	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 4
2024-12-02 (ONLINE)	12/2024	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 5
2025-10-30 (ONLINE)	10/2025	Delivery status PCS 7 Advanced Process Library V9.1 SP2 Update 6