WEXØE

Find us here and get inspired

Studio 5000® Tips & Trix

Dagens presentatör:

Jari Turja,
Commercial Engineer, Wexoe AB

WEXOE

Find us here and get inspired

in ▶

Information

Stäng gärna av er mikrofon.

Ställ dessa i chattfönstret.

Besvaras efter Webinaret och skickas ut via e-post.

Länkar till Webinar kommer att skickas ut via e-post. Dessa kommer även finnas tillgängliga på vår hemsida, www.wexoe.se

WEXØE

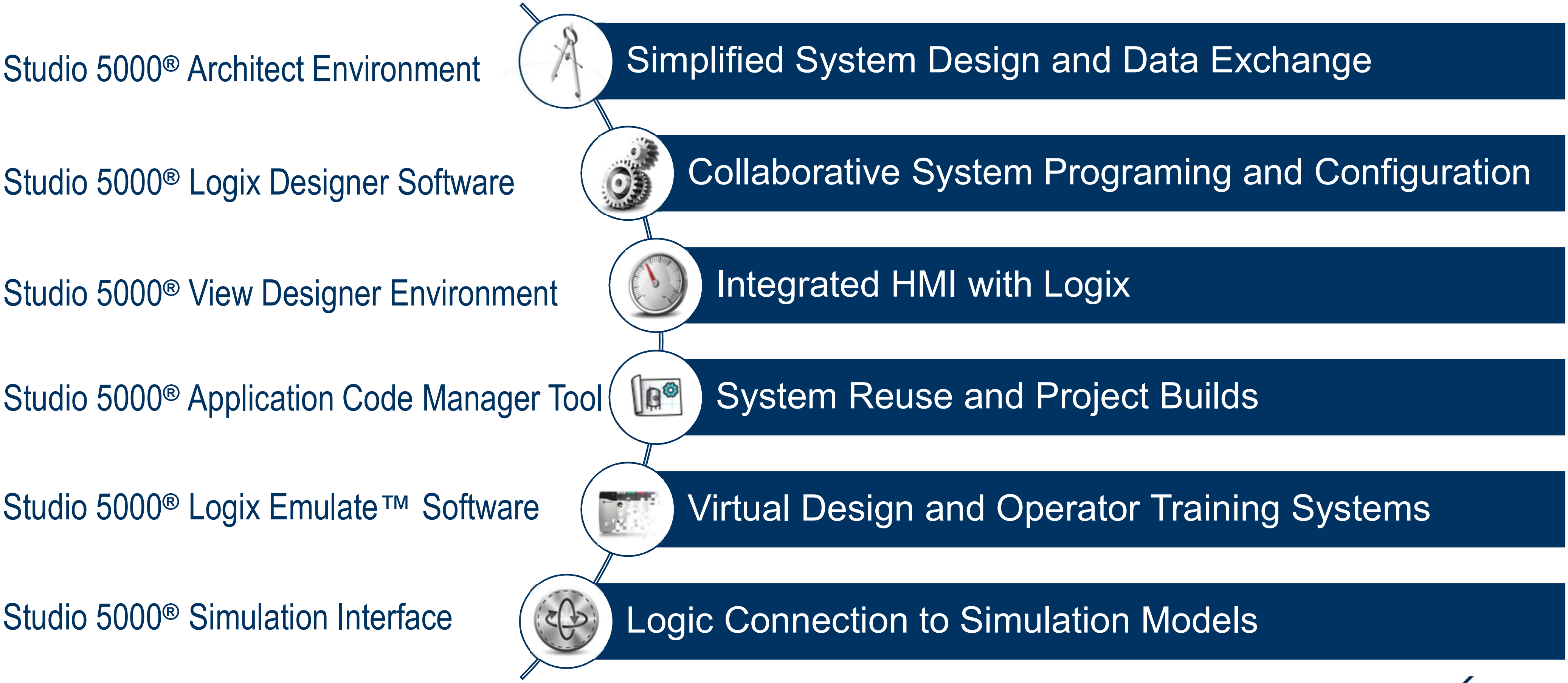
Find us here and get inspired

Dagens agenda

- Studio 5000® Design Environment
- Features of Studio 5000®
- Studio 5000® Logix Designer
- What happens next?

Studio 5000® Design Environment

Rockwell Software Studio 5000



Studio 5000® Architect Environment — Simplified System Design and Data Exchange

Studio 5000® Logix Designer Software — Collaborative System Programing and Configuration

Studio 5000® View Designer Environment — Integrated HMI with Logix

Studio 5000® Application Code Manager Tool — System Reuse and Project Builds

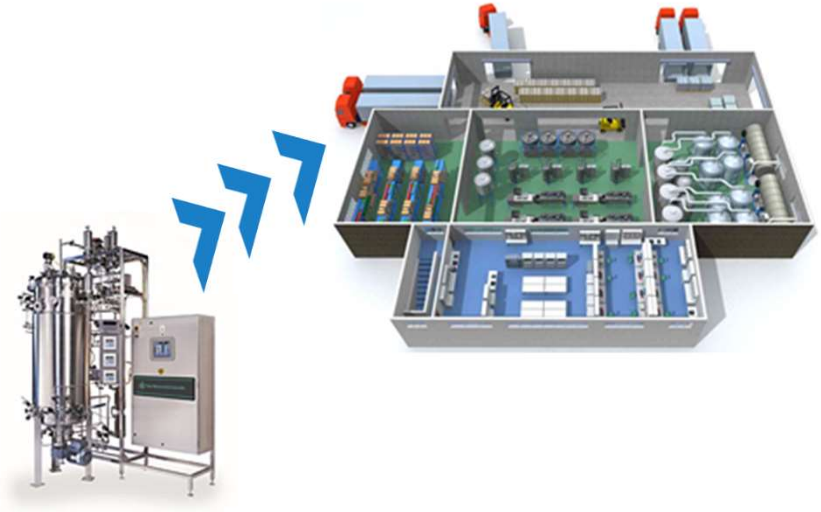
Studio 5000® Logix Emulate™ Software — Virtual Design and Operator Training Systems

Studio 5000® Simulation Interface — Logic Connection to Simulation Models

WEXOE

Features of Studio 5000®

Scalability



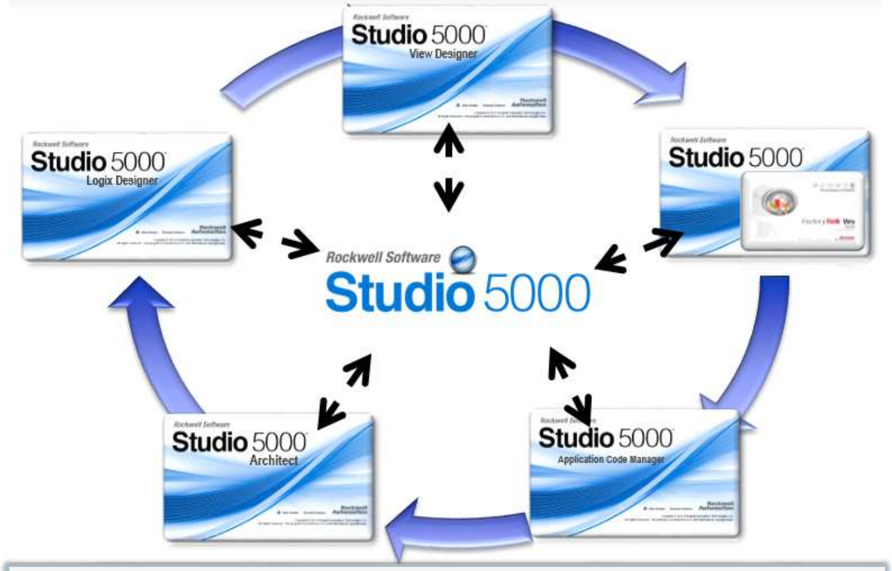
Designed for any system from device to enterprise

Multi-Disciplined

A single design environment that can be used for many disciplines and industries

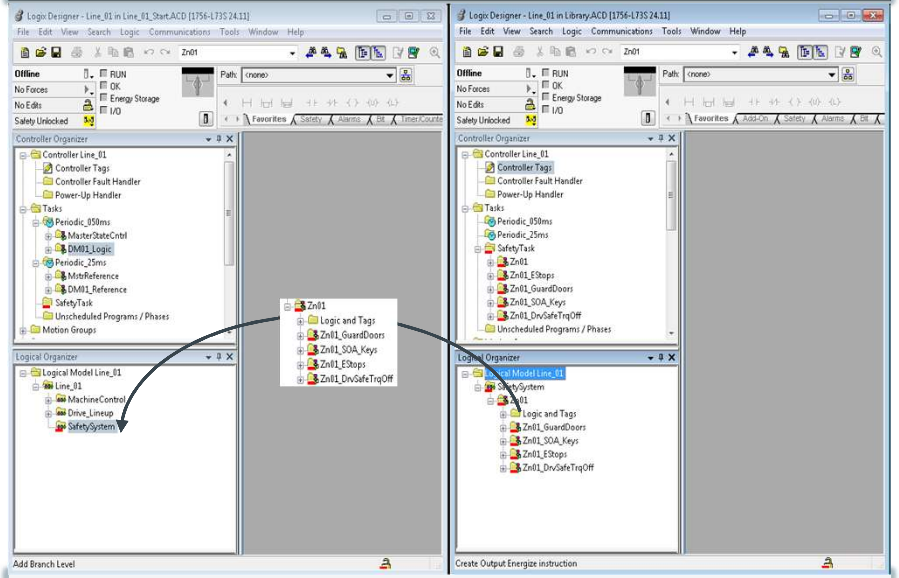
Features of Studio 5000®

Common Infrastructure


Studio 5000 View Designer
Studio 5000 Logix Designer
Studio 5000
Rockwell Software Studio 5000
Studio 5000 Architect
Studio 5000 Application Code Manager

Common launch point for all the applications within Studio 5000

Intuitive Workflow



Simplified wizards, drag and drop configurations, and easier troubleshooting

Logix Designer

Studio *5000®* ***Logix Designer Software*** *Program and Design Securely and Efficiently*

Studio 5000

Studio 5000® Logix Designer

ARCHITECTURE & SOFTWARE

CONTROL PRODUCTS & SOLUTIONS

CONTROL SYSTEMS

VISUALIZATION & SOFTWARE

INTELLIGENT MOTOR CONTROL

CONTROL COMPONENTS

INDUSTRIAL NETWORKS

SAFETY

VALUE-ADD SERVICE

ENGINEERED SYSTEMS & SOLUTIONS

…WHERE TECHNOLOGY DIFFERENTATION AND DOMAIN EXPERTISE MATTER!

WEXØE

Studio 5000® Logix Designer

One place to configure, program and maintain the entire Logix 5000 family of controllers products

Configure

- Configure all Logix 5000 controllers and related peripheral devices
- Automatically create tags
- Protect the design and execution of your Logix content

Program

- Multi-Language IEC 1131-3, multi-discipline, Multi-user, intuitive programing environment
- Modular programming features

Maintain

- Intuitive environment to diagnose and maintain system uptime
- Logical views and simplified tools

Collaborate

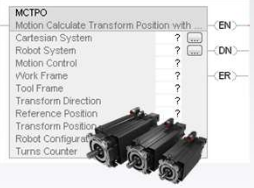
- Simplification of concurrent engineering workflows
- Compare and merge features for multiple user collaboration

Studio 5000® Logix Designer: Configure, program and maintain your application

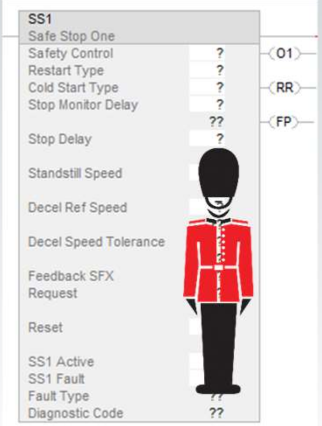
Reduce engineering time, Faster time to market

Integrated Motion, Safety and HMI programming

- ✓ View designer for HMI application design
- ✓ Create application tags only once
- ✓ 73 native motion instructions and 36 native safety instructions



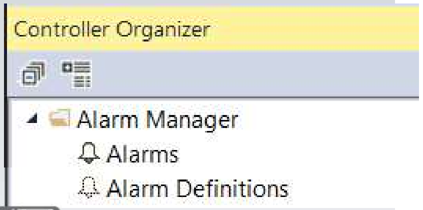
MCTPO
Motion Calculate Transform Position with ...
Cartesian System ?
Robot System ?
Motion Control ?
Work Frame ?
Tool Frame ?
Transform Direction ?
Reference Position ?
Transform Position
Robot Configurat
Turns Counter



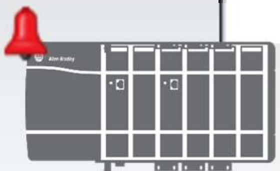
SS1
Safe Stop One
Safety Control ?
Restart Type ?
Cold Start Type ?
Stop Monitor Delay ?
??
Stop Delay ?
Standstill Speed
Decel Ref Speed
Decel Speed Tolerance
Feedback SFX
Request
Reset
SS1 Active
SS1 Fault
Fault Type ??
Diagnostic Code ??

