Release Notes

PPLib800xA

Version 5.4

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

TRADEMARKS

Copyright © 2016 ABB

All rights reserved.

Release: December 2016

Document number: 3BTG811792-3061

Table of Contents

Section 1 - Release Notes

Introduction

This document represents the release notes for Pulp & Paper Library, PPLib800xA 5.4.

This document lists the changes that have been incorporated into this release. It also describes the upgrade procedure for upgrading an existing library to this version.

Version Designation

PPLib800xA 5.4 consist of the following

Package	Components	Version
Core	PP Library Alarm & Event List Configurations	-
	PP Library NLS	-
	PP Library Trend Templates	-
	Pulp & Paper Library Colors	-
	PP_ElementLib	5.4-0
	PP_FunctionLib	5.4-0
	PP_UMCLib	5.4-0
Paper Expansion	PP_PaperExpLib	5.4-0
Power Expansion	PP_PowerExpLib	5.4-0
Smart Device	PP_SmartDeviceCoreLib	1.4-0
	PP_SmartDeviceTemplateLib	1.4-0
	PP_UMC100Lib	1.4-0

Revision History

Revision	Release Date	Remark
PPLib800xA 5.4	16 Dec 2016	
PPLib800xA 5.3 rollup 2	4 Dec 2015	No difference with rollup 1. Only correction on library licensing.
PPLib800xA 5.3 rollup 1	17 Sep 2015	
PPLib800xA 5.3	4 Apr 2015	
PPLib800xA 5.2 rollup 2	24 Jan 2014	
PPLib800xA 5.2 rollup 1	27 May 2013	
PPLib800xA 5.2 hotfix 1	30 Dec 2012	
Pulp & Paper Library 5.2-0	20 Feb 2012	
Pulp & Paper Library 5.1-0	8 Nov 2010	

Compatibility

This release is compatible with previous versions.

Restriction

PPLib800xA 5.4 requires 800xA System Version 5.1 with at least FP4 with revD installed. The following system extension must be loaded:

- ABB SFC Viewer
- ABB Central Licensing System Extension
- AC800M Connect

Related Documentation

The following documents are included in the distribution media of PPLib800xA 5.4.

Document Kind	Title	Document Number	Rev
Release Notes	PPLib800xA 5.4 Release Notes	3BTG811792-3061	A
Product Guide	PPLib800xA 5.4 Product Guide	3BTG811792-3056	C
Functional Description	PPLib800xA 5.4 Resource References	3BTG811792-3003	F
	Functional Description - PP_ElementLib	3BTG811792-3010	F
	Functional Description - PP_FunctionLib	3BTG811792-3011	F
	Functional Description - PP_UMCLib	3BTG811792-3012	F
	Functional Description - PP_SmartDeviceCoreLib	3BTG811792-3013	F
	Functional Description - PP_SmartDeviceTemplateLib	3BTG811792-3014	F
	Functional Description - PP_PaperExpLib	3BTG811792-3057	C
	Functional Description - PP_PowerExpLib	3BTG811792-3058	C
	Functional Description - AIC01	3BTG811792-3015	F
	Functional Description - AIS01	3BTG811792-3016	F
	Functional Description - AOC01	3BTG811792-3017	F
	Functional Description - AOS01	3BTG811792-3018	F
	Functional Description - DIC01	3BTG811792-3019	F
	Functional Description - DIS01	3BTG811792-3020	F
	Functional Description - DOC01	3BTG811792-3021	F
	Functional Description - DOS01	3BTG811792-3022	F
	Functional Description - DriconS	3BTG811792-3023	F
	Functional Description - DriconS02	3BTG811792-3059	B
	Functional Description - Flow01	3BTG811792-3024	F
	Functional Description - GenCon	3BTG811792-3060	A
	Functional Description - GroupStart	3BTG811792-3025	F
	Functional Description - Man01	3BTG811792-3026	F
	Functional Description - Mot01	3BTG811792-3027	F
	Functional Description - Mot02	3BTG811792-3028	F

Document Kind	Title	Document Number	Rev
	Functional Description - MotFreq	3BTG811792-3029	F
	Functional Description - Motval01	3BTG811792-3030	F
	Functional Description - Motval02	3BTG811792-3031	F
	Functional Description - PID01	3BTG811792-3032	F
	Functional Description - PID01A	3BTG811792-3033	F
	Functional Description - Ratio01	3BTG811792-3034	F
	Functional Description - Reporting	3BTG811792-3061	A
	Functional Description - Seq01	3BTG811792-3035	F
	Functional Description - Total01	3BTG811792-3036	F
	Functional Description - Totalizer01	3BTG811792-3037	F
	Functional Description - Valve01	3BTG811792-3038	F
	Functional Description - UMC22	3BTG811792-3039	F
	Functional Description - UMC22_Act	3BTG811792-3040	F
	Functional Description - UMC22_Act02	3BTG811792-3041	F
	Functional Description - UMC100	3BTG811792-3047	F
	Functional Description - UMC100ACT01	3BTG811792-3048	F
	Functional Description - UMC100ACT02	3BTG811792-3049	F

Section 2 - New Features/Improvements

This section describes the new features or improvements in this release and previous releases.

PPLib800xA 5.4

Internal Code Optimization

Internal code inside the function blocks in the library are improved and optimized. It requires lesser memory and also reduce the CPU load. Upgrading from PPLib800xA 5.2 will see a reduction in CPU between 20-30% depending on the number of objects used and controller's cycle time. Upgrading from PPLib800xA 5.3 rollup 2 will see smaller reduction in CPU.

Number of instance (NOI) is also reduced, which means more objects can be added into the same application.

Interlock Navigation

Interlock Display is used for configuration of the interlocking information. It is used to configure the interlock text, operator blockable, interlock override, start interlock and block event of each interlock.

It is now possible to call up the interlocking object's faceplate from the faceplate or Interlock Display. No additional configuration is required.

When the configured interlock text contains object name in the first word (separated by space), its faceplate can be called up. Example:

- PM12MP001.1 not running → will open PM12MP001.1 faceplate
- PM1LI02 low → will open PM1LI02 faceplate

By default, it will call *Reduced Faceplate*. But it's also possible to change the default to *Faceplate* or *Extended Faceplate* by changing the configuration value in **PPLib800xA Customization NLS**.

From faceplate, the active interlock text will be shown as usual. If the interlock text contain object that can be found in the Control Structure, the text will be indicated as button. **Left click** will open the interlocking object's faceplate, and **right click** will show the context menu of the interlocking object. If the text doesn't contain any object, no button style will be indicated.

Similarly, the same feature is also implemented in the Interlock Display. As **left click** is now used to call up the interlocking object's faceplate, to enter the text will now require **Shift + left click**. Interlock text color is also now changed from green to black color to improve visibility.

Diagnostics Display

Diagnostics Display is used for presenting the diagnostics information from data set mapping for communication of smart devices object types.

Diagnostics Display of ACS550, ACS600, ACS800, ACS850 and ACS880, UMC22 and UMC100 are improved with NLS support. Bit number of each bit can be set as visible or hidden.

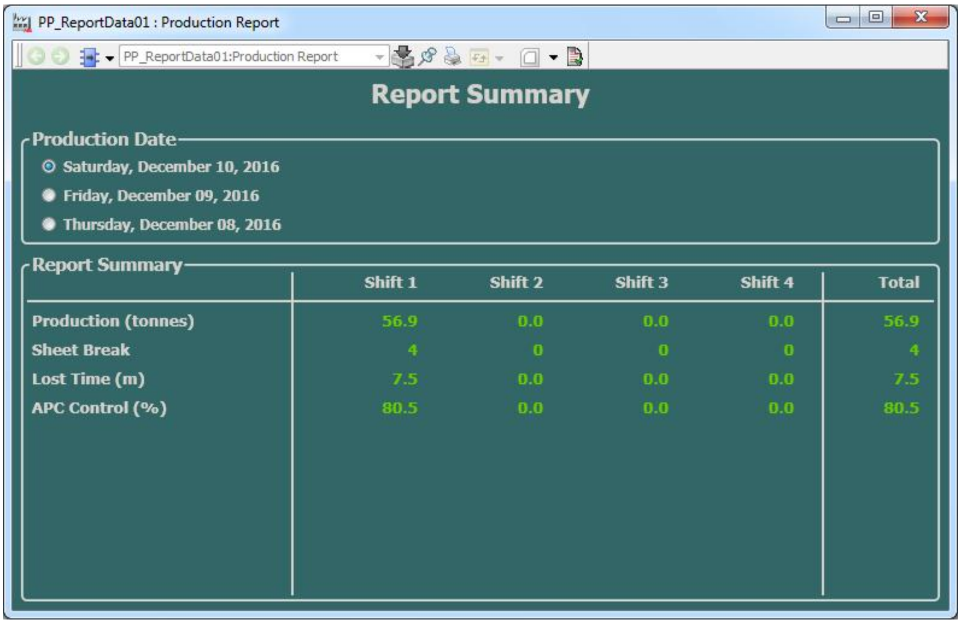
For customized smart devices, it’s also possible to implement the same similar style if required. A set of generic elements in the toolbox is available for use.

Reporting

Reporting function consist of ReportConfig and ReportData01 function blocks.

ReportConfig is non aspect object function block, used to generate shift treatment. Shift configuration is done at the project constant. Max number of shift supported is 4. It has **ReportsPar** parameter which need to be connected to ReportData01. An application requires only one ReportConfig. It can be connected to multiple ReportData01.

ReportData01 is an aspect object function block, used to present the reporting summary. The function block supports up to 10 inputs. Four reporting types are supported (totalizer, counter, timer (in minutes) and ratio/utilization) for each input. Each data will be stored for the last 3 days (today, yesterday and previous day). Input name can be customized accordingly. Operator can see the report summary of each day by selection the corresponding date.


PP_ReportData01 : Production Report
PP_ReportData01:Production Report

Report Summary

Production Date

◉ Saturday, December 10, 2016
○ Friday, December 09, 2016
○ Thursday, December 08, 2016

Report Summary

	Shift 1	Shift 2	Shift 3	Shift 4	Total
Production (tonnes)	56.9	0.0	0.0	0.0	56.9
Sheet Break	4	0	0	0	4
Lost Time (m)	7.5	0.0	0.0	0.0	7.5
APC Control (%)	80.5	0.0	0.0	0.0	80.5

GenCon

GenCon can be used to build and present user defined control function. It consist of GenConO and and GenCon01.

GenConO is used to read operator action through GenConPar and decode it as output parameter. These parameters can be used in application to define the custom control function.

GenCon01 will read the values from the custom control function and present these information on faceplate. Internally, GenCon01 is also built with MV, Dev and ActDev alarm limit, I/O status monitoring and some alarm and event handling.

Bitwise Operation

Some new function block for bitwise operation are added in PP_ElementLib.