Create alarms and events with one click

- ✓ Add alarms to tags or structures without using instructions or need to program
- ✓ Alarms send to FT Alarms and events automatically
- ✓ Alarm manager to configure all alarms



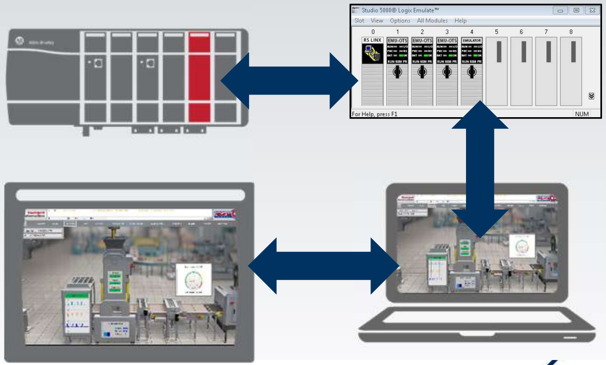
Controller Organizer
Alarm Manager
Alarms
Alarm Definitions



Logix Tag-Based Alarms

Emulate HMI and controller

- ✓ Simulate the controller, IO and HMI without hardware in a safe, virtual environment
- ✓ Program execution time scaling possible
- ✓ Design, test and validate application code before machine is deployed

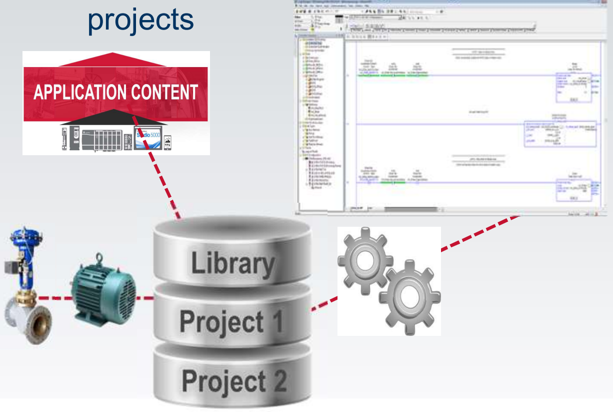


Studio 5000® Logix Designer: Configure, program and maintain your application

Reduce engineering time, Faster time to market

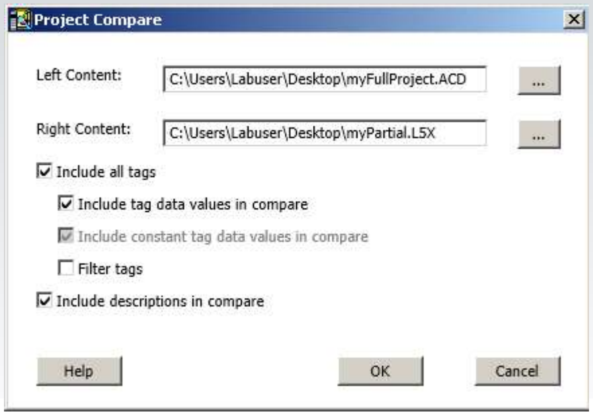
Modular design

- ✓ Create your own library fast and easy
- ✓ Maximized flexibility
- ✓ Decrease engineering time
- ✓ Enable or disable devices in the project with a few clicks
- ✓ Application Code Manager to maximize reuse and quickly build projects



Collaboration

- ✓ Easy and fast merging of code created by multiple programmers, even with running controller
- ✓ Compare code on a granular level like AOI-to-AOI, UDT-to-UDT but also One-to-One and One-to-Many
- ✓ Compare tags, tag properties and tag data values



Project Compare

Left Content: C:\Users\Labuser\Desktop\myFullProject.ACD

Right Content: C:\Users\Labuser\Desktop\myPartial.L5X

☑ Include all tags

☑ Include tag data values in compare

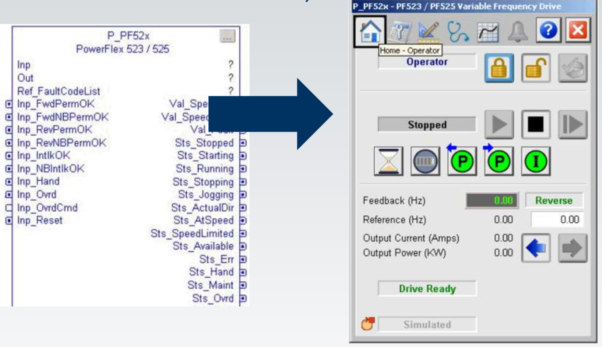
☑ Include constant tag data values in compare

□ Filter tags

☑ Include descriptions in compare

Add-on profiles, add-on instructions and faceplates

- ✓ Each device has its own AOP to configure the devices. Tag structure is created automatically
- ✓ AOI build and tested by Rockwell
- ✓ Faceplates have a similar look and feel. Learn once, use many



Feedback (Hz) 0.00 Reverse

Reference (Hz) 0.00 0.00

Output Current (Amps) 0.00

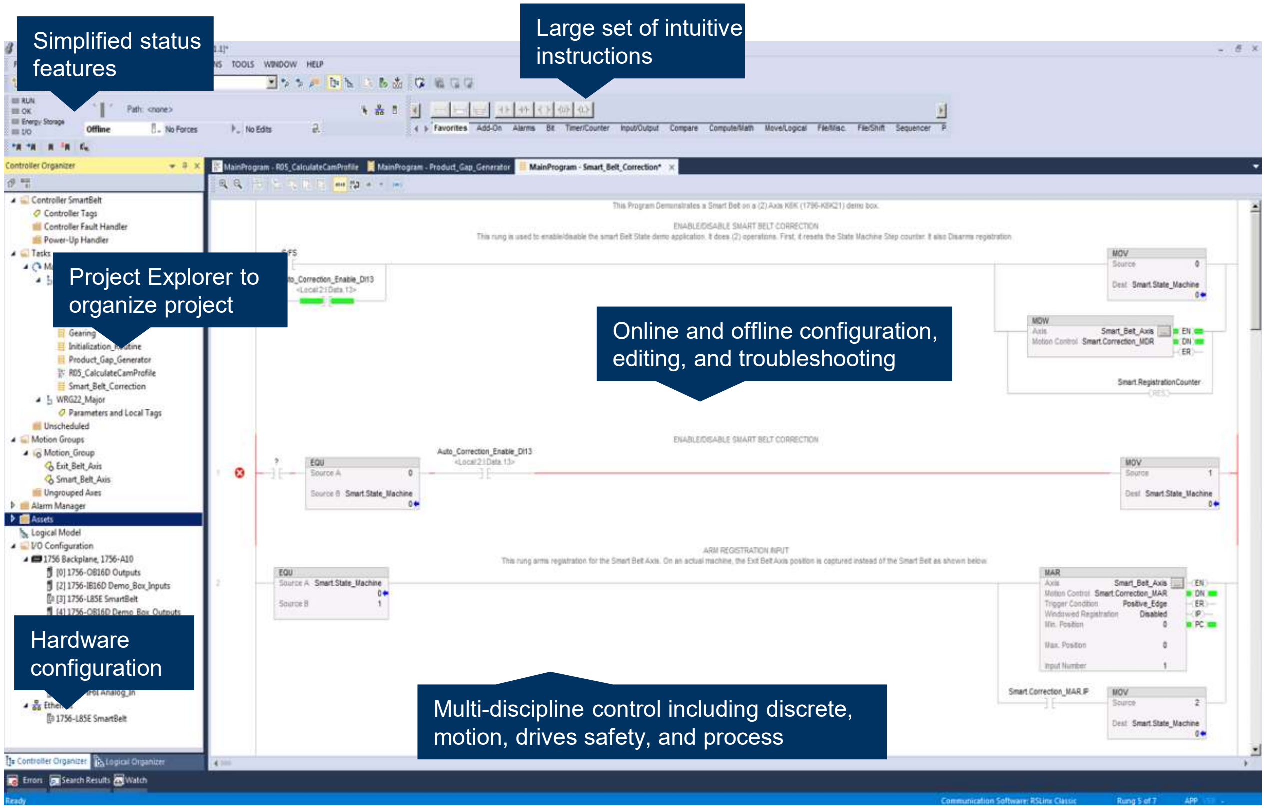
Output Power (KW) 0.00

Building Scalable Solutions

Studio 5000® Logix Designer Software

Studio 5000

Studio 5000® Logix Designer



Simplified status features

Large set of intuitive instructions

Project Explorer to organize project

Online and offline configuration, editing, and troubleshooting

Hardware configuration

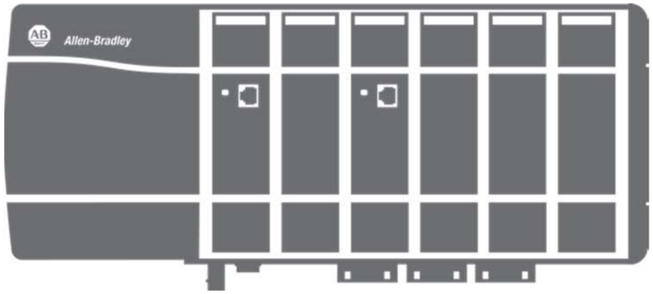
Multi-discipline control including discrete, motion, drives safety, and process

ONE DESIGN ENVIRONMENT FOR ALL DISCIPLINES

- Supports comprehensive design and maintenance of the automation control system

COLLABORATIVE DEVELOPMENT

- Focused on delivering automation productivity, ease-of-use, and simplified workflows

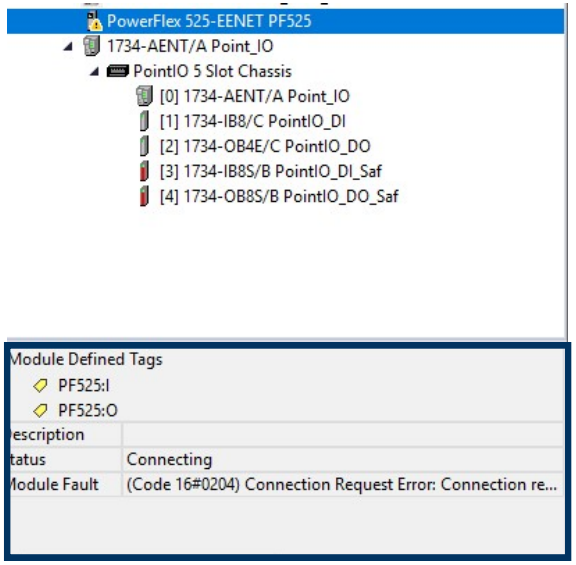


SINGLE PROGRAMMING ENVIRONMENT FOR ALL LOGIX CONTROLLERS

WEXOE

Studio 5000® Logix Designer

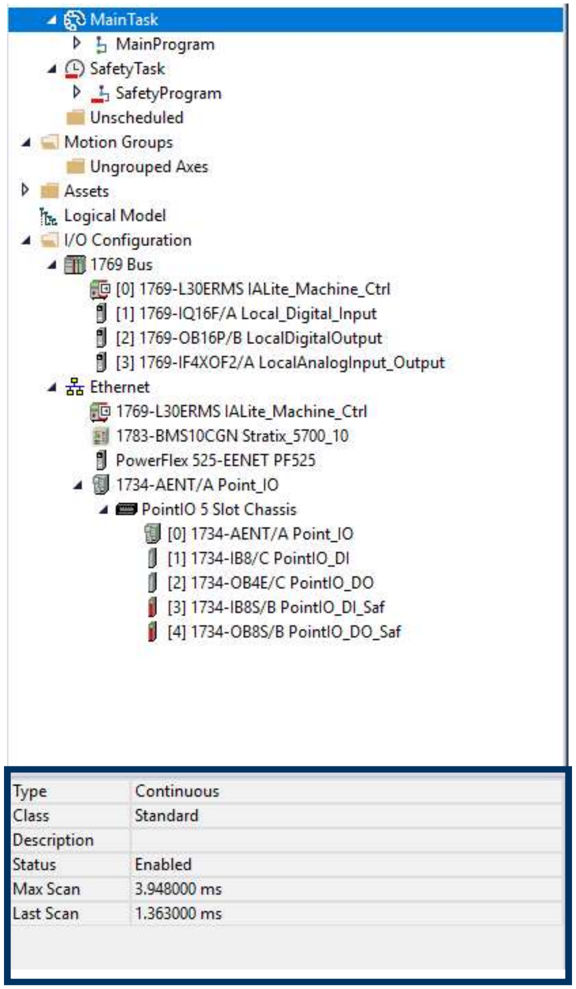
Status information



- PowerFlex 525-EENET PF525
  - 1734-AENT/A Point_IO
    - PointIO 5 Slot Chassis
      - [0] 1734-AENT/A Point_IO
      - [1] 1734-IB8/C PointIO_DI
      - [2] 1734-OB4E/C PointIO_DO
      - [3] 1734-IB8S/B PointIO_DI_Saf
      - [4] 1734-OB8S/B PointIO_DO_Saf

Module Defined Tags
- PF525:I
- PF525:O

escription	
tatus	Connecting
Module Fault	(Code 16#0204) Connection Request Error: Connection re...



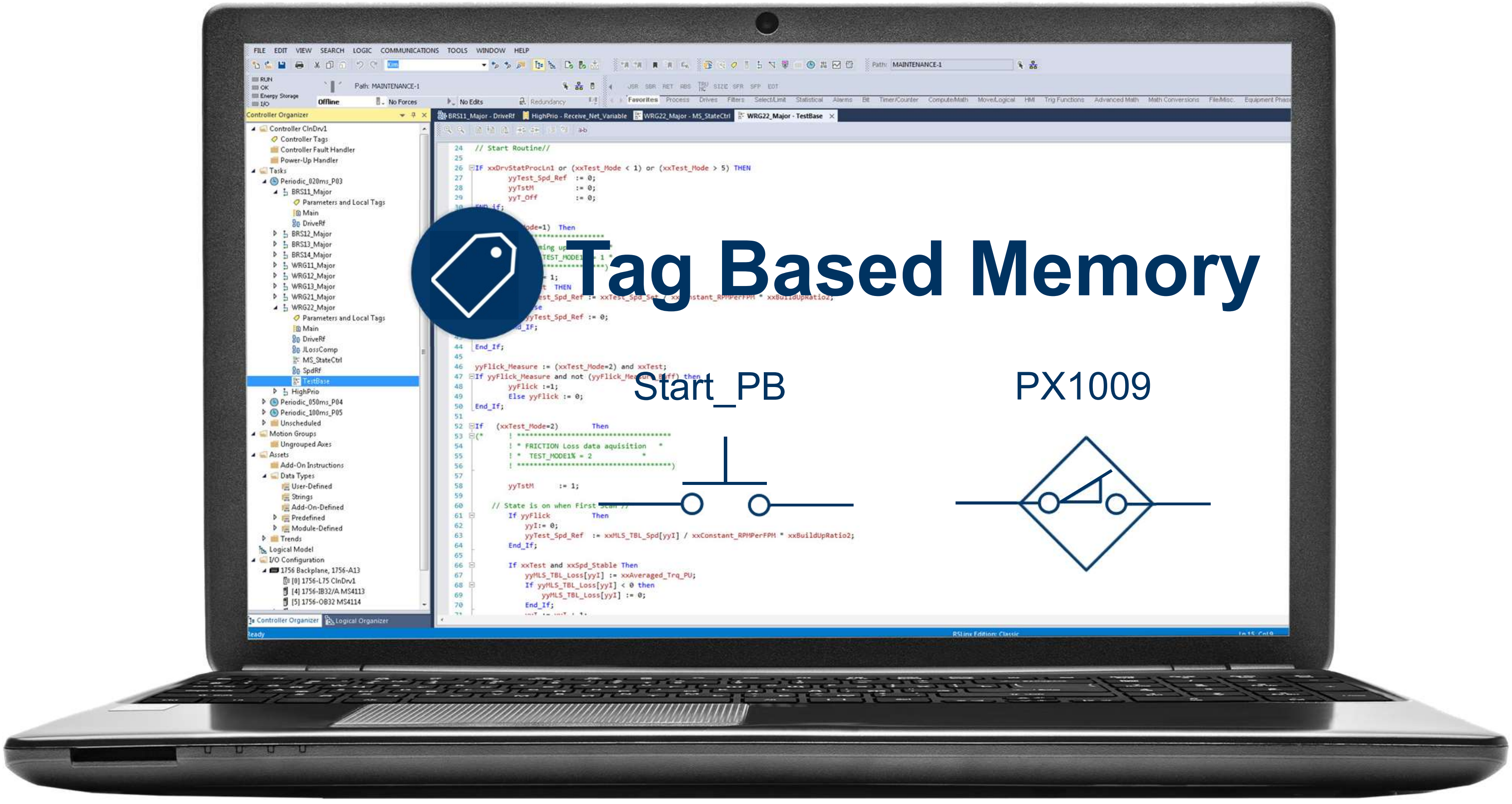
- MainTask
  - MainProgram
- SafetyTask
  - SafetyProgram
- Unscheduled
- Motion Groups
  - Ungrouped Axes
- Assets
- Logical Model
- I/O Configuration
  - 1769 Bus
    - [0] 1769-L30ERMS IALite_Machine_Ctrl
    - [1] 1769-IQ16F/A Local_Digital_Input
    - [2] 1769-OB16P/B LocalDigitalOutput
    - [3] 1769-IF4XOF2/A LocalAnalogInput_Output
  - Ethernet
    - 1769-L30ERMS IALite_Machine_Ctrl
    - 1783-BMS10CGN Stratix_5700_10
    - PowerFlex 525-EENET PF525
    - 1734-AENT/A Point_IO
      - PointIO 5 Slot Chassis
        - [0] 1734-AENT/A Point_IO
        - [1] 1734-IB8/C PointIO_DI
        - [2] 1734-OB4E/C PointIO_DO
        - [3] 1734-IB8S/B PointIO_DI_Saf
        - [4] 1734-OB8S/B PointIO_DO_Saf