- B16ToDInt	: Convert Boolean16 to dint
- B16ToDword	: Convert Boolean16 to dword
- B16SimpleEvent	: Generate event when each value in Boolean16 is true
- B32ToDInt	: Convert Boolean32 to dint
- B32ToDword	: Convert Boolean32 to dword
- B32SimpleEvent	: Generate event when each value in Boolean32 is true
- DIntToB16	: Convert dint to Boolean16
- DIntToB32	: Convert dint to Boolean32
- DIntSimpleEvent	: Generate event when each decoded bit value in dint is true
- DwordToB16	: Convert dint to Boolean16
- DwordToB32	: Convert dint to Boolean32
- DwordSimpleEvent	: Generate event when each decoded bit value in dint is true
- GetBitDInt	: Get value of certain bit number from dint input
- GetBitDword	: Get value of certain bit number from dword input
- SetBitDInt	: Set value of certain bit number from dint input, then update input
- SetBitDword	: Set value of certain bit number from dword input, then update input

GetBit and SetBit are extensible parameters.

Object KPI

Object KPI are added for some motors and control object types:

- **Normal Mode Ratio**
  The percentage value show the proportion of the object operated under normal mode. It is important to define which mode is the normal mode for each loop. More than 1 mode can be configured as normal mode. Higher ratio means the object is operated in mode as it should be.

- **Alarm Ratio**
  The percentage value show the proportion of the object in alarm state, whether acknowledged or unacknowledged. Higher ratio means the object is operated in alarm state most of the time.

- **Root Means Square Error**
  This KPI is implemented on PID only. It indicates the performance of PID loop. Lower value indicates better performance. Value is accumulated for an hour before stored as previous data and start to count again.

By default, KPIs are not enabled. User can enable it on selective object as required.

KPIs value can be reset when necessary.

Alarm and Event

The following improvements are related to alarm and event:

- Alarm and event handling is improved for better execution.
- Multiple event which come at the same time will be recorded separately.
- AE Translator aspect is consolidated into a common place in Library Structure.
- Alarm message for MV, Dev, ActDev and Current will included the actual limit value.
- Alarm message for MCErr, PosErr, etc are improved with better information.
- I/O Error in DIS01 and DOS01 will generate alarm instead of indication only.

- New **Low Current alarm** for all motors.
- New **Stop Interlock alarm** for all motors and valve.
  By default, this is not enabled. If enabled, alarm will be generated when motor is stopped by interlock.
- Interlock text can be added into alarm event message by using **PPLib800xA AE Uploader** tool.

Sootblower Library

SBSeq01 is expanded to support up to 75 pairs of sootblowers.

SBMotor01 is updated with some improvement:

- Local control mode
- Input parameter **Bypass** to omit sootblower from sequence control.
- Sootblower mode can be individually controlled from input parameters.

PPLib800xA Customization

A new NLS Resource Manager is added in the Library structure to store the customization required by project:

- **NLSID_FaceplateViewInterlockLink**: to configure the default faceplate of interlocking object.
- **NLSID_HideModeInNormalMode**; to hide the mode in graphic element when object is in normal mode.
- **NLSID_ShiftxName**; to define name of the shift
- **NLSID_ShowDiagnosticsBitNumber**; to show or hide bit number in Diagnostics Display

PPSupport

PPSupport is now added with scheduling function to read the parameters based on certain project setting. It will be automatically run in the background on the selected interval time. Output files will be stored in the defined folder.

Others

- **TrackIB**, implemented in Man01, is now expanded to PID01, PID01A and Ratio01. **TrackIB** can be used to set the output in Manual mode to remain or return to the previous value when interlock is released.
- Indication of actual time of Valve01 for opening or closing is added in Object Display.
- Online help file is included in the library installation.

PPLib800xA AE Uploader

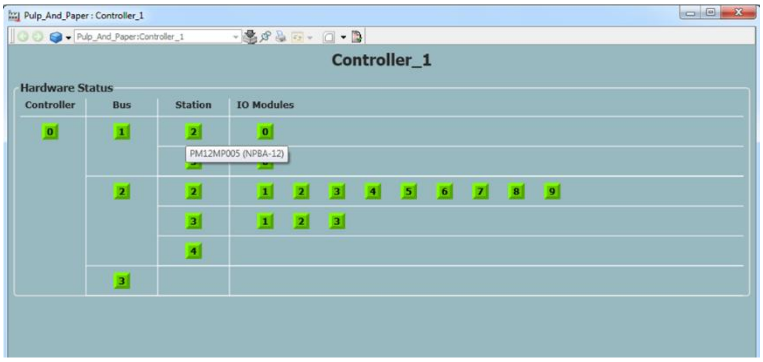
For interlock related alarm and event, it is possible for the Message Description to display the actual interlock text which configured in Text Configuration aspect.

A unique AE Translator for PPLib aspect can be automatically populated in each instance and contain the information of interlock text by using **PPLib800xA AE Uploader** tool.

HwStatus Display Generator

HwStatus Display Generator is a simple tool to generate graphic displays which can be used to monitor the status of all hardware which connected to AC800M controllers. The tool will automatically scan AC800M controllers of all control projects available in Control Structure.

The graphics will be place automatically under [Functional Structure]/Hardware Status Displays. Each controller is presented as one graphic display. Controllers are grouped together per control project.



Controller_1

Hardware Status

Controller	Bus	Station	IO Modules
0	1	2	0
		[illegible]	[illegible]
	2	2	1 2 3 4 5 6 7 8 9
		3	1 2 3
		4	
	3		

PM12MP005 (NPBA-12)

Number inside the indication shows the hardware address. Mouse rollover on the indication will shows tooltip. It presents the hardware name (if configured in Control Builder) and hardware type in the bracket.

PPLib800xA 5.3 Rollup 2

No new improvement in this release.

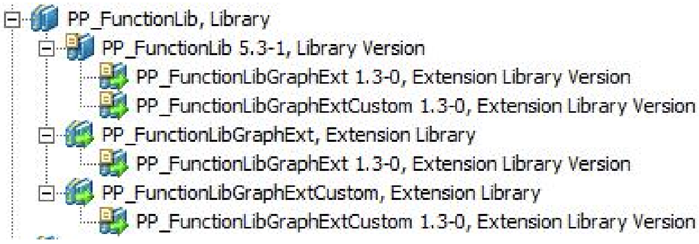
PPLib800xA 5.3 Rollup 1

Language Packs

PPLib800xA 5.3 rollup 1 now comes with Finnish language pack. This is on top of the existing supported language: English, Chinese, Swedish and German.

Library Structure

New graphic extension library is added. The example below is for PP_FunctionLib. Previously, all the graphical presentations are stored in PP_FunctionLibGraphExt.



- PP_FunctionLib, Library
  - PP_FunctionLib 5.3-1, Library Version
    - PP_FunctionLibGraphExt 1.3-0, Extension Library Version
    - PP_FunctionLibGraphExtCustom 1.3-0, Extension Library Version
  - PP_FunctionLibGraphExt, Extension Library
    - PP_FunctionLibGraphExt 1.3-0, Extension Library Version
  - PP_FunctionLibGraphExtCustom, Extension Library
    - PP_FunctionLibGraphExtCustom 1.3-0, Extension Library Version

Now, PP_FunctionLibGraphExt will contain only AC800 Alarm and Lock Control, AE Translator, Graphic Element, Graphic Display and Faceplate Element.

New extension library PP_FunctionLibGraphExtCustom will contain aspect which can be modified by user, such as Faceplate Documentation, Object Trend Display, Trend Signal Properties and Faceplate.

The same structure is applied to PP_PaperExpLib, PP_PowerExpLib, PP_UMC100Lib and PP_UMCLib.

Dricon_S02

New function block for variable speed drive connecting to ABB ACS drives is added. Dricon_S02 has the same profibus mapping as Dricon_S. The main difference is that Dricon_S02 is to be used only for 2 direction speed. In this case, separate start forward and reverse button is available in the faceplate with positive speed setpoint, although it will remain sending negative speed to ACS drive. Dricon_S02 also has separate interlocks for different direction.

AIC01 Signal Error

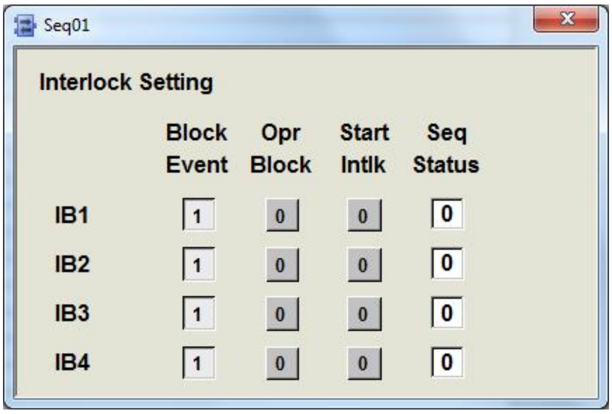
A new parameter **Err** is added on AIC01 with configurable alarm and event from interaction window. If **Err** is *True* and **AEConfigAIErr** is set to *1*, then an alarm will be generated. Numeric indication in faceplate, graphic element and object display will show **???**.

AIS01 Balance Control

A new parameter **Bal** and **BalRef** is added on AIS01. Balance control can be used to override the IOSignal value based on certain process condition. When **Bal** is activated, **BalRef** value will replace **IOSignal.Value**.

Seq01 Interlock

Interlock configuration for Seq01 is now replaced with IBInParType4 which can configured to reset or hold the sequence if interlock is active.



Seq01

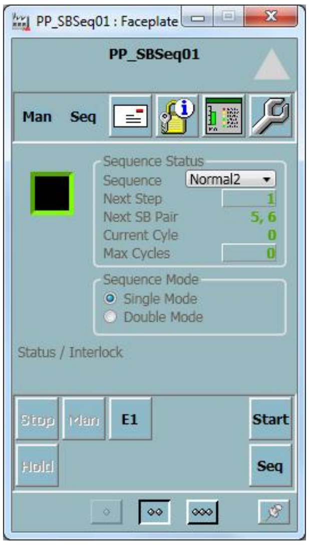
Interlock Setting

	Block Event	Opr Block	Start Intlk	Seq Status
IB1	1	0	0	0
IB2	1	0	0	0
IB3	1	0	0	0
IB4	1	0	0	0

If SeqStatus is set to 1, interlock will hold the sequence. Sequence will be resumed once the interlock is cleared.

By default, SeqStatus is set to 0 which will reset the sequence.