Type	Continuous
Class	Standard
Description	
Status	Enabled
Max Scan	3.948000 ms
Last Scan	1.363000 ms

✓ Status information

- ✓ Hardware
- ✓ Programs
- ✓ etc…

Studio 5000® Logix Designer



```
// Start Routine//

IF xxDrvStatProcLn1 or (xxTest_Mode < 1) or (xxTest_Mode > 5) THEN
        yyTest_Spd_Ref  := 0;
        yyTstM          := 0;
        yyT_Off         := 0;
END_if;

    ode=1)  Then
    *****************
    ming up
    TEST_MODE1 = 1 *
    **************)
    = 1;
    t  THEN
    est_Spd_Ref := xxTest_Spd_Set / xx stant_RPMPerFPM * xxBuildUpRatio2;
    se
    yyTest_Spd_Ref := 0;
    d_IF;

End_If;

yyFlick_Measure := (xxTest_Mode=2) and xxTest;
If yyFlick_Measure and not (yyFlick_Measure_Buff) then
        yyFlick :=1;
        Else yyFlick := 0;
End_If;

If   (xxTest_Mode=2)        Then
(*      ! ************************************
        ! * FRICTION Loss data aquisition   *
        ! *  TEST_MODE1% = 2               *
        ! ************************************)

        yyTstM     := 1;

    // State is on when First Scan //
        If yyFlick          Then
            yyI:= 0;
            yyTest_Spd_Ref  := xxMLS_TBL_Spd[yyI] / xxConstant_RPMPerFPM * xxBuildUpRatio2;
        End_If;

        If xxTest and xxSpd_Stable Then
            yyMLS_TBL_Loss[yyI] := xxAveraged_Trq_PU;
            If yyMLS_TBL_Loss[yyI] < 0 then
                yyMLS_TBL_Loss[yyI] := 0;
            End_If;
```

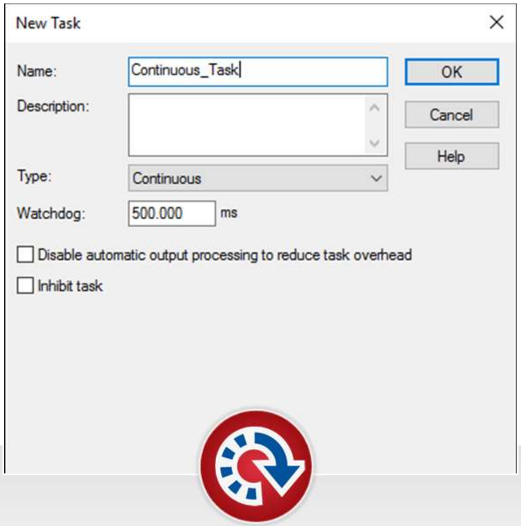
Tag Based Memory

Start_PB

PX1009

Studio 5000® **Logix Designer**

Flexible Task Execution



New Task

Name: Continuous_Task

Description:

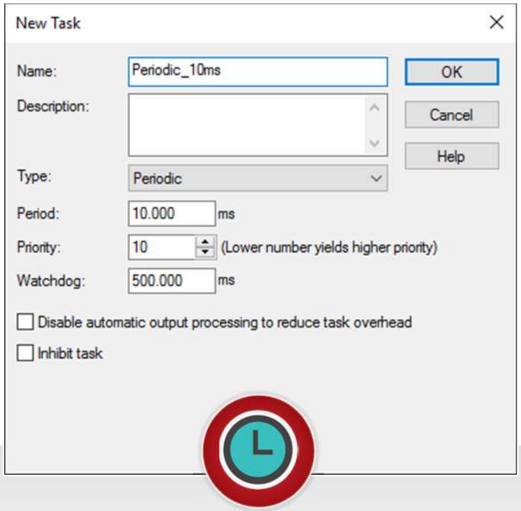
Type: Continuous

Watchdog: 500.000 ms

□ Disable automatic output processing to reduce task overhead

□ Inhibit task

Continuous Tasks



New Task

Name: Periodic_10ms

Description:

Type: Periodic

Period: 10.000 ms

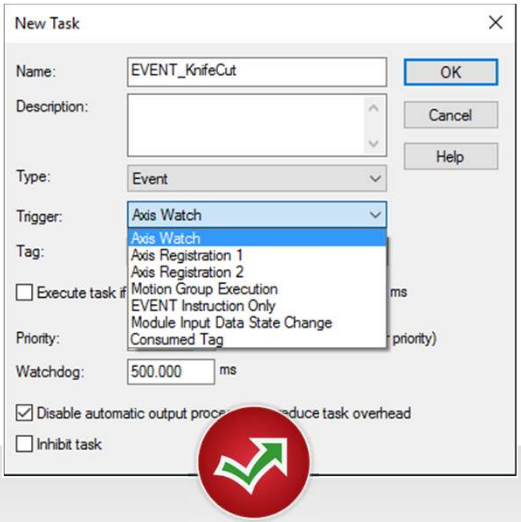
Priority: 10 (Lower number yields higher priority)

Watchdog: 500.000 ms

□ Disable automatic output processing to reduce task overhead

□ Inhibit task

Periodic Tasks



New Task

Name: EVENT_KnifeCut

Description:

Type: Event

Trigger: Axis Watch

Tag:

□ Execute task if ms

Priority: priority)

Watchdog: 500.000 ms

☑ Disable automatic output proce duce task overhead

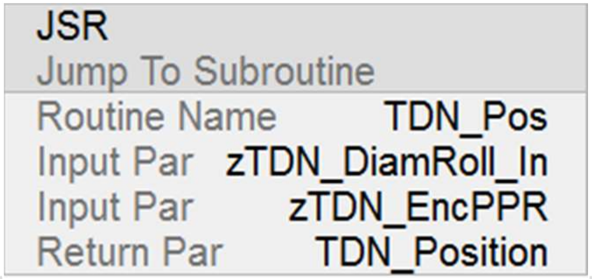
□ Inhibit task

Event Tasks

Studio 5000® Logix Designer

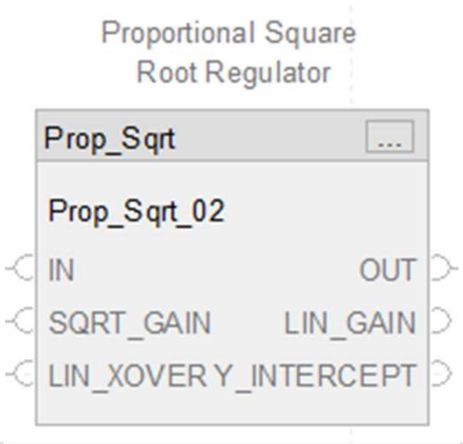
Studio 5000® Logix Designer

Modular Containers



JSR
Jump To Subroutine
Routine Name TDN_Pos
Input Par zTDN_DiamRoll_In
Input Par zTDN_EncPPR
Return Par TDN_Position

Subroutines

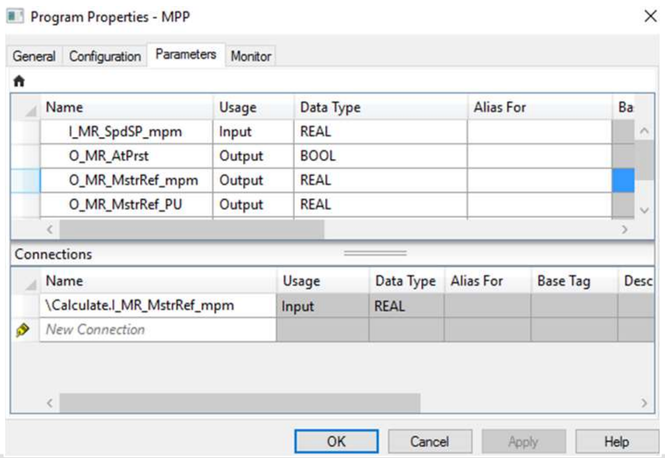


Proportional Square
Root Regulator

Prop_Sqrt

Prop_Sqrt_02

Add-on Instructions



Program Properties - MPP

Name	Usage	Data Type	Alias For	Ba
I_MR_SpdSP_mpm	Input	REAL		
O_MR_AtPrst	Output	BOOL		
O_MR_MstrRef_mpm	Output	REAL		
O_MR_MstrRef_PU	Output	REAL		

Connections

Name	Usage	Data Type	Alias For	Base Tag	Desc
\Calculate.I_MR_MstrRef_mpm	Input	REAL			
New Connection					

Programs

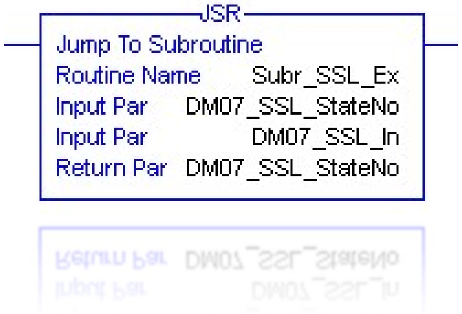
MAKE CHANGES EASIER & UNDERSTAND HOW THEY WILL AFFECT OVERALL DESIGN

Studio 5000® Logix Designer

Modular Containers - Subroutines

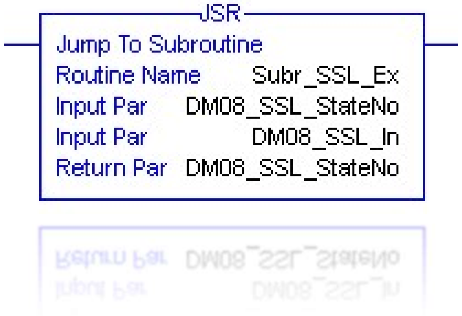
- ✓Logic should reasonable fit into one routine
- ✓Great for PLC-5® / SLC™ migration
- ✓Passes all parameters by value
- ✓Requires highest overhead
- ✓NOT globally accessible
- ✓Can be difficult to troubleshoot, remember to create logic to isolate each instance

Section State Logic
Subroutine
(14 States,
(Structured Text)



```
JSR
Jump To Subroutine
Routine Name      Subr_SSL_Ex
Input Par    DM07_SSL_StateNo
Input Par         DM07_SSL_In
Return Par   DM07_SSL_StateNo
```

Section State Logic
Subroutine
(14 States,
(Structured Text)

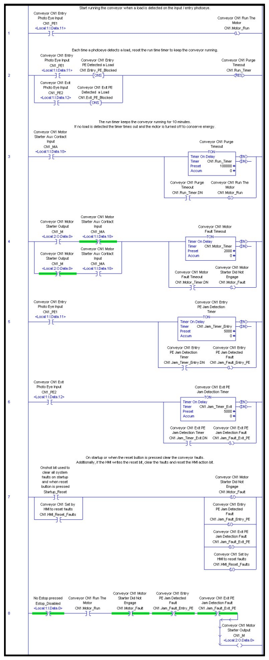


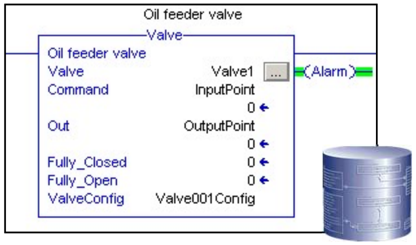
```
JSR
Jump To Subroutine
Routine Name      Subr_SSL_Ex
Input Par    DM08_SSL_StateNo
Input Par         DM08_SSL_In
Return Par   DM08_SSL_StateNo
```

Studio 5000® Logix Designer

Modular Containers - Add-On-instructions (AOI)

- ✓Logic should reasonably fit into one routine
- ✓True “Definition”
- ✓Globally accessible
- ✓Built in signatures and revision history
- ✓Easy to view each instance
- ✓Supports LD, STX and FBD
- ✓Requires lowest overhead
- ✓Intended to encapsulate commonly used functions or device control
- ✓Does not support online edits

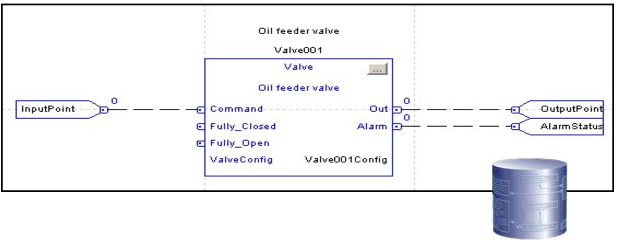




Oil feeder valve

Valve

Oil feeder valve	
Valve	Valve1 ...
Command	InputPoint
	0 ←
Out	OutputPoint
	0 ←
Fully_Closed	0 ←
Fully_Open	0 ←
ValveConfig	Valve001Config



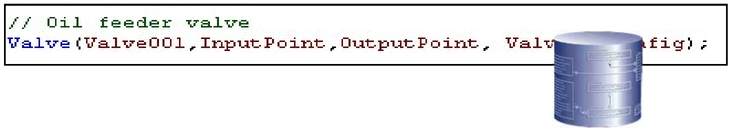
Oil feeder valve

Valve001

Valve ...

Oil feeder valve

Command	Out 0
Fully_Closed	Alarm 0
Fully_Open	
ValveConfig	Valve001Config

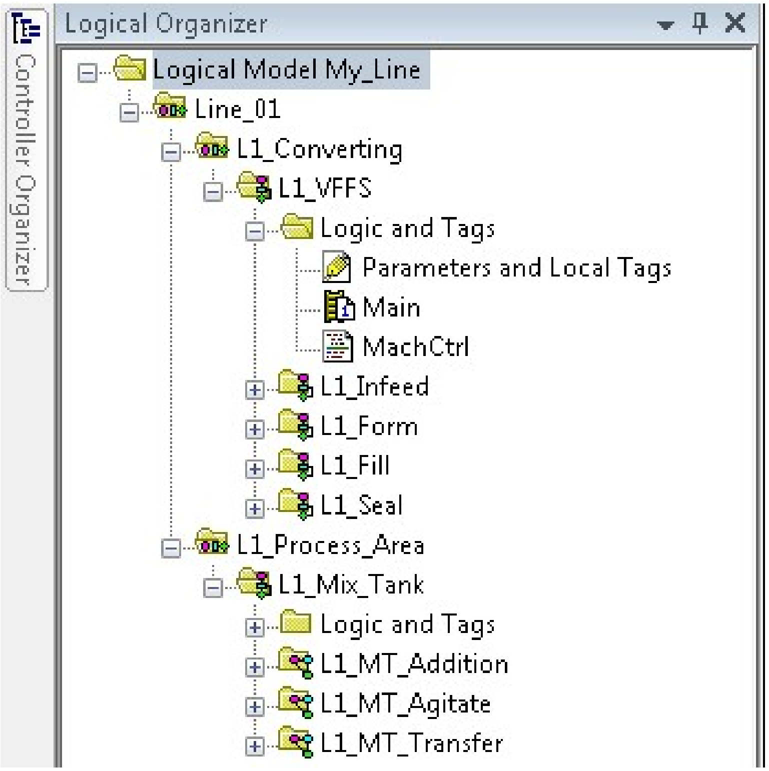


```
// Oil feeder valve
Valve(Valve001,InputPoint,OutputPoint, Valv          fig);
```

Studio 5000® Logix Designer

Modular Containers - Programs with Parameters

- ✓Supports multiple routines and all four programming languages
- ✓Supports online edits
- ✓Parameters and local tags provide same level of encapsulation as AOIs
- ✓Best option for large, complex code modules
- ✓Optimized with Partial Import Online and Logical Organizer
- ✓Not a true “Definition”



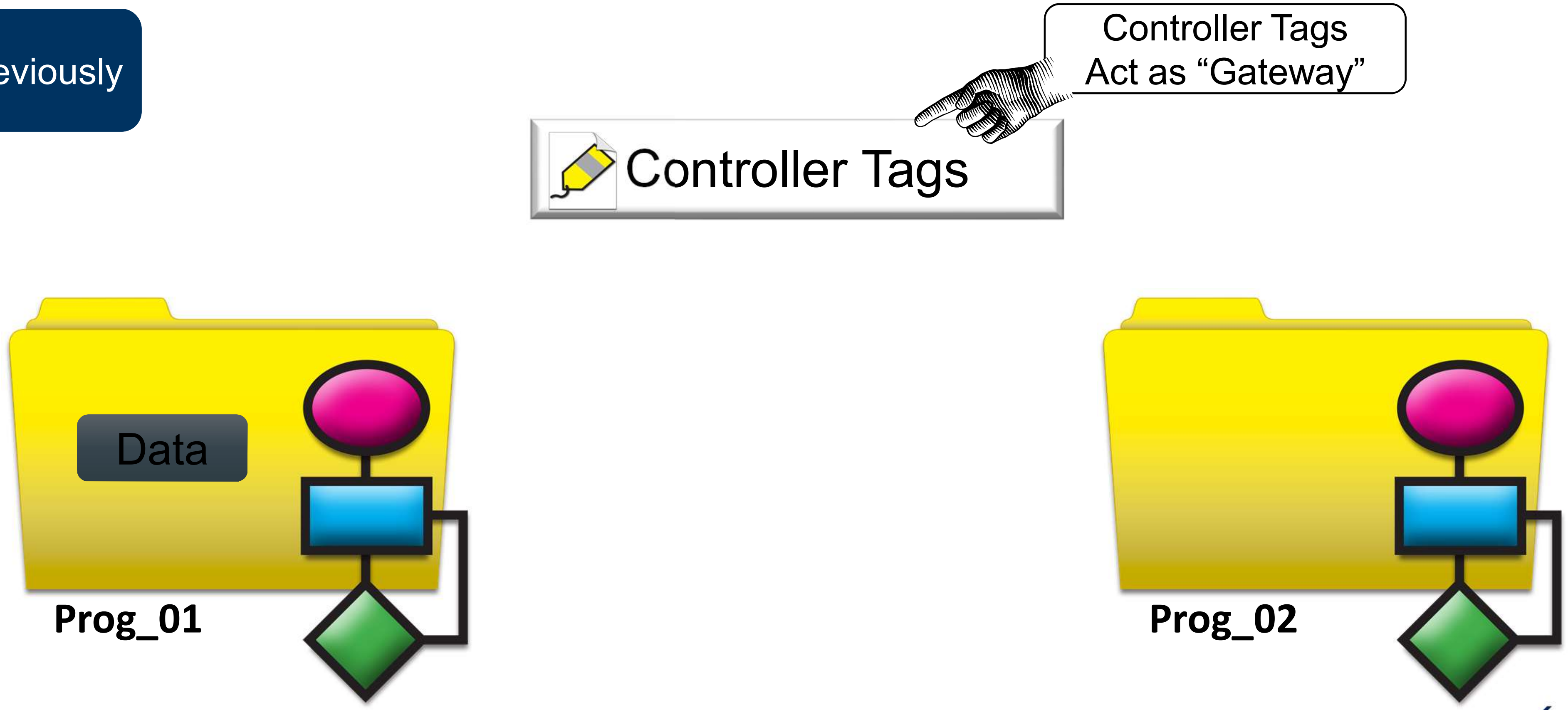
Logical Organizer

```
Logical Model My_Line
  Line_01
    L1_Converting
      L1_VFFS
        Logic and Tags
          Parameters and Local Tags
          Main
          MachCtrl
        L1_Infeed
        L1_Form
        L1_Fill
        L1_Seal
    L1_Process_Area
      L1_Mix_Tank
        Logic and Tags
        L1_MT_Addition
        L1_MT_Agitate
        L1_MT_Transfer
```

Studio 5000® **Logix Designer** - Programs with Parameters

Before - Programs to Program Communications

Previously

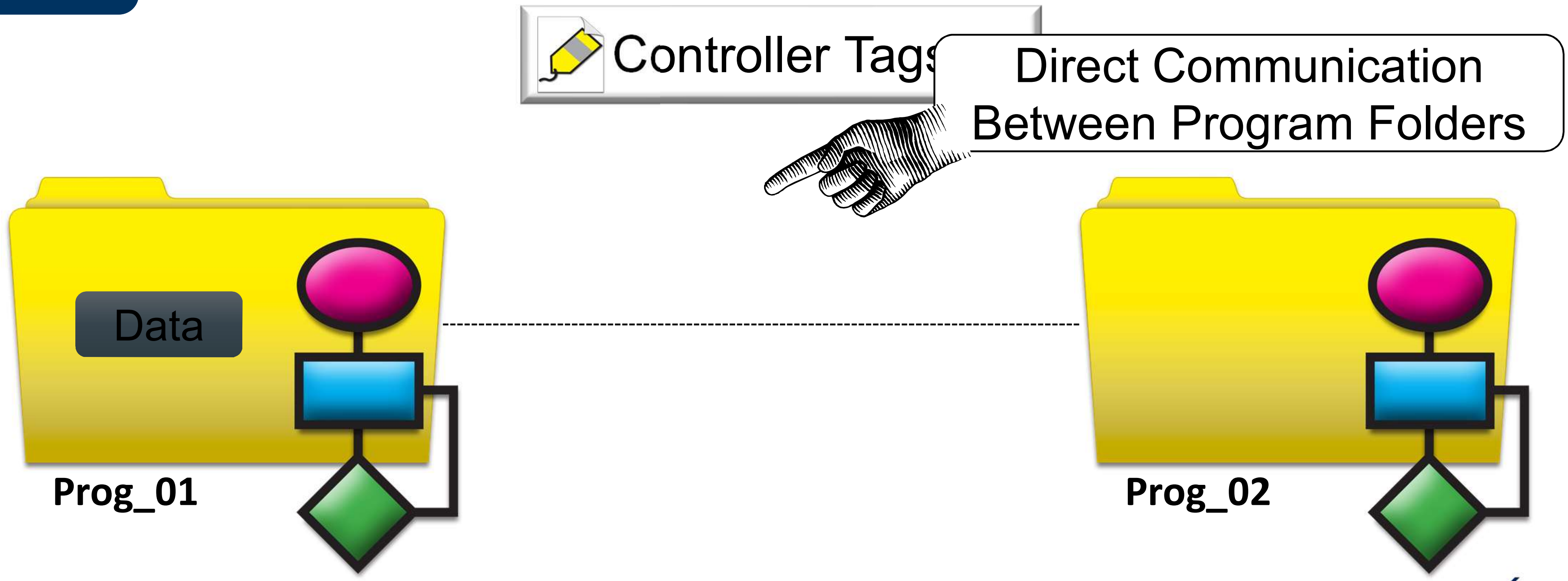


Controller Tags
Act as "Gateway"

Studio 5000® **Logix Designer** - Programs with Parameters

Direct Communication Between Programs

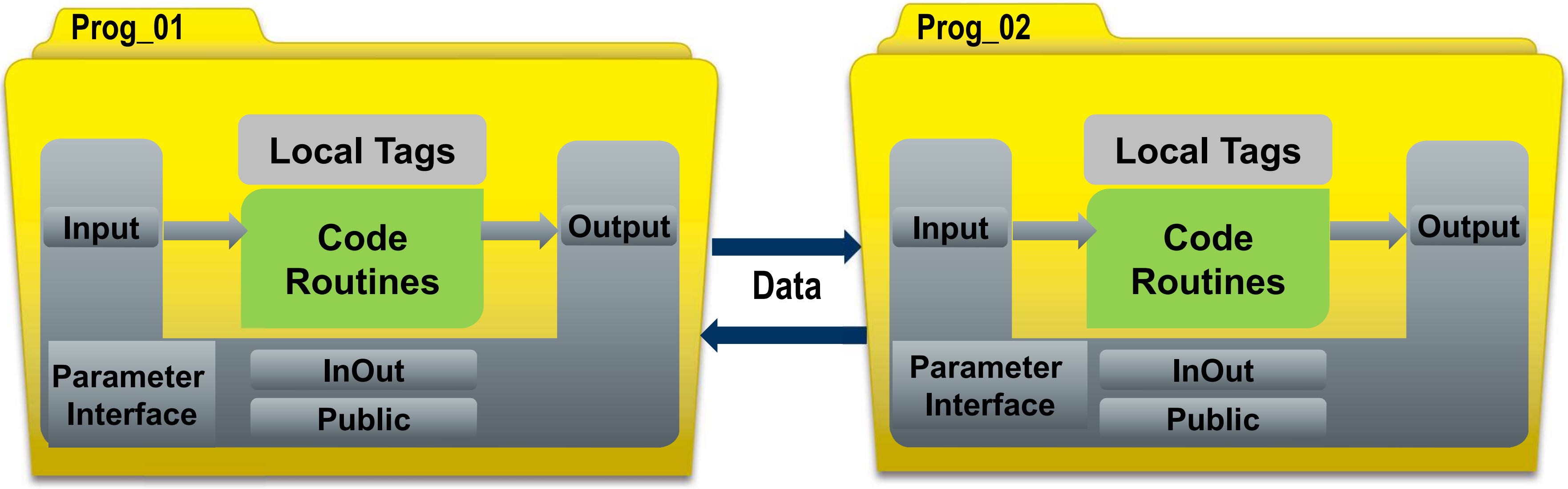
Now



Direct Communication Between Program Folders

Studio 5000® **Logix Designer** - Programs with Parameters

Direct Communication Between Programs



Why Add a Parameter Interface?

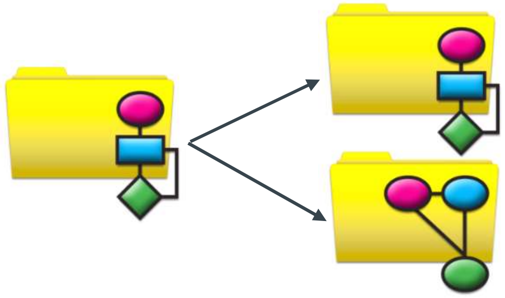
- ✓ Better understanding of interaction between programs
- ✓ Allow direct communication between programs
- ✓ Introduce a larger modular object in Logix Designer

WEXØE

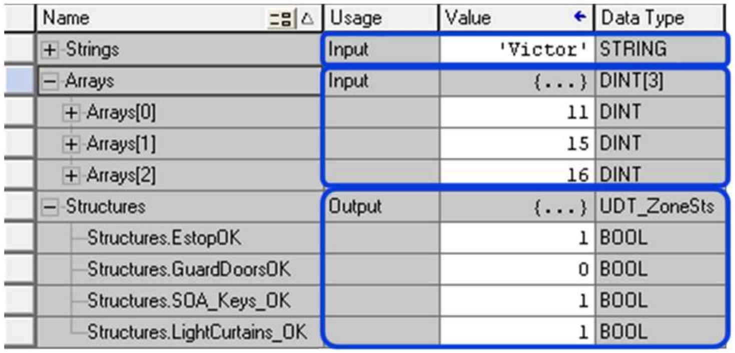
Studio 5000® Logix Designer

Program Parameters - Benefits

Interconnect Programs



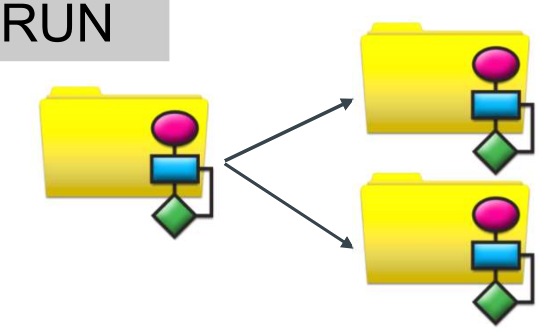
Strings, Arrays & Structures



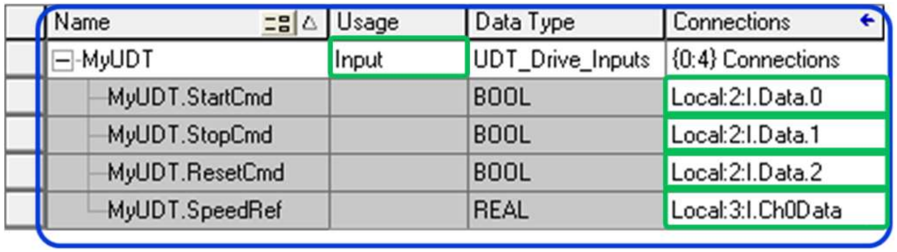
Name	Usage	Value	Data Type
+ Strings	Input	'Victor'	STRING
− Arrays	Input	{...}	DINT[3]
+ Arrays[0]		11	DINT
+ Arrays[1]		15	DINT
+ Arrays[2]		16	DINT
− Structures	Output	{...}	UDT_ZoneSts
Structures.EstopOK		1	BOOL
Structures.GuardDoorsOK		0	BOOL
Structures.SOA_Keys_OK		1	BOOL
Structures.LightCurtains_OK		1	BOOL