Sootblower Sequence Recipe



PP_SBSeq01 : Faceplate

PP_SBSeq01

Sequence Status
Sequence Normal2
Next Step 1
Next SB Pair 5, 6
Current Cyle 0
Max Cycles 0

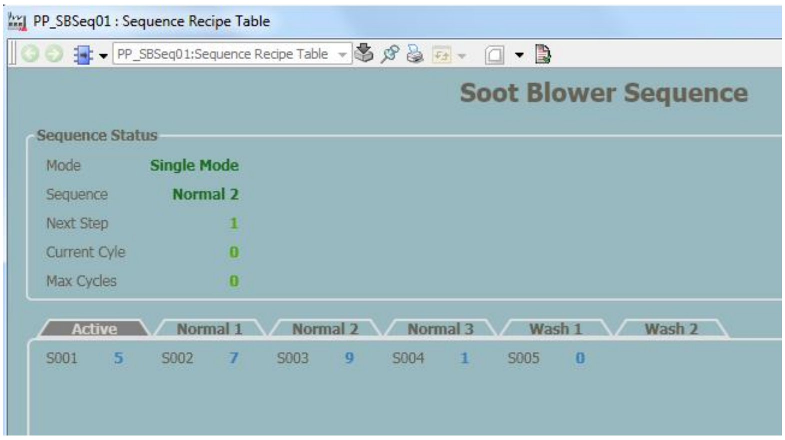
Sequence Mode
● Single Mode
○ Double Mode

Status / Interlock

SBSeq is now added with more recipes storage. It can now stores up to 5 recipes (3 sequences for normal operation and 2 sequences for wash sequence).

Selection of the recipes is done from faceplate only and when sequence is not running.

The sequences can be configured for each recipes from the sequence recipes table. It's possible to have different number of steps between sequences.



PP_SBSeq01 : Sequence Recipe Table

PP_SBSeq01:Sequence Recipe Table

Soot Blower Sequence

Sequence Status

Mode	Single Mode
Sequence	Normal 2
Next Step	1
Current Cyle	0
Max Cycles	0

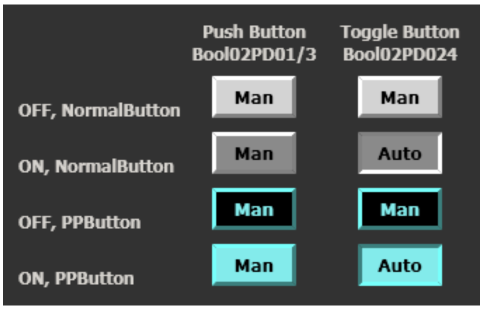
Active | Normal 1 | Normal 2 | Normal 3 | Wash 1 | Wash 2

S001	5	S002	7	S003	9	S004	1	S005	0

Dricon_S added with M2-M5

Dricon_S is previously only has M1. It's now added with M2-M5. M1-M5 fault handling in this function block has the same fault handling priority order as the other motors.

Bool02 button indication



	Push Button Bool02PD01/3	Toggle Button Bool02PD024
OFF, NormalButton	Man	Man
ON, NormalButton	Man	Auto
OFF, PPButton	Man	Man
ON, PPButton	Man	Auto

Bool02 has input parameter **Color** which can be used to manipulate color indication for the text in graphic element Bool02Text. But this is not used for the button.

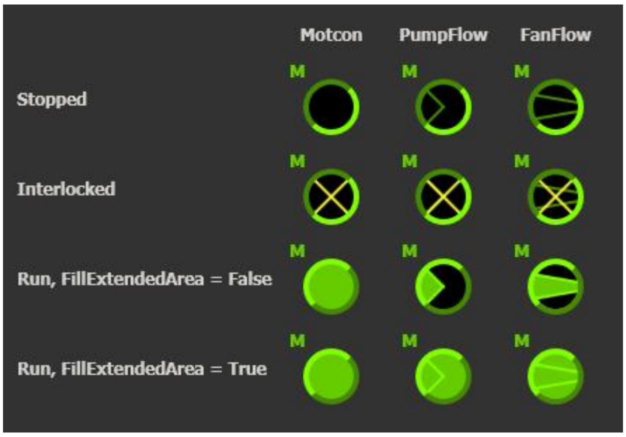
For button indication, a new parameter ButtonStyle is added in the graphic element Bool02PD01 - Bool02PD04. User can select the default *NormalButton* or *PPButton*.

Motval status indication

Status indication of motval opening or closing is now extended to the arrow indication. Arrow indication will blink when motval is opening (up arrow) or closing (down arrow).

Status of limit switches and torque switches is added in Object Display for on-off valve and motval. It's already added before in Signal faceplate element.

Motor graphic elements



Graphic elements MotPD00 and MotPD01 is added with new parameter **MotorType** and **FillExtendedArea**. **MotorType** can be used to select the indication of *Motcon* (default), *PumpFlow* or *FanFlow*. **FillExtendedArea** is *False* by default.

There's no changes to the existing MotPD03 and MotPD07.

Nominal Speed for VSD

New configuration to indicate the nominal speed is added in interaction window. All graphic element which contain numerical speed is added with new parameter **SpeedPresentation**. It has option to present the speed as *Unit* (default) or *Percentage*. If *Unit* is selected, it will show the speed as it is in MV with unit as defined interaction window. If *Percentage* is selected, it will convert the MV into percentage based on the nominal speed defined.

By default, nominal speed is 0. Nominal Speed indication in Object Display is only visible when it is not 0.

Others

- Regulatory valve faceplate improvement (Man01, PID01, PID01A and Ratio01)
    - Faceplate buttons are dimmed when **BalIn** or **Clamp** is activated.
    - Input *ManOut* is disabled if **ManEnbl** is *False*.
    - Input *AutoSP* is disabled if **AutoEnbl** is *False*.
- Add ActuatorVisibility for Valve01 graphic elements Valve01PD00-02. Default is *False*.
- The column width in Alarm and Event ConfigurationTemplate is updated.
- The column width in Trend Template is updated. Extrapolate column for each trace is set to *Value* instead of *None*.
- PP_Simulation is improved. It’s not added as part of the system installation, but can manually imported from the installation path folder.

PPLib800xA 5.3

System Extension

Instead of manual importing, PPLib800xA 5.3 is now using system extension **Pulp and Paper Library**. Loading the system extension will importing the library and its dependency in the correct order.

Licensing

Starting from PPLib800xA 5.3, the use of this library will require the sla file containing library license to be added into the License Entry. The licensing is scaled on the installed controller capacity where the library is used.

Paper Expansion Library

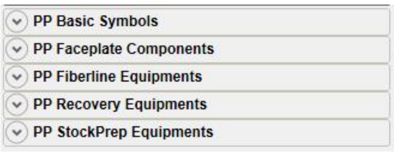
PP_PaperExPLib is now part of standard PPLib800xA. It contains Refiner01 function block which act as operator interface to Refiner01A for the refiner control.

Power Expansion Library

PP_PowerExpLib is a new expansion library for power area. At present, it contains 2 function blocks used for soot blower. SBSeq01 is for controlling the sequence of soot blowers while SBMotor01 is the soot blower motor.

Graphic Toolbox

A set of predefined graphic toolbox is provided with this release. It contains standard graphic of several equipment in pulp and paper.



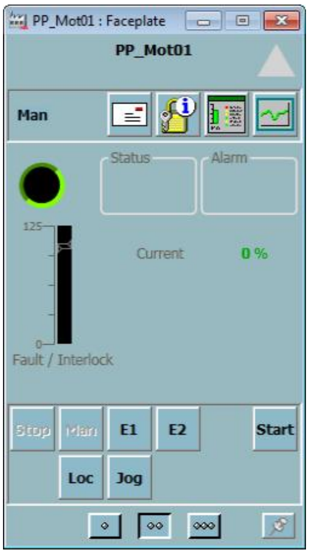
PPSupport Tool

PPSupport is replacing PPCRT. It can be used to read and write parameters in bulk from any function block from user defined library. It can also be used to compare value between backup file and online system. Refer to *3BTG811792-3054 User Guide - PPSupport Tool.pdf* for more information.

Resource Backup Tool

Resource backup tool is included in this release. It can be used to back up the customized NLS and Logical Color from the existing installation and restored it again after library upgrade. However this tool only support for upgrade from PPLib 5.1 or later only.

Faceplate button



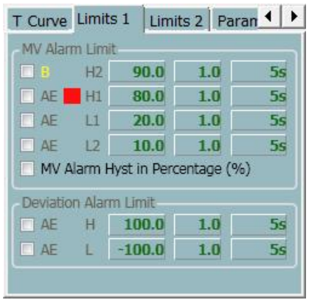
Faceplate button is dimmed when not available in certain mode or blocked in configuration. This is further improved to help operator awareness.

Faceplate button representing state of the object current in will be also dimmed.

For example, when in Man mode, the Man button will be dimmed. When motor is stopped, Stop button will be dimmed, etc.

This feature will focus the attention on the buttons that are actually available to perform a function at any given time.

Repeat Control



MV Alarm Limit

☐	B	H2	90.0	1.0	5s
☐	AE 🟥	H1	80.0	1.0	5s
☐	AE	L1	20.0	1.0	5s
☐	AE	L2	10.0	1.0	5s

☐ MV Alarm Hyst in Percentage (%)

Deviation Alarm Limit

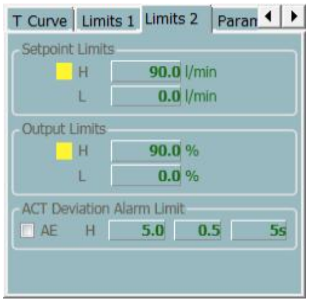
☐	AE	H	100.0	1.0	5s
☐	AE	L	-100.0	1.0	5s

Repeat Control is now added for MV, Deviation and Actuator Deviation alarm limits for *AIC01*, *AIS01*, *Flow01*, *Man01*, *PID01*, *PID01A* and *Ratio01*. It is also added for *DIC01* and *DIS01*.

If Repeat Control is enabled (ticked checkbox) and AEConfig = 1, the number alarm on and off repetitions without acknowledge is limited to 3.

The max limit can be adjusted from project constant **cRPCtrl**. Note that changing this value will affect Repeat Control limit for all objects in that project.

Actuator Deviation Alarm



Setpoint Limits

🟨	H	90.0 l/min
	L	0.0 l/min

Output Limits

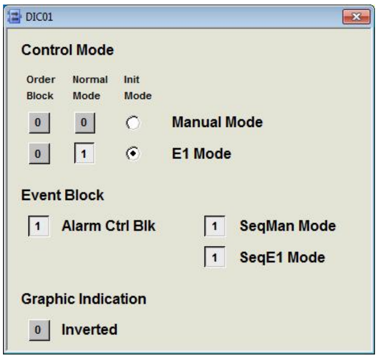
🟨	H	90.0 %
	L	0.0 %

ACT Deviation Alarm Limit

☐	AE	H	5.0	0.5	5s

Man01, *PID01* and *PID01A* now support actuator deviation alarm limit. It monitors the difference between the output *Out* or *OutP* and the actual valve position *ActPos*. If the difference is greater than the limit, *OutDev_GT_H* will be set to *True*. Alarm is generated when AEConfig is = 1. No changes in the control process.