Online Editable

🟩 RUN



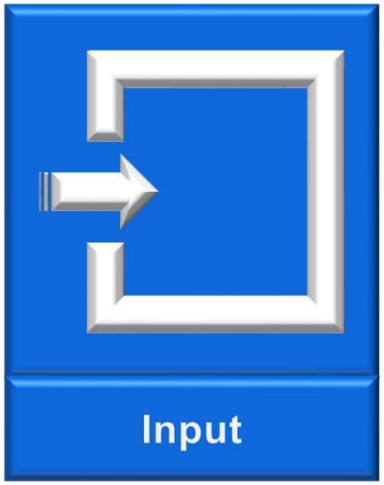
UDT Member Level Connections

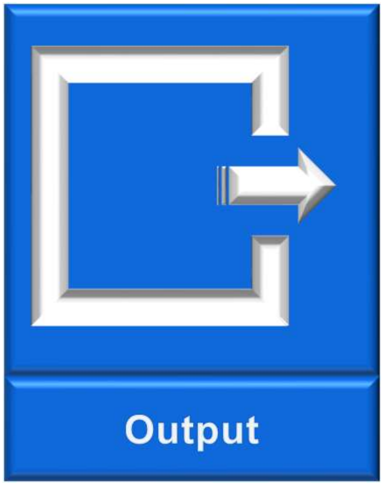


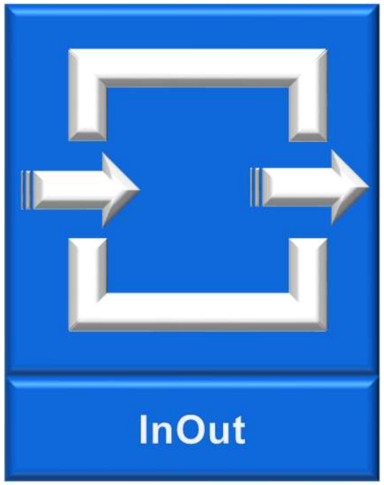
Name	Usage	Data Type	Connections
− MyUDT	Input	UDT_Drive_Inputs	{0:4} Connections
MyUDT.StartCmd		BOOL	Local:2:I.Data.0
MyUDT.StopCmd		BOOL	Local:2:I.Data.1
MyUDT.ResetCmd		BOOL	Local:2:I.Data.2
MyUDT.SpeedRef		REAL	Local:3:I.Ch0Data

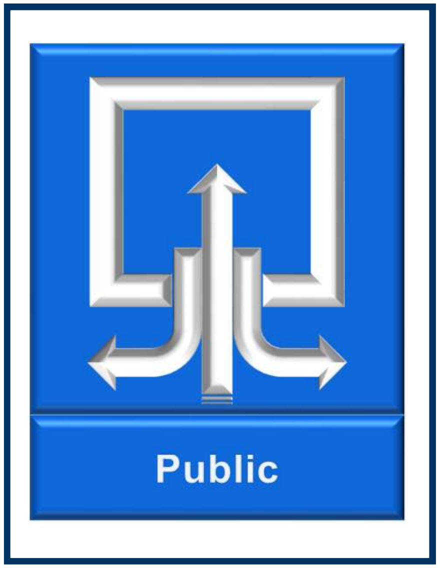
Studio 5000® **Logix Designer**

Public Parameters – Global Tag at a Program Level



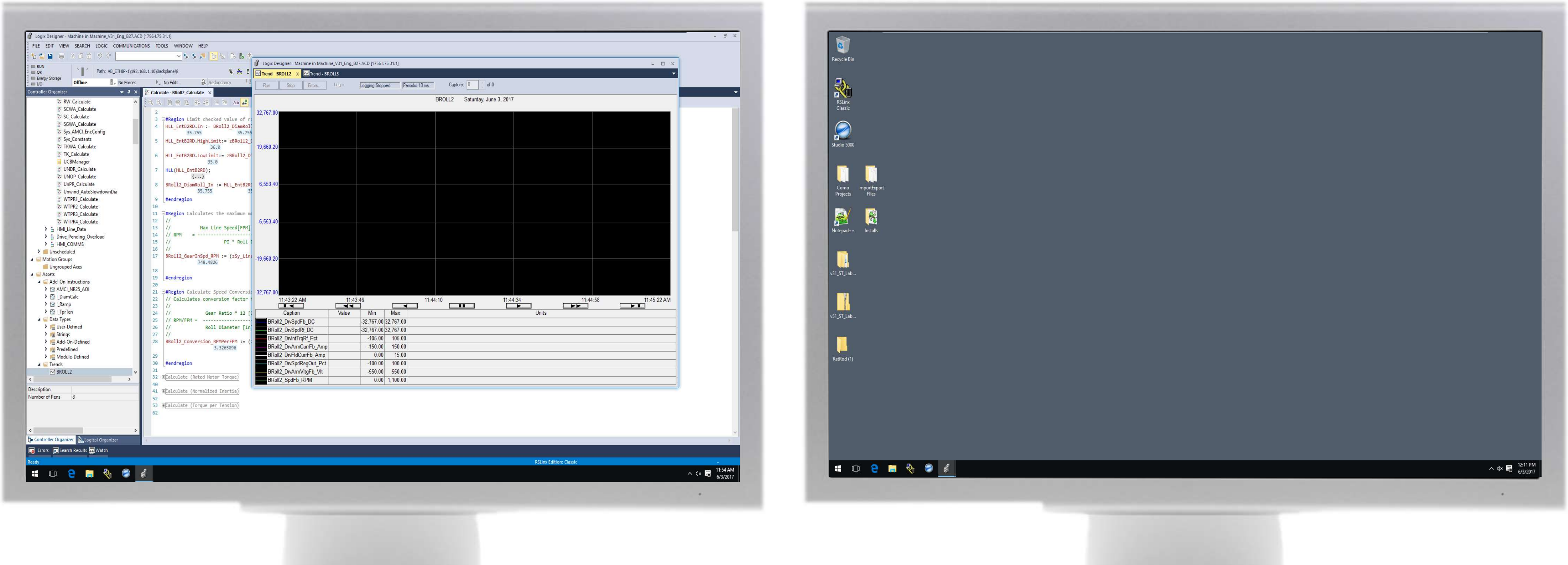






Studio 5000® Logix Designer

Multi Monitor Support



Description	
Number of Pens	8

```
#Region Limit checked value of r
HLL_EntB2RD.In := BRoll2_DiamRol
       35.755              35.755
HLL_EntB2RD.HighLimit:= zBRoll2_D
                      36.0
HLL_EntB2RD.LowLimit:= zBRoll2_Di
                     35.0
HLL(HLL_EntB2RD);
        {...}
BRoll2_DiamRoll_In := HLL_EntB2RD
          35.755              35
#endregion

#Region Calculates the maximum m
//
//          Max Line Speed[FPM]
// RPM    = --------------------
//                   PI * Roll D
//
BRoll2_GearInSpd_RPM := (zSy_Line
           748.4826

#endregion

#Region Calculate Speed Conversio
// Calculates conversion factor f
//
//           Gear Ratio * 12 [I
// RPM/FPM =  ------------------
//           Roll Diameter [In]
//
BRoll2_Conversion_RPMPerFPM := (z
                  3.3265896

#endregion

Calculate (Rated Motor Torque)
Calculate (Normalized Inertia)
Calculate (Torque per Tension)
```

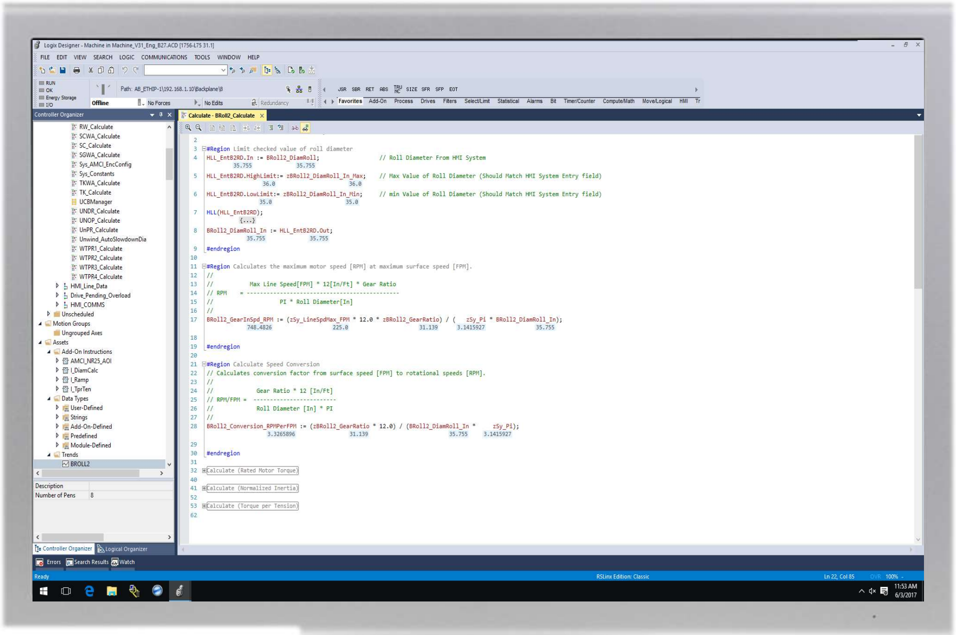
Caption	Value	Min	Max	Units
BRoll2_DrvSpdFb_DC		-32,767.00	32,767.00	
BRoll2_DrvSpdRf_DC		-32,767.00	32,767.00	
BRoll2_DrvIntTrqRf_Pct		-105.00	105.00	
BRoll2_DrvArmCurrFb_Amp		-150.00	150.00	
BRoll2_DrvFldCurrFb_Amp		0.00	15.00	
BRoll2_DrvSpdRegOut_Pct		-100.00	100.00	
BRoll2_DrvArmVltgFb_Vlt		-550.00	550.00	
BRoll2_SpdFb_RPM		0.00	1,100.00	

Organize your view **the way that you like** in a productive manner

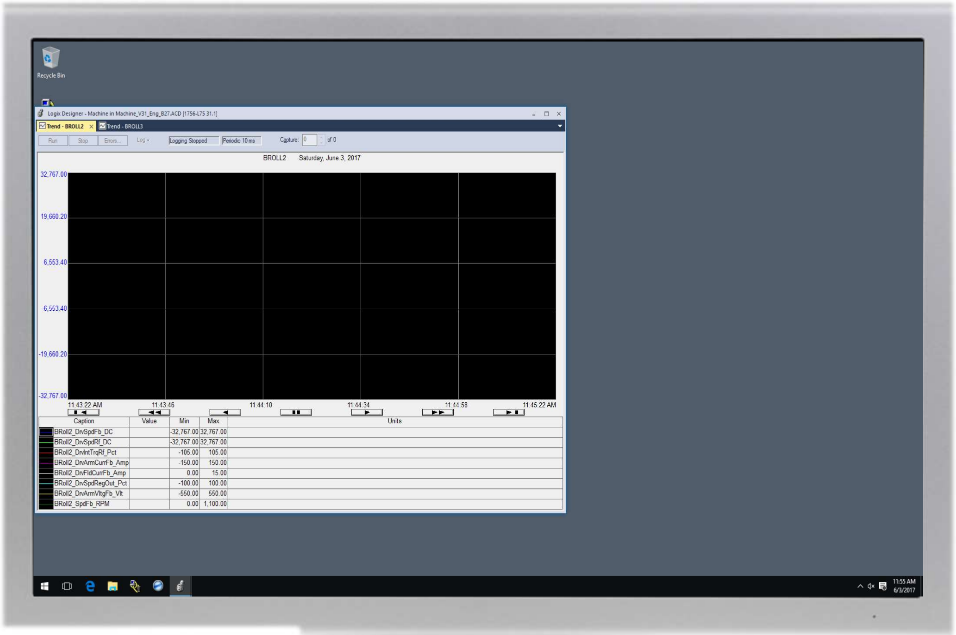
WEXØE

Studio 5000® Logix Designer

Multi Monitor Support



```
#Region Limit checked value of roll diameter
HLL_EntB2RD.In := BRoll2_DiamRoll;                    // Roll Diameter From HMI System
        35.755          35.755

HLL_EntB2RD.HighLimit:= zBRoll2_DiamRoll_In_Max;    // Max Value of Roll Diameter (Should Match HMI System Entry field)
                 36.0                   36.0

HLL_EntB2RD.LowLimit:= zBRoll2_DiamRoll_In_Min;     // min Value of Roll Diameter (Should Match HMI System Entry field)
                35.0                   35.0

HLL(HLL_EntB2RD);
       {...}

BRoll2_DiamRoll_In := HLL_EntB2RD.Out;
        35.755          35.755

#endregion

#Region Calculates the maximum motor speed [RPM] at maximum surface speed [FPM].
//
//          Max Line Speed[FPM] * 12[In/Ft] * Gear Ratio
// RPM    = ---------------------------------------------
//                   PI * Roll Diameter[In]
//
BRoll2_GearInSpd_RPM := (zSy_LineSpdMax_FPM * 12.0 * zBRoll2_GearRatio) / (   zSy_Pi * BRoll2_DiamRoll_In);
          748.4826               225.0                       31.139     3.1415927          35.755


#endregion

#Region Calculate Speed Conversion
// Calculates conversion factor from surface speed [FPM] to rotational speeds [RPM].
//
//            Gear Ratio * 12 [In/Ft]
// RPM/FPM =  ------------------------
//            Roll Diameter [In] * PI
//
BRoll2_Conversion_RPMPerFPM := (zBRoll2_GearRatio * 12.0) / (BRoll2_DiamRoll_In *     zSy_Pi);
              3.3265896               31.139                      35.755     3.1415927


#endregion

Calculate (Rated Motor Torque)

Calculate (Normalized Inertia)

Calculate (Torque per Tension)
```



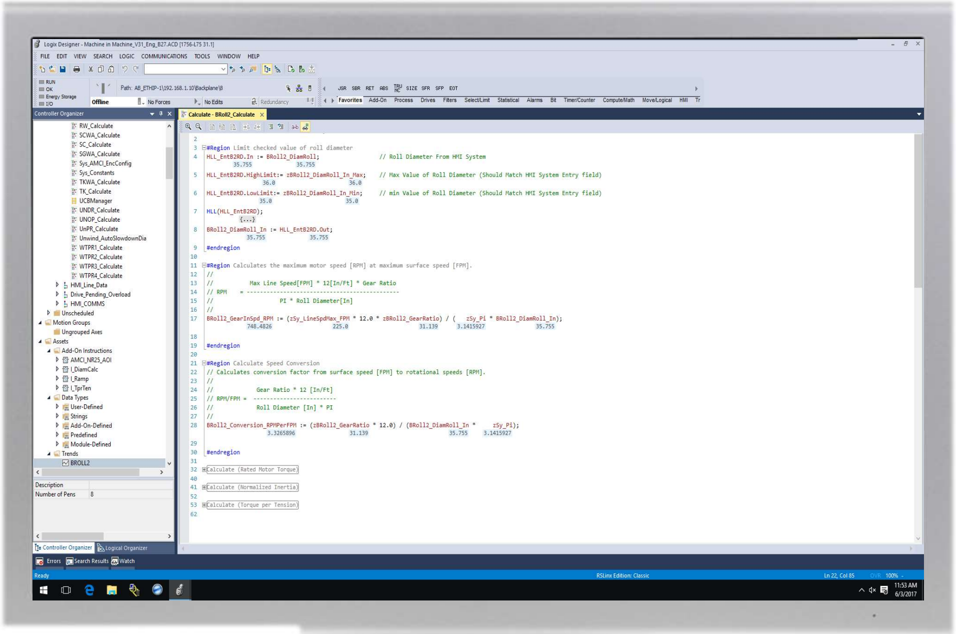
Caption	Value	Min	Max	Units
BRoll2_DrvSpdFb_DC		-32,767.00	32,767.00	
BRoll2_DrvSpdRf_DC		-32,767.00	32,767.00	
BRoll2_DrvIntTrqRf_Pct		-105.00	105.00	
BRoll2_DrvArmCurrFb_Amp		-150.00	150.00	
BRoll2_DrvFldCurrFb_Amp		0.00	15.00	
BRoll2_DrvSpdRegOut_Pct		-100.00	100.00	
BRoll2_DrvArmVltgFb_Vlt		-550.00	550.00	
BRoll2_SpdFb_RPM		0.00	1,100.00	

Organize your view **the way that you like** in a productive manner

WEXØE

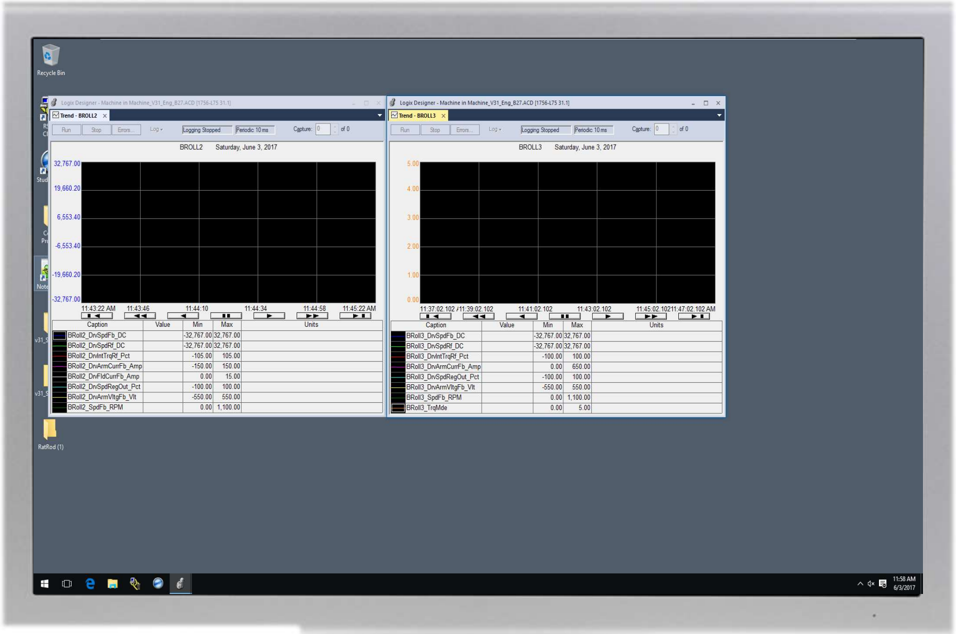
Studio 5000® Logix Designer

Multi Monitor Support



Logix Designer - Machine in Machine_V31_Eng_B27.ACD [1756-L75 31.1]
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WINDOW HELP
Path: AB_ETHIP-1\192.168.1.10\Backplane\8
Offline
No Forces
No Edits
Redundancy
JSR SBR RET ABS TRUNC SIZE SFR SFP EOT
Favorites Add-On Process Drives Filters Select/Limit Statistical Alarms Bit Timer/Counter Compute/Math Move/Logical
Controller Organizer
Calculate - BRoll2_Calculate
Controller Organizer
Logical Organizer
Errors
Search Results
Watch
Ready
RSLinx Edition: Classic

```
#Region Limit checked value of roll diameter
HLL_EntB2RD.In := BRoll2_DiamRoll;                    // Roll Diameter From HMI System
        35.755           35.755
HLL_EntB2RD.HighLimit:= zBRoll2_DiamRoll_In_Max;      // Max Value of Roll Diameter (Should Match HMI System Entry field)
             36.0                 36.0
HLL_EntB2RD.LowLimit:= zBRoll2_DiamRoll_In_Min;       // min Value of Roll Diameter (Should Match HMI System Entry field)
             35.0                35.0
HLL(HLL_EntB2RD);
       {...}
BRoll2_DiamRoll_In := HLL_EntB2RD.Out;
          35.755          35.755
#endregion

#Region Calculates the maximum motor speed [RPM] at maximum surface speed [FPM].
//
//           Max Line Speed[FPM] * 12[In/Ft] * Gear Ratio
// RPM    = ---------------------------------------------
//                    PI * Roll Diameter[In]
//
BRoll2_GearInSpd_RPM := (zSy_LineSpdMax_FPM * 12.0 * zBRoll2_GearRatio) / (   zSy_Pi * BRoll2_DiamRoll_In);
          748.4826               225.0                  31.139      3.1415927           35.755

#endregion

#Region Calculate Speed Conversion
// Calculates conversion factor from surface speed [FPM] to rotational speeds [RPM].
//
//             Gear Ratio * 12 [In/Ft]
// RPM/FPM =  ------------------------
//             Roll Diameter [In] * PI
//
BRoll2_Conversion_RPMPerFPM := (zBRoll2_GearRatio * 12.0) / (BRoll2_DiamRoll_In *     zSy_Pi);
                3.3265896              31.139                        35.755    3.1415927

#endregion

Calculate (Rated Motor Torque)

Calculate (Normalized Inertia)

Calculate (Torque per Tension)
```



Caption	Value	Min	Max	Units
BRoll2_DrvSpdFb_DC		-32,767.00	32,767.00	
BRoll2_DrvSpdRf_DC		-32,767.00	32,767.00	
BRoll2_DrvIntTrqRf_Pct		-105.00	105.00	
BRoll2_DrvArmCurrFb_Amp		-150.00	150.00	
BRoll2_DrvFldCurrFb_Amp		0.00	15.00	
BRoll2_DrvSpdRegOut_Pct		-100.00	100.00	
BRoll2_DrvArmVltgFb_Vlt		-550.00	550.00	
BRoll2_SpdFb_RPM		0.00	1,100.00	

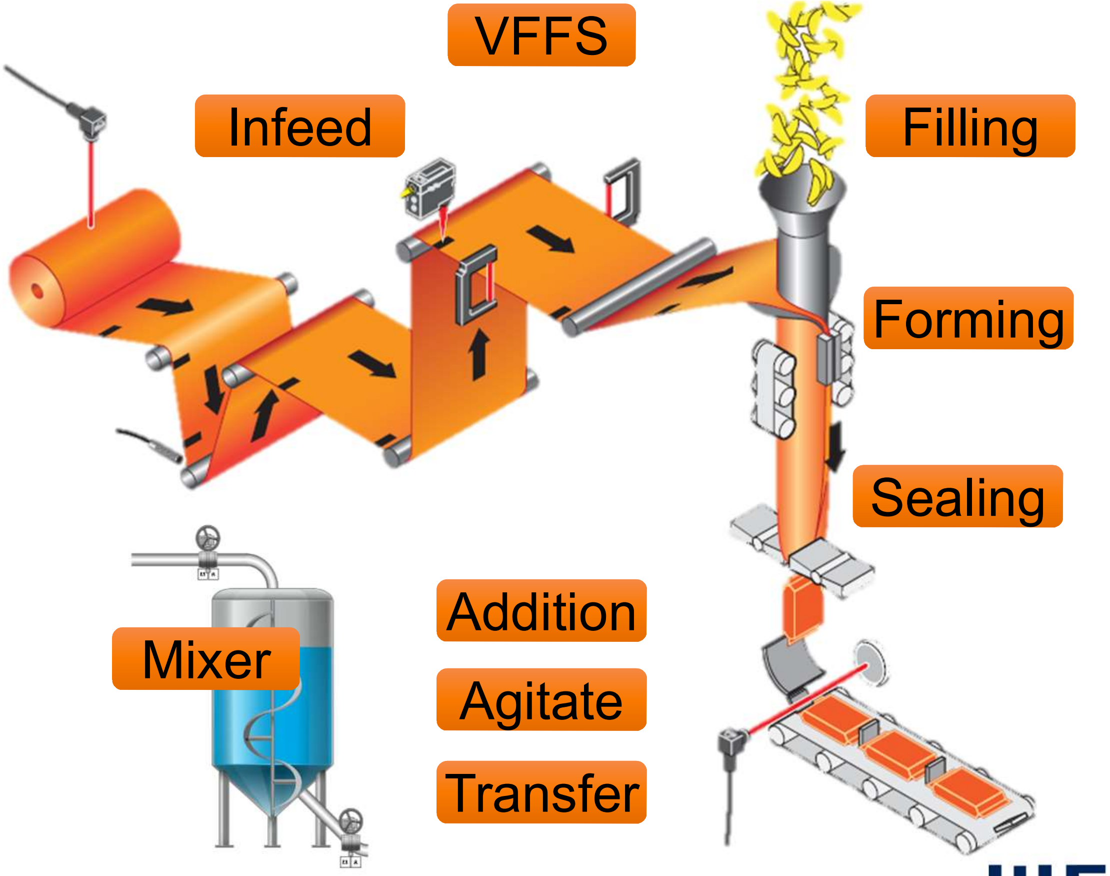
Caption	Value	Min	Max	Units
BRoll3_DrvSpdFb_DC		-32,767.00	32,767.00	
BRoll3_DrvSpdRf_DC		-32,767.00	32,767.00	
BRoll3_DrvIntTrqRf_Pct		-100.00	100.00	
BRoll3_DrvArmCurrFb_Amp		0.00	650.00	
BRoll3_DrvSpdRegOut_Pct		-100.00	100.00	
BRoll3_DrvArmVltgFb_Vlt		-550.00	550.00	
BRoll3_SpdFb_RPM		0.00	1,100.00	
BRoll3_TrqMde		0.00	5.00	

Organize your view **the way that you like** in a productive manner

Studio 5000® Logix Designer

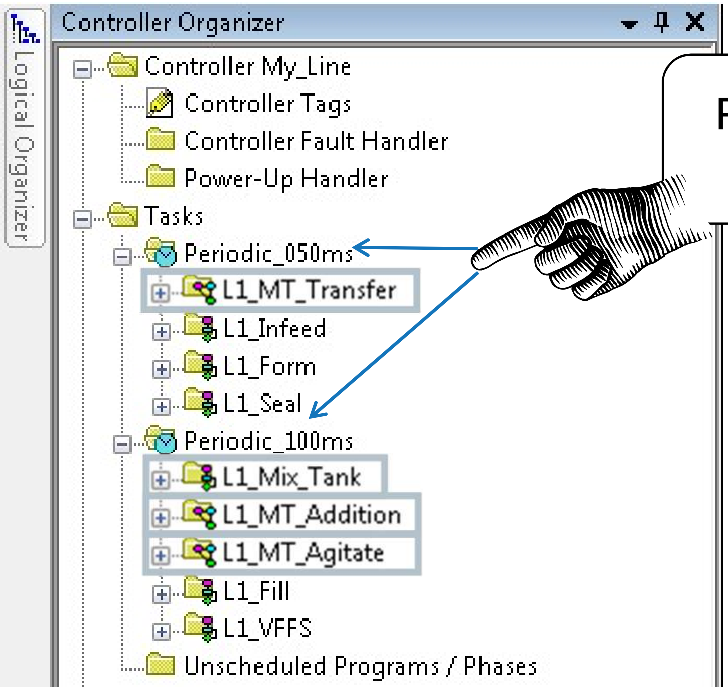
Logical Organizer

- VFFS
  - Infeed
  - Forming
  - Filling
  - Sealing
- Mixer
  - Addition
  - Agitate
  - Transfer



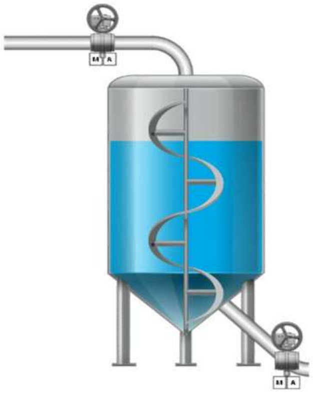
Studio 5000® **Logix Designer**

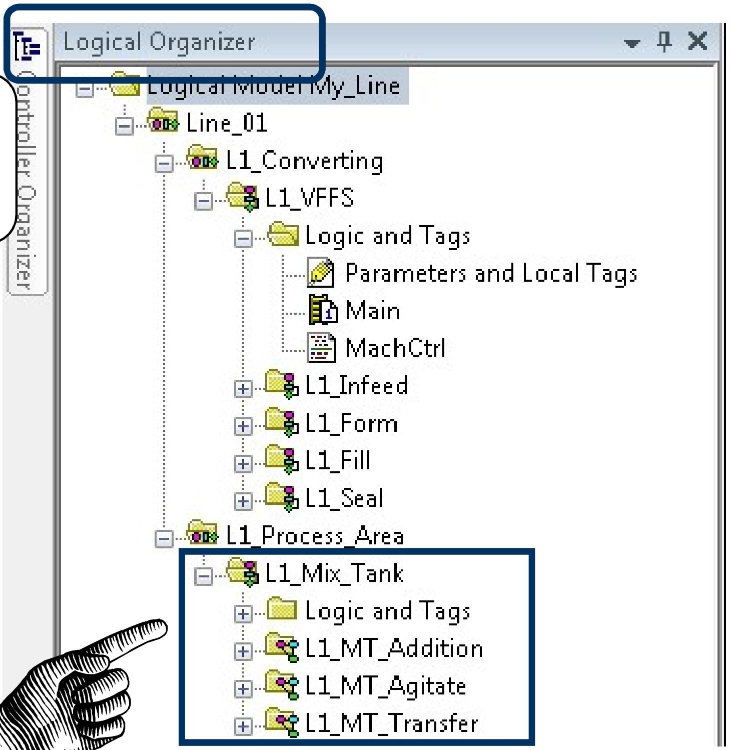
Why Add a Logical Organizer?



```
Controller Organizer
⊟ Controller My_Line
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
⊟ Tasks
  ⊟ Periodic_050ms
    ⊞ L1_MT_Transfer
    ⊞ L1_Infeed
    ⊞ L1_Form
    ⊞ L1_Seal
  ⊟ Periodic_100ms
    ⊞ L1_Mix_Tank
    ⊞ L1_MT_Addition
    ⊞ L1_MT_Agitate
    ⊞ L1_Fill
    ⊞ L1_VFFS
  Unscheduled Programs / Phases
```