Inverted Indication



Control Mode

Order Block	Normal Mode	Init Mode	
0	0	○	Manual Mode
0	1	●	E1 Mode

Event Block

1 Alarm Ctrl Blk

1 SeqMan Mode

1 SeqE1 Mode

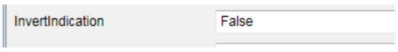
Graphic Indication

0 Inverted

DIC01 and *DIS01* now support inverted indication which can be configured from interaction windows.

When **InPar.InvertIndication** is set to *True*, **Value** is *False* will be represented with a full green symbol. This will apply to all symbol indication in graphic elements, faceplate and object display for that particular object. Text presentation in DIC01PD03 and DIS01PD03 is not inverted.

For Valve01, graphic element Valve01PD00, Valve01PD01 and Valve02PD02 now support invert indication which can be configured from **InvertIndication** parameter on the graphic properties. Changes in this parameter will effect on the particular graphic element only.



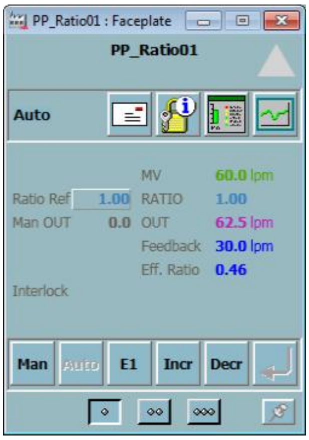
InvertIndication	False

This can be applied when Valve01 is used to control 2 valves with reverse action such as pilot valves on gas burners.

Actual Ratio

Ratio01 is added with new input parameter called **FeedBack**. It can be connected to feedback signal from process. With this information and ratio parameters, *Ratio01* will then calculate the effective ratio **EffRatio**.

To show the feedback and effective ratio on faceplate, **Show Feedback** must be enabled from interaction windows.



PP_Ratio01 : Faceplate

PP_Ratio01

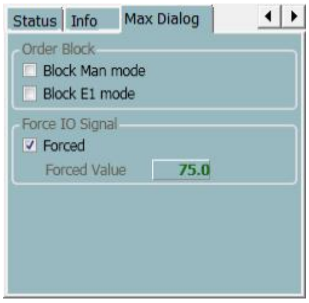
Auto

		MV	60.0 lpm
Ratio Ref	1.00	RATIO	1.00
Man OUT	0.0	OUT	62.5 lpm
		Feedback	30.0 lpm
		Eff. Ratio	0.46

Interlock

Force I/O Signal

I/O signal can be forced directly from faceplate *Max Dialog* tab for *AIS01*, *AOS01*, *DIS01* and *DOS01*.



Order Block

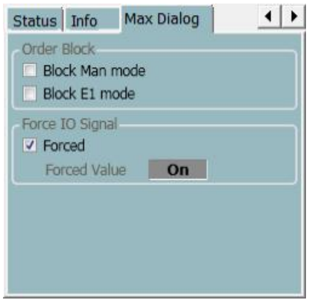
☐ Block Man mode

☐ Block E1 mode

Force IO Signal

☑ Forced

Forced Value 75.0



Order Block

☐ Block Man mode

☐ Block E1 mode

Force IO Signal

☑ Forced

Forced Value On

Load and memory usage

Internal code optimization is done in this release and PPLib 5.2 rollup 2. Compared to PPLib 5.1 or PPLib 5.2, the load reduction is expected about 2-15% depending on the object types used, controller cycle times and controller types. However, due to additional features, there’s an increase in the memory usage between 1-2%.

Others

- Functional Description can be accessed from Extended Faceplate.
- Status of limit switches and torque switches is presented in faceplate **Signal** tab for on-off valve and motorized valve.
- Simulation can be enabled/disabled from faceplate button. Enabling or disabling simulation will generate an event.
- Object with alarm shelved is presented with yellow frame on the graphic element similar to alarm blocking.

- A new input parameter to present motor temperature, **MT**. It's displayed in the Object Display.
- Motor current presentation of an object is configured in the interaction windows and it will be applied in the graphic element, faceplate element and object display. It's now possible to insert a graphic element with different presentation. Graphic element *MotPD52* has been added with property called **CurrentPresentation**. It has three options, *Default* (follows interaction windows), *Percentage* or *Unit*.
- Fung IV now support descending order in X tab.

PPLib800xA 5.2 Rollup 2

Logical Colors

New colors are added in the Pulp and Paper Library Colors aspect.

Previously, bargraph in faceplate element is using the same logical color as numeric and unit. New logical color is now assigned for bargraph to allow numeric and unit color customization without affecting the bargraph color.

Example: **ppFaceplateMV** is used for numeric and unit of MV, while **ppFaceplateMVBar** is used for bargraph.

Logical color name	RGB Value		Description
ppFaceplateACTBar	0,0,255	🟦	Actuator color for bargraph on faceplate
ppFaceplateCurrentBar	0,165,0	🟩	Current color for bargraph on faceplate
ppFaceplateMVBar	83,166,0	🟩	Measured value color for bargraph on faceplate
ppFaceplateMVForcedBar	255,245,51	🟨	Forced measured value color for bargraph on faceplate
ppFaceplateMVManBar	255,245,51	🟨	Manual measured value color for bargraph on faceplate
ppFaceplateOUTBar	182,36,185	🟪	Output color for bargraph on faceplate
ppFaceplateSPBar	60,128,188	🟦	Setpoint color for bargraph on faceplate

Some new logical colors are also added for user defined Trend.

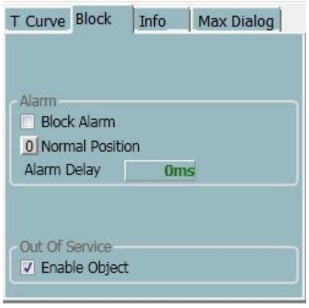
Logical color name	RGB Value		Description
ppTrend01	102,204,0	🟩	Trace color for user defined trend
ppTrend02	131,235,235	🟦	Trace color for user defined trend
ppTrend03	255,118,228	🟪	Trace color for user defined trend
ppTrend04	0,0,255	🟦	Trace color for user defined trend
ppTrend05	255,245,51	🟨	Trace color for user defined trend

ACS850 and ACS880 support

Dricon_S now support ACS850 and ACS880. The selection of ACS drive can be set from parameter **DriveType**. The table below shows the correlation between the ACS selection, DriveType and usage of DrivePar.

ACS Type	DriveType	DrivePar.Act3	DrivePar.Act4	DrivePar.Act5	DrivePar.Act6
ACS550	1	Fault Word 1	Alarm Word 1	Fault Word 2	Alarm Word 2
ACS600	2	Fault Word 1	Alarm Word 1	Fault Word 2	Alarm Word 2
ACS800	3	Fault Word 1	Alarm Word 1	Fault Word 2	Alarm Word 2
ACS850	4	1st Active Fault	1st Active Warning	2nd Active Fault	2nd Active Warning
ACS880	5	1st Active Fault	1st Active Warning	2nd Active Fault	2nd Active Warning

Alarm Delay for Digital Input



Alarm
□ Block Alarm
0 Normal Position
Alarm Delay 0ms

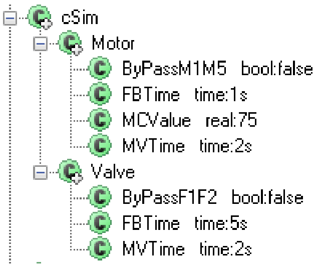
Out Of Service
☑ Enable Object

DIC01 and DIS01 now support alarm configuration for normal position and alarm delay from faceplate element **Block**. *Normal Position* checkbox and *Alarm Delay* input field is visible when alarm and event is enabled (AEConfig > 0).

Simulation

Internal simulation built in some function blocks can be adjusted based on the required conditions. This will help during application testing.

New project constants are added in PP_ElementLib.



```
cSim
    Motor
        ByPassM1M5  bool:false
        FBTime  time:1s
        MCValue  real:75
        MVTime  time:2s
    Valve
        ByPassF1F2  bool:false
        FBTime  time:5s
        MVTime  time:2s
```

For motors (*Dricon_S*, *Mot01*, *Mot02*, *MotFreq* and *UMC22*) and motorized valve (*Motval01*, *Motval02*, *UMC22_Act* and *UMC22_Act02*), control circuit *M1* to *M5* is still monitored by default during simulation. If *M1* to *M5* is to be ignored, then set **cSim.Motor.ByPassM1M5** to *True*. Motors will not trip if any of the *M1* to *M5* is *False* and motor is in simulation. If motor is not in simulation, it will still trip even if **cSim.Motor.ByPassM1M5** to *True*.

Motor running feedback is simulated *1s* after start command is issued. This is defined in **cSim.Motor.FBTime**. For motor current, **cSim.Motor.MCValue** will define the simulated current in percentage. Default is *75%*.

cSim.Motor.MVTime set the filter time of simulated speed in *Dricon_S* and *MotFreq* or actuator position in *Motval02* and *UMC22_Act02*. Simulated speed or actuator position is filtered by *2s* from the speed or actuator setpoint.

For Valve01, fault *F1* and *F2* can be ignored during simulation by setting **cSim.Valve.ByPassF1F2** to *True*. Limit Switch Open and Close will be simulated from Open or Close command after a time period defined in **cSim.Valve.FBTime**.

cSim.Valve.MVTime will set filter time of simulated MV in *PID01* and *PID01A*. Simulated MV is filtered by *2s* from the output.

NOTE: PP_ElementLib is a password protected library, hence the project constants there cannot be modified. You can only modify the value through the project constants in your project.

Load Reduction

Some internal code optimization is implemented in this rollup. Depending on how the library usage and controller cycle time, the load will reduce by 2 to 10% compared to using the previous SV5.1 libraries (5.1-0 or 5.2-0).

PDQ22 support

Objects in the PP_UMC100Lib now support both PDP22 and PDQ22. Selection of PDP22 or PDQ22 can be set from the input parameter **PlugType**. If set to 1, then PDP22 is used. If PDQ22 is used, then set to 2.

The connection to plug unit status remains on **PDPUnitStatus**. It shall be connected to the unit status of PDP22 or PDQ22 in the hardware structure.

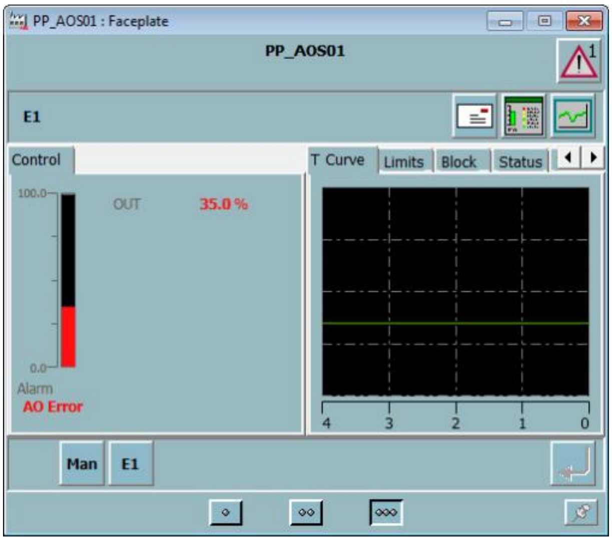
PPLib800xA 5.2 Rollup 1

Language Packs

PPLib800xA 5.2 rollup 1 now comes with Chinese and Swedish language pack.