Programs Span the Tasks



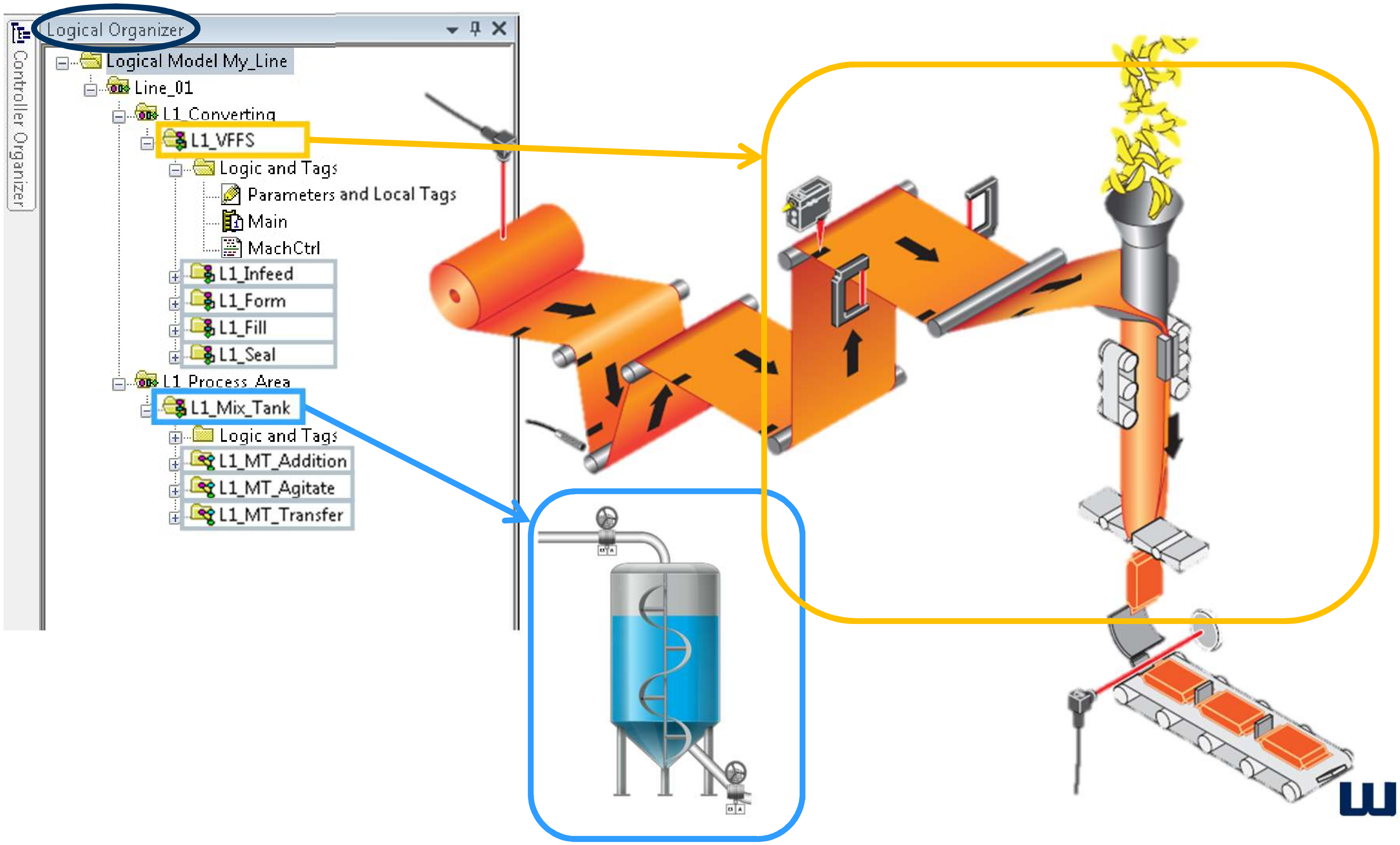


```
Logical Organizer
⊟ Logical Model My_Line
  ⊟ Line_01
    ⊟ L1_Converting
      ⊟ L1_VFFS
        ⊟ Logic and Tags
            Parameters and Local Tags
            Main
            MachCtrl
        ⊞ L1_Infeed
        ⊞ L1_Form
        ⊞ L1_Fill
        ⊞ L1_Seal
    ⊟ L1_Process_Area
      ⊟ L1_Mix_Tank
        ⊞ Logic and Tags
        ⊞ L1_MT_Addition
        ⊞ L1_MT_Agitate
        ⊞ L1_MT_Transfer
```

Logically Organized

Studio 5000® Logix Designer

Logical Organizer View



Logical Organizer

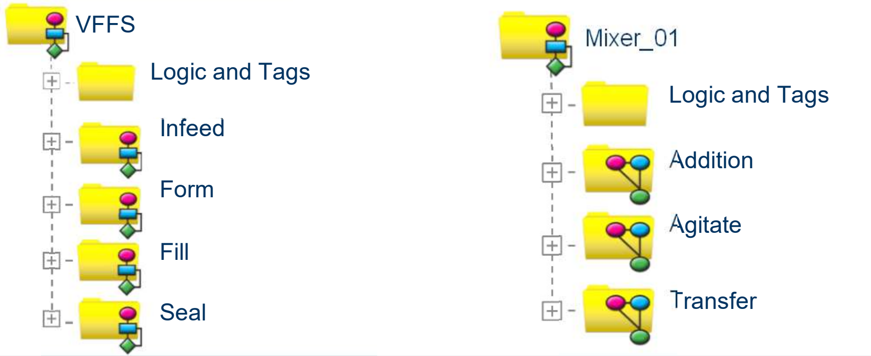
Controller Organizer

- Logical Model My_Line
  - Line_01
    - L1_Converting
      - L1_VFFS
        - Logic and Tags
          - Parameters and Local Tags
          - Main
          - MachCtrl
        - L1_Infeed
        - L1_Form
        - L1_Fill
        - L1_Seal
    - L1_Process_Area
      - L1_Mix_Tank
        - Logic and Tags
        - L1_MT_Addition
        - L1_MT_Agitate
        - L1_MT_Transfer

Studio 5000® Logix Designer

Logical Organizer Benefits

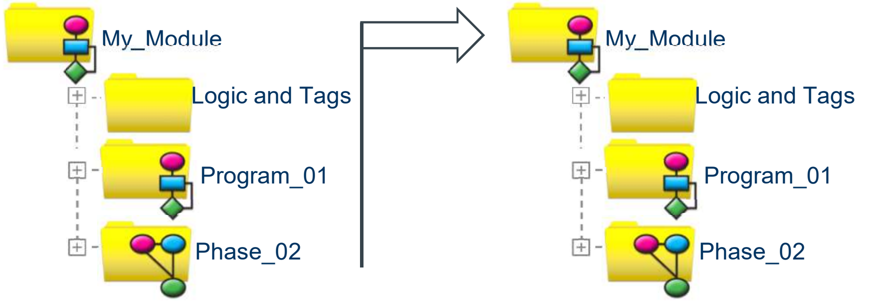
Create Logical Hierarchy of Code



VFFS
Logic and Tags
Infeed
Form
Fill
Seal

Mixer_01
Logic and Tags
Addition
Agitate
Transfer

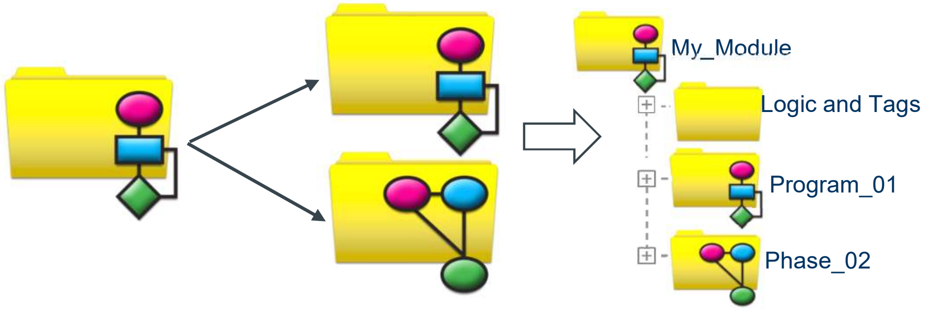
Easier to Manage



My_Module
Logic and Tags
Program_01
Phase_02

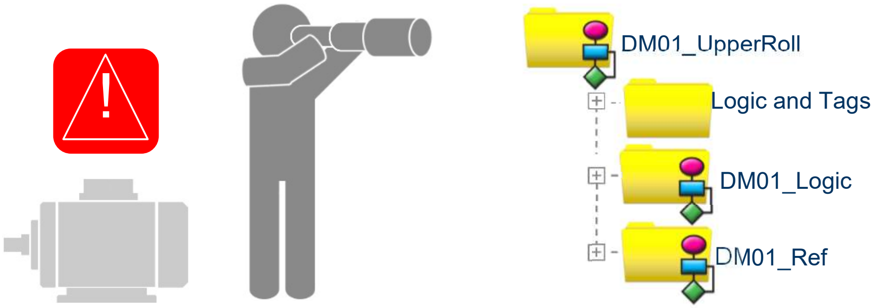
My_Module
Logic and Tags
Program_01
Phase_02

Aids Modular Coding



My_Module
Logic and Tags
Program_01
Phase_02

Easier to Locate Relevant Code



DM01_UpperRoll
Logic and Tags
DM01_Logic
DM01_Ref

Studio 5000® **Logix Designer**

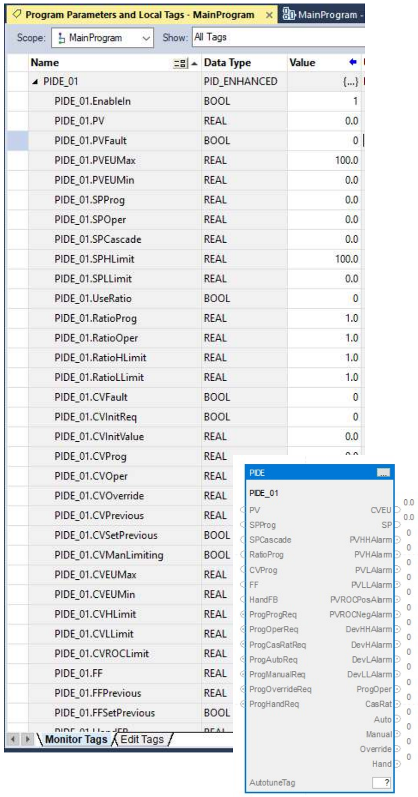
Automatic Generated Descriptive Tags

Automatic generation of descriptive Logix tag names with proper data types respective of each parameter

- ✓ Tags are the actual network I/O as set up in the Module Definition window
- ✓ Can have a mixture of data types

Saves development time

- ✓ Descriptive tag names reduce need to manually add tag descriptions



Program Parameters and Local Tags - MainProgram | MainProgram -

Scope: MainProgram Show: All Tags

Name	Data Type	Value
PIDE_01	PID_ENHANCED	{...}
PIDE_01.EnableIn	BOOL	1
PIDE_01.PV	REAL	0.0
PIDE_01.PVFault	BOOL	0
PIDE_01.PVEUMax	REAL	100.0
PIDE_01.PVEUMin	REAL	0.0
PIDE_01.SPProg	REAL	0.0
PIDE_01.SPOper	REAL	0.0
PIDE_01.SPCascade	REAL	0.0
PIDE_01.SPHLimit	REAL	100.0
PIDE_01.SPLLimit	REAL	0.0
PIDE_01.UseRatio	BOOL	0
PIDE_01.RatioProg	REAL	1.0
PIDE_01.RatioOper	REAL	1.0
PIDE_01.RatioHLimit	REAL	1.0
PIDE_01.RatioLLimit	REAL	1.0
PIDE_01.CVFault	BOOL	0
PIDE_01.CVInitReq	BOOL	0
PIDE_01.CVInitValue	REAL	0.0
PIDE_01.CVProg	REAL	
PIDE_01.CVOper	REAL	
PIDE_01.CVOverride	REAL	
PIDE_01.CVPrevious	REAL	
PIDE_01.CVSetPrevious	BOOL	
PIDE_01.CVManLimiting	BOOL	
PIDE_01.CVEUMax	REAL	
PIDE_01.CVEUMin	REAL	
PIDE_01.CVHLimit	REAL	
PIDE_01.CVLLimit	REAL	
PIDE_01.CVROCLimit	REAL	
PIDE_01.FF	REAL	
PIDE_01.FFPrevious	REAL	
PIDE_01.FFSetPrevious	BOOL	

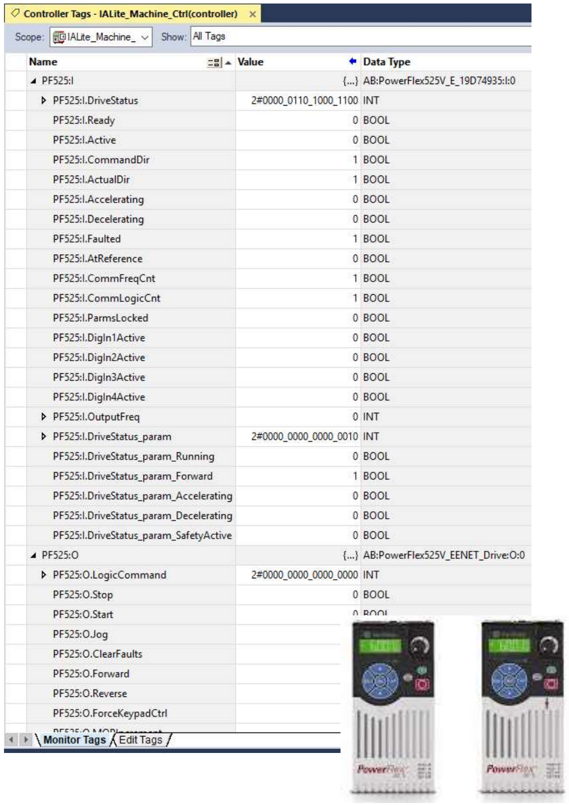
Monitor Tags | Edit Tags

PIDE ...

PIDE_01

Input	Output	Value
PV	CVEU	0.0
SPProg	SP	0.0
SPCascade	PVHHAlarm	0
RatioProg	PVHAlarm	0
CVProg	PVLAlarm	0
FF	PVLLAlarm	0
HandFB	PVROCPosAlarm	0
ProgProgReq	PVROCNegAlarm	0
ProgOperReq	DevHHAlarm	0
ProgCasRatReq	DevHAlarm	0
ProgAutoReq	DevLAlarm	0
ProgManualReq	DevLLAlarm	0
ProgOverrideReq	ProgOper	0
ProgHandReq	CasRat	0
	Auto	0
	Manual	0
	Override	0
	Hand	0

AutotuneTag ?