Analog Output (AO) status monitoring

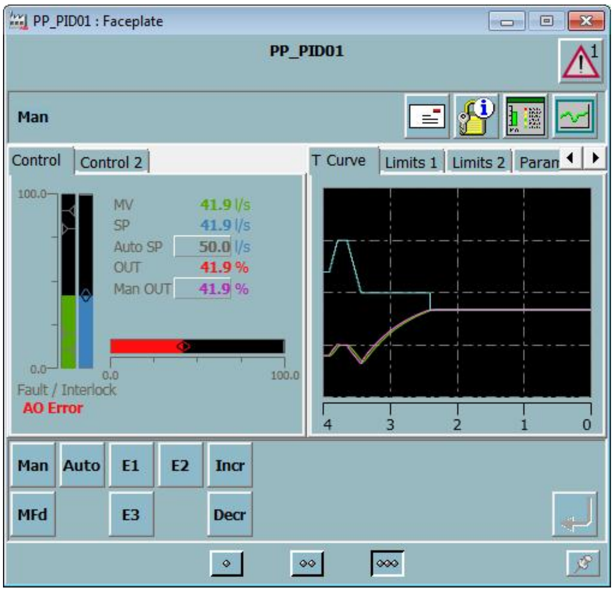
Analog output status now can be monitored with alarm handling for *AOS01*, *Man01*, *PID01* and *PID01A*.



OUT 35.0 %

Alarm
AO Error

In *AOS01*, the signal status is obtained directly from **IOSignal** variable and alarm configuration is done from the Interaction Window. When an AO error occurs whilst AEConfig is set to 1, all graphic elements, except AOS01Mode, will be indicated in alarm color (default is red).



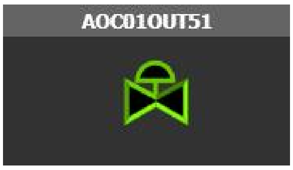
MV 41.9 l/s
SP 41.9 l/s
Auto SP 50.0 l/s
OUT 41.9 %
Man OUT 41.9 %

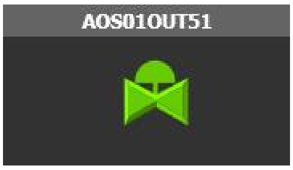
Fault / Interlock
AO Error

In *Man01*, *PID01* and *PID01A*, the signal status is obtained from new input parameter **AOErr** and alarm configuration is done from the Interaction Window. When there is an error in the analog output signal, the control mode will switch to *Manual* mode. Alarm indication will be displayed in graphic elements of MV and OUT, but alarm will only be indicated for OUT signal in faceplate and object display.

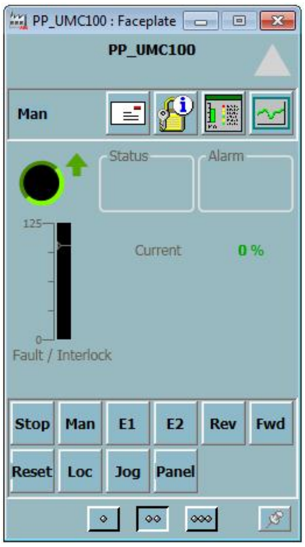
Graphic Element

New graphic elements are added in AOC01 and AOS01.





Panel Request



Panel Request is a new function in the Smart Device library package. Panel mode can be requested by an operator from the Panel faceplate button.

This sends a command to the MCC and upon receiving a feedback that the mode has been switched, it will be indicated by Panel mode. The Panel mode can be subsequently released by pressing the Man button and the object will change to Man mode.

Section 3 - Changed Features

This section describes the changed features in this release and previous releases.

PPLib800xA 5.4

MCAlarm

With the additional Low Current alarm, MCAlarm data type is changed from **AlarmParCurr** to **Alarm2LimitCurr**. If input parameter MCAlarm is connected to a variable, connection must be updated accordingly.

Alarm and event handling

With the improvement on the alarm and event handling, the following has been affected:

- Changes in **Class** and **Severity** requires warm download to take effect.
- Input parameter **EventName** is removed. Interlock text can be incorporated into alarm and event automatically using **PPLib800xA AE Uploader** tool.
- Project constant has been restructured. Unused project constant is removed. All removed project constant is kept in **PP_ProjConstOld 1.0-0.afw** under **Old afw** folder in the installation media.
- AE translator has been consolidated into Library Structure. AE Translator aspect in each object type is removed. All removed AE translator aspects are kept in **Old afw** folder in the installation media.
- Changes in parameters of B16SimpleEvent and B32SimpleEvent. **Desription**, **EventName** and **EventType** are removed. A new parameter **Prefix** is added. Changes are done to improved the execution and to support multiple event when several bits become active at the same time.
- Compiler switches to exclude warning of multiple calls to the same function block must be set to **Allowed** for all Pulp and Paper Library in every project. Refers to installation/upgrade procedure to set it up.

PID01A Tuning Parameter

PID01A always has warning of **Parameter with direction 'in by_ref' may be modified through parameter PID01A_In1.InPar**. PID01A stores the tuning parameter in InPar which is accesible from interaction window or faceplate. This is where user normally enter the values. However, there is a function for Apply and Undo which come from PidCC used internally in PID01A to confirm the changes in tuning parameter. This means the InPar will be also written from the control module. As such, the above warning is generated.

To eliminate the above warning, these tuning parameters are removed from InPar and now are stored directly inside the function block. If tuning parameters are written by logic, use the **ExtCtrl** instead.

PID01 BalRef limitation

Among all control loops available, only PID01 which has BalRef limited by output limitation. This is now changed to reflect similar behaviour across control loops. When PID01 in Balance mode, the output will take the value of BalRef without being limited by output High or Low Limit. However, it’s still limited by the output range.

PPLib800xA 5.3 Rollup 2

No changed feature in this release.

PPLib800xA 5.3 Rollup 1

Torque Error in Motval

Torque Close Error and Torque Open Error are no longer treated as fault. Both are treated as alarm only. When torq error occurs, it’s still possible to operate the motval to opposite direction.

When Torque Close Error occurs, it’s still possible to send Open command from faceplate or program. Faceplate button Close is dimmed and Close command from program is also not valid. When Torque Open Error occurs, it’s still possible to send Close command from faceplate or program. Faceplate button Open is dimmed and Open command from program is also not valid.

Position Error in Motval

Position Error is no longer treated as fault. It is treated as alarm only. When travelling time is exceeding the limit defined, it’s still possible to operate the motval. Alarm will be generated.

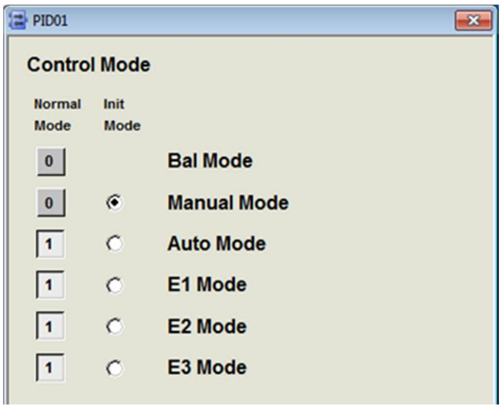
Jog SP in Variable Speed Motor

In previous releases, when motor in Jog mode, there’s no specific speed setting. Motor will send the speed setpoint based on the selected setpoint mode.

A new parameter named **JogRef** is added in MotFreq, Dricon_S and SmartVSD. When motor is in Jog Mode, the speed setpoint mode will automatically goes to Jog SP Mode. It will take the speed reference from JogRef. When motor leaving Jog Mode, the speed SP will resume to previously selected mode.

PPLib800xA 5.3

Normal Mode



PID01

Control Mode

Normal Mode	Init Mode	
0		Bal Mode
0	●	Manual Mode
1	○	Auto Mode
1	○	E1 Mode
1	○	E2 Mode
1	○	E3 Mode

Previously, normal mode is set when the object’s mode is the same as init mode. This has been changed now.

Normal mode defines the mode(s) where object shall be operated normally. More than 1 mode can be assigned as normal mode depending on how the object is used. By default, normal mode to those modes like *Auto*, *E1*, *E2* or *E3*.

Mode presentation in the graphic elements are indicated with *ppMode* and *ppNormalMode*. Both are assigned with the same color as default. If desired, project can implement different colors to highlight if object is not in the normal mode.

PPLib800xA 5.2 Rollup 2

Alarm Blocking Indication

In previous versions, yellow dash border appears in graphic elements whenever alarm is blocked, regardless whether the alarm is blocked by operator or program.

The yellow border shall indicate something that’s not normal that will get operator’s attention on the process graphic. When an alarm is blocked from program, the intention is to hide unwanted operator indication.

Therefore starting in this version, the yellow dash border will only appear in graphic elements when operator blocks the alarm. The *BX* indication in the faceplate element will remain as long as alarm is blocked, either by operator or program.

EventName

Changes in EventName value requires warm download to take effect.

InitMode

Changes in initmode value requires warm download to take effect. Otherwise, when object is disabled then enabled again, it will go back to previous init mode.

Section 4 - Correction

This section describes the correction in this release and previous releases.

PPLib800xA 5.4

Filt_1P

Object Types	Issues	Correction
Filt_1P	Output is not updated to high limit if the input value multiplied by the scale is greater than high limit	This problem has been corrected

Fung_1V

Object Types	Issues	Correction
Fung_1V	When XTab is configured in decreasing order or YTab in decreasing order with Bal, it may cause the connected task being aborted	This problem has been corrected

PPLib800xA 5.3 Rollup 2

Library Licensing

Object Types	Issues	Correction
	When PP_FunctionLib, PP_UMCLib, PP_PaperExpLib or PP_PowerExpLib are used together, additional library license quanta required is multiple than it should be.	This problem has been corrected. Refer to 3BTG811792-3060-1 PPLib800xA Incorrect Calculation of Library License Quanta

PPLib800xA 5.3 Rollup 1

Aspect customization

Object Types	Issues	Correction
	It's not possible to modify the Trend Signal Properties aspect and AE Translator aspect in PPLib800xA 5.3	This problem has been corrected

Limit wind-up

Object Types	Issues	Correction
Filt_1P, INTegrator	Although, the output is limited to the correct OLL and OHL, internally the calculation is still continuing	This problem has been corrected

Integrator

Object Types	Issues	Correction
INTegrator	When Bal is released, the output doesn't take BalRef as base value, but revert back to previous value.	This problem has been corrected

Enable Permission

Object Types	Issues	Correction
All object types	No permission is assigned to the Enable parameters. It prevents the security definition to work properly	This problem has been corrected

Filter time

Object Types	Issues	Correction
AIC01, AIS01, Flow01	Changing filter time from 0 to large value causes the output value to drop	This problem has been corrected

Totalizer

Object Types	Issues	Correction
Totalizer01	Input parameter Reset doesn't reset the totalizer value	This problem has been corrected

Valve01

Object Types	Issues	Correction
Valve01	If local open or close remains active, it's not possible to open or close valve in other modes.	This problem has been corrected
	Damper symbol doesn't show correct information when there is OFF alarm (Ellipse and Line indication can't be seen)	This problem has been corrected

Smart Device 2 direction

Object Types	Issues	Correction
UMC22, SmartMCC	If Reversing is not enabled, it's still possible to run motor in reverse in Panel Mode or using JogRev in Jog Mode.	This problem has been corrected

PPLib800xA 5.3

Limit indication

Object Types	Issues	Correction
AOC01, AOS01	No clear indication which upper and lower limit is used from faceplate view	This problem has been corrected

Adding small value to large value

Object Types	Issues	Correction
INTegrator, Total01, Totalizer01	Adding small number to a very large number sometimes doesn’t add up correctly	This problem has been corrected

Clamp release

Object Types	Issues	Correction
Man01	If TrackA = 0 and TrackB = True, when Clamp/BalIn is released, the Out will follow to previously set Manual Out instead of maintaining the Out value. This issue is introduced in 5.2 rollup 1	This problem has been corrected

PPLib800xA 5.2 Rollup 2

Alarm for I/O Error

Object Types	Issues	Correction
AIS01, AOS01, Flow01, Man01, PID01, PID01A	Error from I/O signal still generates alarm even after object is not enabled (Out of Service)	This problem has been corrected

Motor Current Alarm Limit

Object Types	Issues	Correction
Mot01, Mot02, MotFreq, Motval01, Motval02	Current alarm limit is not indicated correctly in the bargraph for graphic element *MotPD51*, faceplate element Extended Control and Object Display when Current in % is selected	This problem has been corrected

Ratio Unit

Object Types	Issues	Correction
Ratio01	Unit for Ratio High and Low limit in interaction window is following MV unit instead of Ratio unit	This problem has been corrected

Negative Value for Totalizer

Object Types	Issues	Correction
Totalizer01	Negative input value will reduce the total value	This problem has been corrected. Negative input is treated as zero

PPLib800xA 5.2 Rollup 1

Trend Templates

Issues	Correction
Traces color orders in PP Library Control Object Trend template are currently defined as ppTrendMV, ppTrendOUT, ppTrendSP and ppTrendActuator	Traces color order has been changed to ppTrendMV, ppTrendSP, ppTrendOUT and ppTrendActuator
PP Library Standard Object Trend contains only 1 trace	It has been expanded to contain 4 traces with color defined as ppTrendMV, ppTrendSP, ppTrendOUT and ppTrendActuator

Comp_R

Object Types	Issues	Correction
Comp_R	Output I_GT_L1 is set to true when I > L1 but still not reset even when I < (L1 - LHys).	This is now fixed. Refer to changes below.
	- Output I_LT_H1 is set to true when I < H1 and only set to false when I > (H1 + HHys) - Output I_GT_L1 is set to true when I > L1 and only set to false when I < (L1 - LHys)	Changes in reset: - I_LT_H1 is set to true when I < H1 and set to false when I ≥ (H1 + HHys) - I_GT_L1 is set to true when I > L1 and set to false when I ≤ (L1 – LHys)

Counter

Object Types	Issues	Correction
CountL	Out value will continue to increase beyond the HL limit and decrease below the LL limit.	Out is now limited to low limit LL and high limit HL
	When input L and C are set simultaneously, the Out value is preloaded with the value at input In, but no up or down count is performed.	When input L and C are set simultaneously, the Out value is preloaded with the value at input In, then an up or down count is performed.

INTegrator

Object Types	Issues	Correction
INTegrator	When Bal is activated, the output O takes the value of BalRef as it is and not limited by LL and HL limit	This problem has been corrected

Min_R and Max_R

Object Types	Issues	Correction
Min_R, Max_R	A change in input value within deadband still causes output to be updated with new value.	This problem has been corrected

Time01

Object Types	Issues	Correction
Time01	Time indication is not presented correctly in RelativeTime01.	This problem has been corrected

Total01

Object Types	Issues	Correction
Total01	Q and Q_Prev are lost during the library upgrade or cold download. Backup value cannot be restored.	This problem has been corrected

Faceplate InputField stepsize

Object Types	Issues	Correction
Man01, PID01, PID01A, Ratio01, Motval02, NewACT02, UMC22_Act02, UMC100ACT02	Stepsize for input field in Reduced Control and Extended Control faceplate element is fixed to 1% of range.	Stepsize is no longer fixed to 1% but linked to the value In/Dec parameter.

IOSignal Range

Object Types	Issues	Correction
AIS01	*Value* is always limited to Min and Max range. IOSignal.IOValue value still can be lower than Min when undercurrent or higher than Max when overcurrent when IO module supports.	Internal range limitation in widened by 10% to allow the undercurrent and overcurrent value.

MV Alarm Hysteresis

Object Types	Issues	Correction
AIC01, AIS01, Flow01, Man01, PID01, PID01A, Ratio01	Checkbox MV Alarm Hysteresis in Percentage (%) is not shown in Limits faceplate element when alarm limits are configured as event only	This problem has been corrected.

Motor Current Presentation

Object Types	Issues	Correction
Mot01, Mot02, MotFreq, Motval01, Motval02	When Current presentation in % is selected, the highest alarm limit that can be entered in the Limits faceplate elements is Max Current.	The highest alarm limit can be entered is corrected to (Max Current *100)/Rated Current

Motor Speed

Object Types	Issues	Correction
Dricon_S, MotFreq, Smart VSD	When motor is not running and minimum speed limit is > 0, the SP_OUT is still set to the minimum speed limit.	When motor is not running, SP_OUT is now set to 0 regardless of the minimum speed limit.

Motor Interlocks

Object Types	Issues	Correction
Mot02, UMC22, SmartMCC	When motor is in Jog Mode and JogFunc is not used, motor can be started and not stopped even if IC interlock is active.	Motor cannot be started and will be stopped if IC interlock is active.
	When motor is running reverse in Jog Mode and JogFunc is not used, motor will be stopped when RunInt1 is active.	When motor is running reverse in Jog Mode and JogFunc is not used, motor will not be stopped when RunInt1 is active.

Motorized Valve

Object Types	Issues	Correction
Motval01, Motval02, UMC22_Act, UMC22_Act02, SmartACT01, SmartACT02	The previous status was not retained when a mode is changed.	A mode change will now retain the last status except during entering and leaving Jog Mode.
	IB interlocks are monitored when in Jog Mode.	IB interlocks are now bypassed in Jog Mode.
	In Jog Mode, when JogFunc is used, the open and close command can only be reset with Stop parameter.	In Jog Mode, when JogFunc is used, open and close command will be reset when JogFunc is released.

MCC Diagnostics

Object Types	Issues	Correction
UMC22, UMC22_Act, UMC22_Act02, NewACT01, NewACT02	Indication for Monitoring Data bit 12, bit 14 and bit15 is red color when bit is active.	Indication is changed to green color (ppDiagnosticsHealthy) when bit is active.

UMC100 Devices

Object Types	Issues	Correction
UMC100, UMC100ACT01, UMC100ACT02	Missing Alarm Configuration for Overload Warning in Interaction Window S3.	This problem has been corrected
UMC100	Missing event translator for Warning.	Resource id Warning is changed to W in AE Translator aspect.

Object Types	Issues	Correction
UMC100ACT01	Missing event translator for ICs.IC interlock.	Resource id ICs1Off, ICs1On, ICs2Off and ICs2On are added in AE Translator aspect.

Regulatory Control Valves

Object Types	Issues	Correction
Man01, PID01, PID01A, Ratio01	Releasing BalIn and triggering SeqE1, SeqE2, SeqE3 or SeqAuto at the same cycle time does not change mode to E1, E2, E3 or Auto. Mode still remained at Man mode.	This problem has been corrected
Man01, Ratio01	Out_EQ_LL and Out_EQ_HL are set to *True* regardless of Out value in the following conditions: - BalIn or Local is active (set to *True*) and/or - Either IB1, IB2, IB3 or IB4 interlocks is active (set to *False*)	Out_EQ_LL is: - *True* if Out ≤ L limit - *False* if Out > L limit Out_EQ_HL is: - *True* if Out ≥ H limit - *False* if Out < H limit
Man01	Possibility of having different signal range for MV and OUT is missing in PPLib 5.1-0 and 5.2-0.	New parameter MVRange is added in InPar and existing parameter Range is renamed to OUTRange.
	Object Trend Display shows MV and Out in trace 1 and 2.	MV and Out is shown in Trace 1 and Trace 3 to accommodate the changes in Trend Template.
PID01, PID01A	OutEqLL and OutEqHL are set to *True* regardless of Out value in the following conditions: - Either IB1, IB2, IB3 or IB4 interlocks is active (set to *False*)	OutEqLL is: - *True* if OutP ≤ L limit - *False* if OutP > L limit OutEqHL is: - *True* if OutP ≥ H limit - *False* if OutP < H limit
	Traces order in the Object Trend Display are currently defined as MV, Out and WSP	Traces order in the Object Trend Display are corrected to MV, WSP and Out
	Internal simulation always assumes that the minimum range is 0	Internal simulation will use the InPar.MVRange.Min

Section 5 - Installation and Upgrading

This section describes the installation and upgrading procedure for installing this release.

New Installation

Installation

To install Pulp and Paper Library on aspect server nodes:

1. Insert PPLib800xA Installation CD into CD/DVD drive.
2. Double click on the **Pulp and Paper Library.exe** in the *Installation* directory to begin installation.
3. Click on **Automated Installation**
    a. Select **Aspect Server** if installing on aspect server nodes. This will install *PPLib800xA* only.
    b. Select **Engineering Station** if installing on engineering nodes. This will install *PPLib800xA*, *PPLib800xA CBM Help*, *PPSupport*, *PPLib800xA AE Uploader* and *HwStatus Display Generator*.
4. Follow the Installation Wizard to complete the installation.

Post Installation

Post installation covers the procedure to load PPLib800xA system extension into 800xA system. This can be performed using System Extension Load or System Extension Maintenance.

System Extension Load is used if the system has no Pulp and Paper Library system extension.

System Extension Maintenance is used if the system has Pulp and Paper Library system extension which was introduced starting from PPLib800xA 5.3.

To load/update system extension from primary aspect server:

1. Start the Configuration Wizard from the primary Aspect Server node.
2. Open System Extension Load or System Extension Maintenance.

   To open the System Extension Load dialog box by going to:

   System Administration > Select System > System Extension Load

   To open the System Extension Maintenance dialog box by going to:

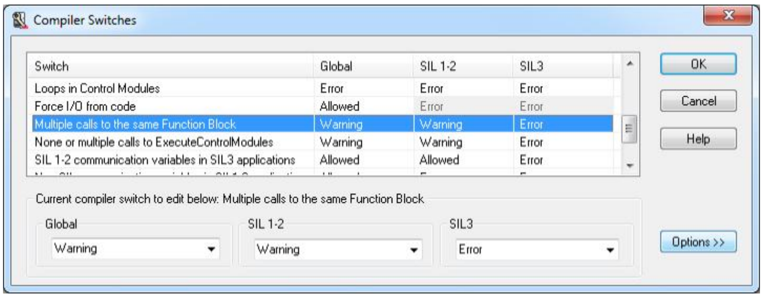
   System Administration > Select System > System Extension Maintenance
3. Select **Pulp and Paper Library** from the list in the left pane and move it to the list in the right pane by clicking **>**.
4. If Function Designer support is required, add **Pulp and Paper Library for Function Designer** to the right pane.

5. If all system extension in the right pane are marked with green check mark, then click **Next** and the **Apply Settings** dialog box appears.
6. Click **Finish** to load the system extensions.

Compiler Switches Setting

Once project is created and Pulp and Paper Library connected to project, follow the steps below to configure the compiler switches setting:

1. Open Control Builder project.
2. Right click on the project name, select **Settings > Compiler Switches**.
3. Select **Multiple calls to the same Function Block**. Then click on **Option >>** button.

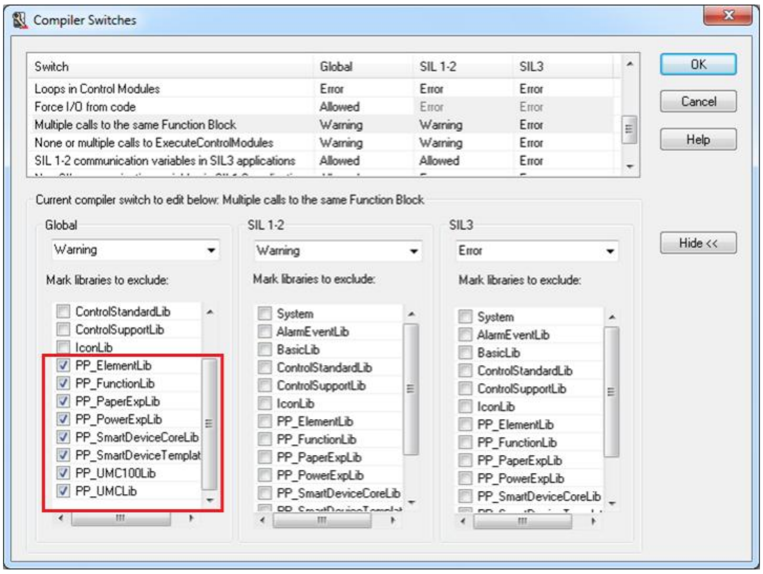


Switch	Global	SIL 1-2	SIL3
Loops in Control Modules	Error	Error	Error
Force I/O from code	Allowed	Error	Error
Multiple calls to the same Function Block	Warning	Warning	Error
None or multiple calls to ExecuteControlModules	Warning	Warning	Error
SIL 1-2 communication variables in SIL3 applications	Allowed	Allowed	Error

Current compiler switch to edit below: Multiple calls to the same Function Block

Global: Warning
SIL 1-2: Warning
SIL3: Error

4. Exclude libraries which belongs to Pulp and Paper Libraries.



Switch	Global	SIL 1-2	SIL3
Loops in Control Modules	Error	Error	Error
Force I/O from code	Allowed	Error	Error
Multiple calls to the same Function Block	Warning	Warning	Error
None or multiple calls to ExecuteControlModules	Warning	Warning	Error
SIL 1-2 communication variables in SIL3 applications	Allowed	Allowed	Error

Current compiler switch to edit below: Multiple calls to the same Function Block

Global: Warning

Mark libraries to exclude:

- □ ControlStandardLib
- □ ControlSupportLib
- □ IconLib
- ☑ PP_ElementLib
- ☑ PP_FunctionLib
- ☑ PP_PaperExpLib
- ☑ PP_PowerExpLib
- ☑ PP_SmartDeviceCoreLib
- ☑ PP_SmartDeviceTemplat
- ☑ PP_UMC100Lib
- ☑ PP_UMCLib

SIL 1-2: Warning

Mark libraries to exclude:

- □ System
- □ AlarmEventLib
- □ BasicLib
- □ ControlStandardLib
- □ ControlSupportLib
- □ IconLib
- □ PP_ElementLib
- □ PP_FunctionLib
- □ PP_PaperExpLib
- □ PP_PowerExpLib
- □ PP_SmartDeviceCoreLib

SIL3: Error

Mark libraries to exclude:

- □ System
- □ AlarmEventLib
- □ BasicLib
- □ ControlStandardLib
- □ ControlSupportLib
- □ IconLib
- □ PP_ElementLib
- □ PP_FunctionLib
- □ PP_PaperExpLib
- □ PP_PowerExpLib
- □ PP_SmartDeviceCoreLib

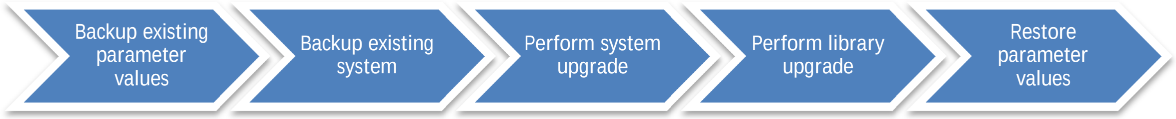
5. Click **OK** button to continue.

The above setting will only hide warning of *multiple calls to the same function block* inside the selected library.

It’s not recommended to switch off this warning completely to prevent error in application. If an application calls the same function block, warning will still be generated.

Upgrading PPLib800xA on an existing 800xA system

The procedure for upgrading to the latest PPLib800xA is as follow:



Backup existing parameter values

Note: Ensure that projects are downloaded to the controllers and verify that live data can be observed in the 800xA system before attempting the following steps.

1. Use Windows Explorer to locate a number of Microsoft Excel documents (*.xlsm) in the *Upgrade Tools / ColdRetain* directory.

Note: Use only Microsoft Excel 2003 (32- bit) or later. Install FileFormatConverters (download from Microsoft website) if the installed Excel does not support xlsm.

2. Open the Excel document that corresponds to your existing version of PPLib800xA.

   Upgrade from PP Library <Existing version> to PP Library 5.4.xlsm
   For example: Upgrade from PP Library 4.0-1 to PP Library 5.4.xlsm

Note: Always use the provided Excel file for the upgrade as some parameters may be automatically mapped internally when you restore.

3. Beginning with the first control project or application, retrieve all the PPLib800xA objects from the project / application by performing the following:
   a. Click on **Retrieve Objects** button to open the **Retrieve Objects** dialog box.
   b. Click on **Browse Object** button and browse for the desired project / application in the Control Structure. Click **Ok** to confirm. *(The selected path can be seen in the* ***Retrieve Objects*** *dialog box)*
   c. Select **Append Existing Data** if new data is to be added to the next available row. Otherwise, select **Clear Existing Data** to remove any existing data.
   d. Click on **Read Parameters** button to open the **Read Parameters** dialog box.
   e. Specify the connectivity server by selecting the appropriate options.
   f. Click on **Read Parameters** button to start retrieving the values of the parameters from the system.
   g. Visually verify that all parameters have been retrieved successful.
   h. Save and assign a unique name to this Excel document.
   i. Repeat step (a) to (h) for all other projects or applications.

Note: Resolve all errors before proceeding with the next steps.

4. Release any reservation in the system and close any opened workplaces and Control Builder.

5. If upgrading from PPLib800xA 5.1-0 or later, modification on resources such as **Logical Color Definition** and **NLS Resource Manager** aspect category which belongs to PP Library can be backed up.
   a. Double click **ResourcesBackupApp.exe** in the *Upgrade Tools / Resources* directory to launch the **Resource Backup** application.
   b. Leave everything as default and click on **Backup** button.
   c. Click on **OK** button when backup is completed.

Backup existing 800xA system

Follow the instructions described in the maintenance documents of the installed 800xA system to back up the existing system. It is highly recommended to take a backup of the Functional Structure, Control Project and all used libraries.

System Upgrade (if applicable)

The existing system must be upgraded to System 800xA 5.1 including the Control Project(s).

Follow the table below for upgrading from different system version: SV3.1 SP3, SV4.1 and SV5.0 SP2. Follow system upgrade procedures as describe in **3BSE036342-511 System 800xA 5.1 Upgrade** in 800xA SV5.1 Documentation.

No	Procedure	Upgrade from		
		SV3.1	SV4.1	SV5.0
1	Upgrade the existing 800xA system to System 800xA 5.0 SP2	√		
2	If upgrade from PPLib 5.0-1, 5.0-2 or 5.0-3: a. Double click on the LoadPPLib800xA505.bat in the *Upgrade Tools / Special Upgrade* directory to import the temporary library. b. Launch a Plant Explorer Workplace and browse to the Control Structure. Remove all Text Properties aspects associated with PP_FunctionLib, PP_UMCLib and PP_PaperExpLib from the Control Structure. *Hint: use the Find Tool to search for all Text Properties aspects in Control Structure*. c. Update all the projects and applications by launching the Control Builder. The temporary libraries (5.0-5) will be automatically updated in the projects. d. If Function Designer is used: • Double click on the LoadPPLib800xA505FD.bat in the *Upgrade Tools / Special Upgrade* to import the FD aspect. • Generate the code for all diagrams. e. Reload any customized aspects that may have been removed during the import. f. If VB Graphics is used, deploy all VB Graphics using the Display Tool.			√
3	Follow the system upgrade procedure as described in 3BSE036342-511 System 800xA 5.1 Upgrade	√	√	√

Library Upgrade

The following library upgrade procedure is performed in System 800xA 5.1 environment.

1. Follow the instructions as described in New Installation to install the new library into 800xA System.
2. *(Optional, only for existing graphics in VB)* Double click on the **LoadVBGraphics.bat** in the *Installation / VB Graphics Aspects* directory to import the VB aspects.
3. Update all the projects and applications by launching the Control Builder.

Library upgrade from PPLib 5.0-0 or later will see the libraries are automatically updated in the projects and application. If upgrading from PPLib 4.0-5 or earlier, the new libraries must be manually inserted to the projects and applications.

4. Configure the compiler setting as described in Compiler Switches Setting.
5. For each project or application, update any control logic changes if necessary.

   Refer to Appendix A - Special Note depending on the original library version installed.
6. If Function Designer is used, generate the code for all diagrams.
7. Perform a ***warm*** download of the project to the controller.
8. Restore the parameter values that were archived earlier.
   a. Open the Excel document containing the parameter values for this project / application.
   b. Click on **Write Parameters** button to open the **Write Parameters** dialog box.
   c. Select the appropriate Connectivity Server node.
   d. Click **Write Parameters** to start the restore process.
   e. Once completed, click **Check Parameters** to verify that all parameters have been restored in the controller.
9. Retune PID01 parameters if affected based on the condition described in Appendix A - Special Note.
10. Repeat step 3 to 9 for all the projects / applications.
11. Reload any customized aspects that may have been removed during the import.
12. *(Optional, only for existing graphics in VB)* Migrate all existing VB graphics to PG2 graphics using the steps listed below.
    a. Deploy all VB Graphics using the **Display Tool**.
    b. Run **VB Graphics Migration Tool** from 800xA to migrate any existing VB graphics to PG2 graphics.
    c. Remove VBPG aspects from the system. Use System 800xA VBPG Search Tool to find VBPG aspect.
    d. Remove VB graphic extension library from PP_FunctionLib, PP_UMCLib and PP_PaperExpLib from Library Structure.
13. *(Optional, only for upgrade from PPLib800xA 5.1 or later)* Restore modification on resources such as **Logical Color Definition** and **NLS Resource Manager** aspect category.
    a. Double click **ResourcesBackupApp.exe** in the *Upgrade Tools / Resources* directory to launch the **Resource Backup** application.
    b. Leave everything as default and click on **Restore** button.
    c. For NLS, the default language set is English. Select other supported language if required. Any difference between the backup and the new installation will be displayed. Select the items to be restored and click on **Proceed** button.
    d. Click on **OK** button when backup is completed.

License Installation

The use of PPLib800xA requires license key to be incorporated into the 800xA system. To install the PPLibrary sla file, launch the License Entry program from the license server.

1. Select **Start > All Programs > ABB Industrial IT 800xA > System > Licensing > License Entry**
2. Select **File > Add Extension…** in the License Entry program
3. The Open dialog box appears. Find the license file (.sla) which contain the PPLib800xA license key and click **Open**.
4. PPLib800xA license is defined under **800xA Control and IO** group and **PPLib_QUANTA** feature.
5. Select **File > Exit** to exit the License Entry program when finished.

Section 6 - Technical Support

Reporting Error

Please send mail to ***PulpAndPaperCSSupport@sg.abb.com*** for reporting errors or technical support.

In all communication regarding questions or complaints about the function in PPLib800xA, please include version number of relevant library as well as the system version installed.

Appendix A - Special Note for Library Upgrade

Special Note contains the information need to be addressed during library upgrade. It list the changes of feature that may or may not affect the project applications.

Object Types	Special Notes	Existing PPLib version							
		1.x-x 3.x-x 4.0-x	5.0-0	5.0-1	5.0-2 5.0-3	5.1	5.2 5.2h1	5.2r1 5.2r2	5.3
Mot01, Mot02 MotFreq Motval01 Motval02	MCAlarm datatype is changed from AlarmParCurr to Alarm2LimitCurr. If MCAlarm is connected to variable, adjust the connection accordingly.					√	√	√	√
All object types	EventName is removed.					√	√	√	√
B16SimpleEvent B32SimpleEvent	Description, EventName and EventType is replaced with Prefix. Update accordingly.								√
Dricon_S MotFreq	Speed setpoint for JogMode is assigned to dedicated parameter *JogRef*. It can be set with fixed speed or logic.	√	√	√	√	√	√	√	√
	The parameters *OrdBlk* and *OprOrder* have been removed. Replace the connection with *InPar* and *Opr* accordingly.	√	√	√	√				
Comp_R	Hysteresis is used when calculating output *I_LT_H1* and *I_GT_L1*. If no Hysteresis is required, use *GT or LT* function block instead.	√	√	√	√	√	√		
Time01	You can choose from three different formats representation in RelativeTime01 aspect. They are *HourMinSecMilliSec*, *HourMin* and *HourMinSec* (*Default)*	√	√	√	√	√	√		
MotVal01 MotVal02 UMC22_Act UMC22_Act02	If *JogFunc* is used, you may issue a stop order by releasing *JogFunc* instead of the *Stop* parameter.	√	√	√	√	√	√		
	Mode changes will retain the last status except entering and leaving Jog Mode. Stop order should be added if you want MotVal to stop during mode changes.	√	√	√	√	√	√		
	IC1Opn, *IC2Cls*, *IA1Opn* and *IA2Cls* have been replaced by *IC1*, *IC2*, *IA1* and *IA2* respectively	√	√	√	√				
	The parameters *OrdBlk* and *OprOrder* have been removed. Replace the connection with *InPar* and *Opr* accordingly.	√	√	√	√				

Object Types	Special Notes	Existing PPLib version							
		1.x-x 3.x-x 4.0-x	5.0-0	5.0-1	5.0-2 5.0-3	5.1	5.2 5.2h1	5.2r1 5.2r2	5.3
Mot01, Mot02 UMC22, Valve01	The parameters *OrdBlk* and *OprOrder* have been removed. Replace the connection with *InPar* and *Opr* accordingly.	√	√	√	√				
Object Trends	• Any new objects instantiated after the upgrade will use the new Trend templates. Modify the Object Trend Display aspects in the Object Type structure to point to an existing template if required. • Any existing objects with no Trend templates configured will use the new Trend template. On the other hand, if a template exists, all existing objects will utilize the existing. • It is recommended to use function block parameters for trending as the number of HSI variables has been reduced as of PPLib800xA 5.1	√	√	√	√				
AIC01, AIS01 Flow01 GainSched	The attribute of *ExtCtrl* has been changed from *retain* to *by_ref*. To access these parameters' properties, connection must be done using a variable, instead of directly using function block. Otherwise, errors will be prompted in Control Builder.	√	√	√	√				
PID01A	Tuning parameter is removed from *InPar*. Use *ExtCtrl* if tuning parameter need to be controlled by application.					√	√	√	√
	HotInit parameter has been removed.	√	√	√	√				
	The attribute of *ExtCtrl* and *ExtParOut* has been changed from *retain* to *by_ref*. To access these parameters' properties, connection must be done using a variable, instead of directly using function block. Otherwise, errors will be prompted in Control Builder.	√	√	√	√				
	If FeedForward is used, note that the calculation for FeedForward has been changed to *FeedForward = Par FeedFwd* It was previously calculated as *FeedForward = Par FeedFwd /(MV Max - MV Min)* In addition, there is now no limit on the FeedForward value.	√	√	√					
PID01	*BalRef* value will passed to Output and limited only by the output range, not by Output Limit. Review the value.	√	√	√	√	√	√	√	√
	If derivative control *D* is used, you will need to modify the existing *TD* value to the following: *New TD = Old TD/ Gain*	√	√	√	√	√			
	HotInit parameter has been removed.	√	√	√	√				
	The attribute of *ExtCtrl* and *ExtParOut* has been changed from *retain* to *by_ref*. To access these parameters' properties, connection must be done using a variable, instead of directly using function block. Otherwise, errors will be prompted in Control Builder.	√	√	√	√				

<table>
<tr><th rowspan="2">Object Types</th><th rowspan="2">Special Notes</th><th colspan="8">Existing PPLib version</th></tr>
<tr><th>1.x-x
3.x-x
4.0-x</th><th>5.0-0</th><th>5.0-1</th><th>5.0-2
5.0-3</th><th>5.1</th><th>5.2
5.2h1</th><th>5.2r1
5.2r2</th><th>5.3</th></tr>
<tr><td>PID01</td><td>Tuning parameter may need to be updated depending on the PID01 output range and its limitation. <i>PO range</i> refers to the range of output defined in the Interaction Window as <i>PO Min</i> and <i>PO Max</i>. <i>Out LL</i> and <i>Out HL</i> refer to the output limitation defined in the Interaction Window or <i>EOLL</i> and <i>EOHL</i> in the function block if <i>EOLIM</i> is used.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Upgrading from 5.0-1
<table>
<tr><th>PO range</th><th>Out L</th><th>Out H</th><th>New Gain</th><th>New TI</th></tr>
<tr><td>0-100</td><td>= 0</td><td>= 100</td><td>no change</td><td>no change</td></tr>
<tr><td>0-100</td><td>&gt; 0</td><td>&lt; 100</td><td>$\frac{\text{Old Gain * (Out HL - Out LL)}}{100}$</td><td>$\frac{\text{Old TI * (Out HL - Out LL)}}{100}$</td></tr>
<tr><td>≠ 0-100</td><td>= PO Min</td><td>= PO Max</td><td>no change</td><td>$\frac{\text{Old TI * (PO Max - PO Min)}}{100}$</td></tr>
<tr><td>≠ 0-100</td><td>&gt; PO Min</td><td>&lt; PO Max</td><td>$\frac{\text{Old Gain * (Out HL - Out LL)}}{\text{(PO Max - PO Min)}}$</td><td>$\frac{\text{Old TI * (Out HL - Out LL)}}{100}$</td></tr>
</table>
Upgrading from 5.0-1 rollup 1
<table>
<tr><th>PO range</th><th>Out L</th><th>Out H</th><th>New Gain</th><th>New TI</th></tr>
<tr><td>0-100</td><td>= 0</td><td>= 100</td><td>no change</td><td>no change</td></tr>
<tr><td>0-100</td><td>&gt; 0</td><td>&lt; 100</td><td>no change</td><td>no change</td></tr>
<tr><td>≠ 0-100</td><td>= PO Min</td><td>= PO Max</td><td>no change</td><td>$\frac{\text{Old TI * (PO Max - PO Min)}}{100}$</td></tr>
<tr><td>≠ 0-100</td><td>&gt; PO Min</td><td>&lt; PO Max</td><td>no change</td><td>$\frac{\text{Old TI * (PO Max - PO Min)}}{100}$</td></tr>
</table></td><td></td><td></td><td>√</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Upgrading from 5.0-0 or older
<table>
<tr><th>PO range</th><th>Out L</th><th>Out H</th><th>New Gain</th><th>New TI</th></tr>
<tr><td>0-100</td><td>= 0</td><td>= 100</td><td>no change</td><td>no change</td></tr>
<tr><td>0-100</td><td>&gt; 0</td><td>&lt; 100</td><td>$\frac{\text{Old Gain * (Out HL - Out LL)}}{100}$</td><td>$\frac{\text{Old TI * (Out HL - Out LL)}}{100}$</td></tr>
<tr><td>≠ 0-100</td><td>= PO Min</td><td>= PO Max</td><td>no change</td><td>$\frac{\text{Old TI * (PO Max - PO Min)}}{100}$</td></tr>
<tr><td>≠ 0-100</td><td>&gt; PO Min</td><td>&lt; PO Max</td><td>$\frac{\text{Old Gain * (Out HL - Out LL)}}{\text{(PO Max - PO Min)}}$</td><td>$\frac{\text{Old TI * (Out HL - Out LL)}}{100}$</td></tr>
</table></td><td>√</td><td>√</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>