Controller Tags - IALite_Machine_Ctrl(controller)

Scope: IALite_Machine_ Show: All Tags

Name	Value	Data Type
PF525:I	{...}	AB:PowerFlex525V_E_19D74935:I:0
PF525:I.DriveStatus	2#0000_0110_1000_1100	INT
PF525:I.Ready	0	BOOL
PF525:I.Active	0	BOOL
PF525:I.CommandDir	1	BOOL
PF525:I.ActualDir	1	BOOL
PF525:I.Accelerating	0	BOOL
PF525:I.Decelerating	0	BOOL
PF525:I.Faulted	1	BOOL
PF525:I.AtReference	0	BOOL
PF525:I.CommFreqCnt	1	BOOL
PF525:I.CommLogicCnt	1	BOOL
PF525:I.ParmsLocked	0	BOOL
PF525:I.DigIn1Active	0	BOOL
PF525:I.DigIn2Active	0	BOOL
PF525:I.DigIn3Active	0	BOOL
PF525:I.DigIn4Active	0	BOOL
PF525:I.OutputFreq	0	INT
PF525:I.DriveStatus_param	2#0000_0000_0000_0010	INT
PF525:I.DriveStatus_param_Running	0	BOOL
PF525:I.DriveStatus_param_Forward	1	BOOL
PF525:I.DriveStatus_param_Accelerating	0	BOOL
PF525:I.DriveStatus_param_Decelerating	0	BOOL
PF525:I.DriveStatus_param_SafetyActive	0	BOOL
PF525:O	{...}	AB:PowerFlex525V_EENET_Drive:O:0
PF525:O.LogicCommand	2#0000_0000_0000_0000	INT
PF525:O.Stop	0	BOOL
PF525:O.Start		
PF525:O.Jog		
PF525:O.ClearFaults		
PF525:O.Forward		
PF525:O.Reverse		
PF525:O.ForceKeypadCtrl		

Monitor Tags | Edit Tags

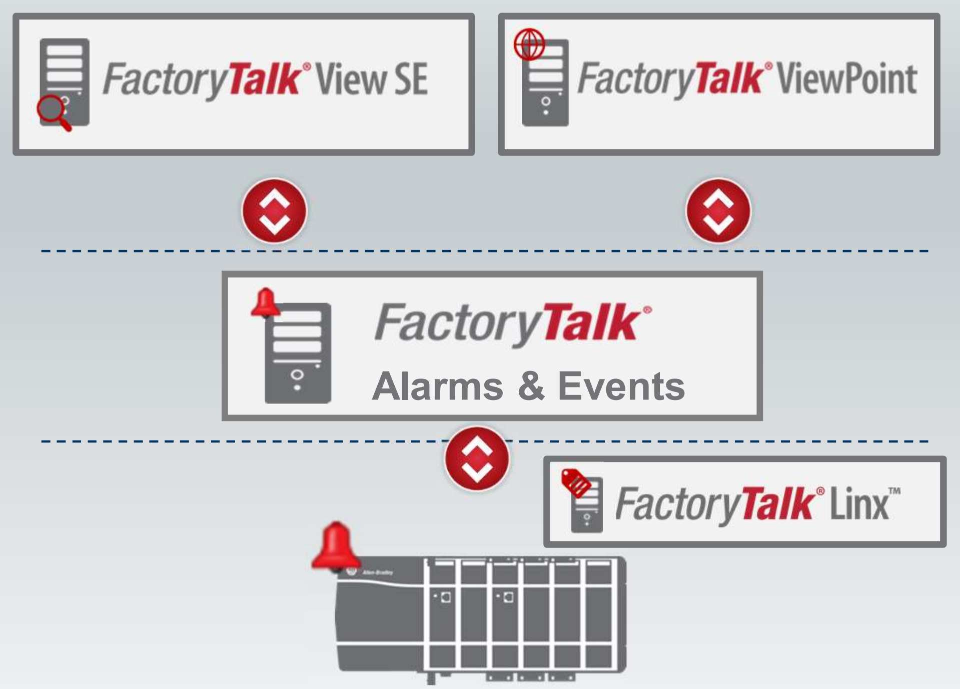
Alarming Capabilities

***Studio** 5000® **Logix Designer** Driving Efficiency During Design Time*

Studio 5000

Studio 5000® Logix Designer

Logix Tag-based Alarms



Supported in CompactLogix 5380, CompactLogix 5480, and ControlLogix 5580 controllers only

Overview

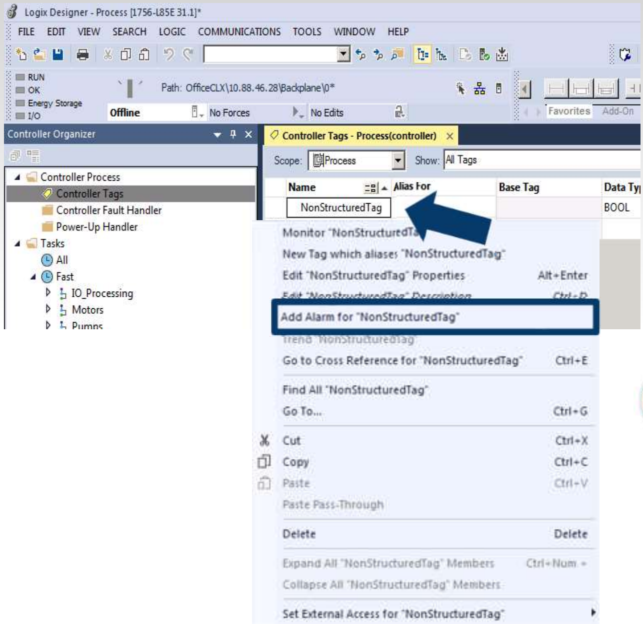
- ✓ V31 Logix Designer application added new alarming functionality in the controller
- ✓ With the new Logix tag-based alarming, alarms can now be defined on “tags” or “structures” in the controller with periodic evaluation

Benefits

- ✓ No need to add an instruction – simplified design workflows inside Studio 5000 Logix Designer® application
- ✓ No additional programming required – alarms automatically sent to FactoryTalk® Alarms and Events application
- ✓ New alarm manager provides one place to configure
- ✓ Small memory footprint – great for applications that have high alarm counts
- ✓ Alarm definitions allow for increased modularity
- ✓ Supports bulk generation of alarms via XML import/export

Studio 5000® **Logix Designer –** Logix Tag-Based Alarms

Adding an Alarm to a Tag

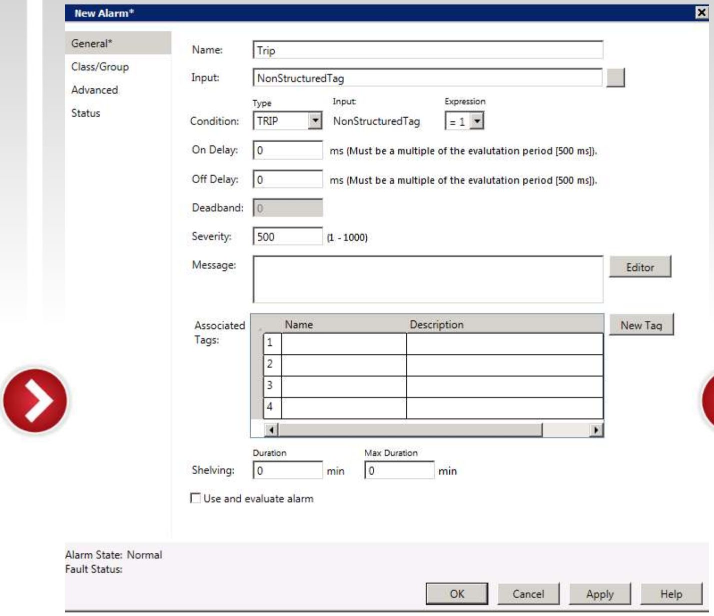


Logix Designer - Process [1756-L85E 31.1]*
FILE
EDIT
VIEW
SEARCH
LOGIC
COMMUNICATIONS
TOOLS
WINDOW
HELP
RUN
OK
Energy Storage
I/O
Offline
No Forces
No Edits
Favorites
Add-On
Controller Organizer
Controller Process
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
All
Fast
IO_Processing
Motors
Pumps
Controller Tags - Process(controller)
Monitor "NonStructuredTa
New Tag which aliases "NonStructuredTag"
Edit "NonStructuredTag" Properties Alt+Enter
Edit "NonStructuredTag" Description Ctrl+D
Add Alarm for "NonStructuredTag"
Trend "NonStructuredTag"
Go to Cross Reference for "NonStructuredTag" Ctrl+E
Find All "NonStructuredTag"
Go To... Ctrl+G
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Pass-Through
Delete Delete
Expand All "NonStructuredTag" Members Ctrl+Num +
Collapse All "NonStructuredTag" Members
Set External Access for "NonStructuredTag"

Path: OfficeCLX\10.88.46.28\Backplane\0*

Scope: Process Show: All Tags

Name	Alias For	Base Tag	Data Typ
NonStructuredTag			BOOL



Name: Trip

Input: NonStructuredTag

Condition: Type TRIP Input NonStructuredTag Expression = 1

On Delay: 0 ms (Must be a multiple of the evalutation period [500 ms]).

Off Delay: 0 ms (Must be a multiple of the evalutation period [500 ms]).

Deadband: 0

Severity: 500 (1 - 1000)

Message:

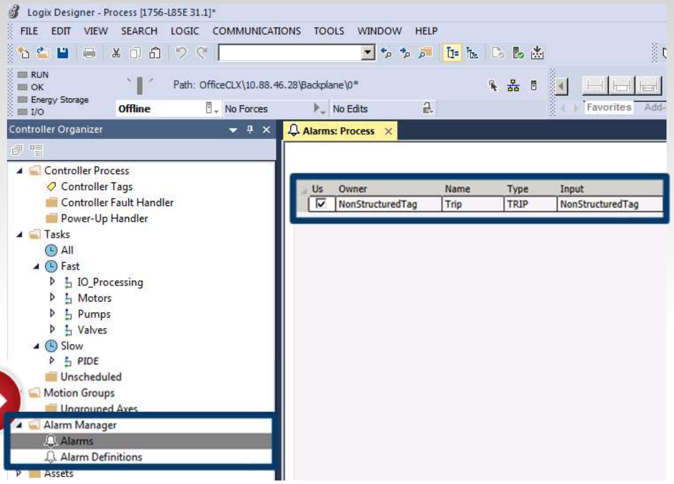
Associated Tags:

	Name	Description
1		
2		
3		
4		

Shelving: Duration 0 min Max Duration 0 min

☐ Use and evaluate alarm

Alarm State: Normal
Fault Status:



Path: OfficeCLX\10.88.46.28\Backplane\0*

Us	Owner	Name	Type	Input
☑	NonStructuredTag	Trip	TRIP	NonStructuredTag

- ✓ Right click on a “tag”
- ✓ Select “Add Alarm” from context menu

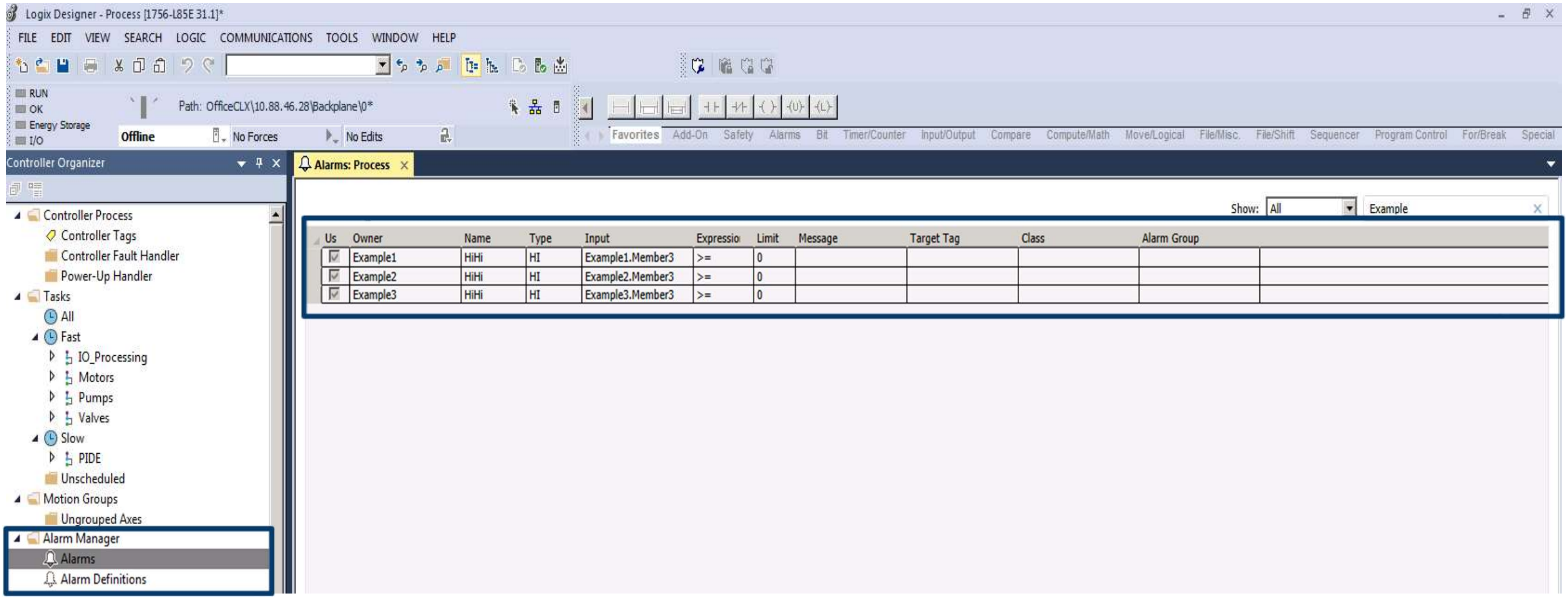
- ✓ Name the alarm and add configuration parameters
- ✓ Select “OK”

- ✓ Alarm can now be seen in the new alarm manager and is active in controller

WEXØE

Studio 5000® **Logix Designer –** Logix Tag-Based Alarms

Alarm Manager



Path: OfficeCLX\10.88.46.28\Backplane\0*

Show: All

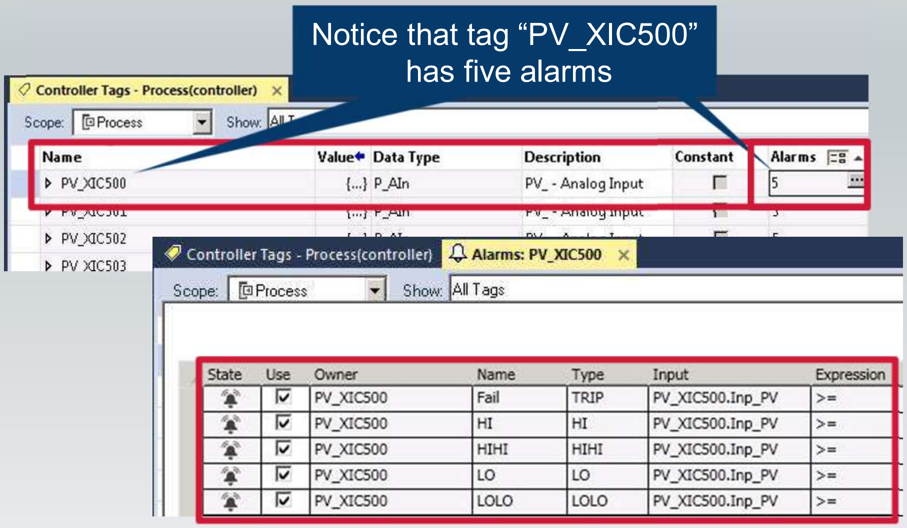
Example

Us	Owner	Name	Type	Input	Expressio	Limit	Message	Target Tag	Class	Alarm Group
☑	Example1	HiHi	HI	Example1.Member3	>=	0				
☑	Example2	HiHi	HI	Example2.Member3	>=	0				
☑	Example3	HiHi	HI	Example3.Member3	>=	0				

Every structure instance will now have an alarm. Alarm instances can now be customized!

Studio 5000® **Logix Designer –** Alarm Library Management

Troubleshooting and Diagnostic Enhancements



Notice that tag "PV_XIC500" has five alarms

Controller Tags - Process(controller)

Scope: Process Show: All T

Name	Value	Data Type	Description	Constant	Alarms
PV_XIC500	{...}	P_AIn	PV_ - Analog Input	□	5
PV_XIC501	{...}	P_AIn	PV_ - Analog Input		
PV_XIC502					
PV_XIC503					

Controller Tags - Process(controller) | Alarms: PV_XIC500

Scope: Process Show: All Tags

State	Use	Owner	Name	Type	Input	Expression
	☑	PV_XIC500	Fail	TRIP	PV_XIC500.Inp_PV	>=
	☑	PV_XIC500	HI	HI	PV_XIC500.Inp_PV	>=
	☑	PV_XIC500	HIHI	HIHI	PV_XIC500.Inp_PV	>=
	☑	PV_XIC500	LO	LO	PV_XIC500.Inp_PV	>=
	☑	PV_XIC500	LOLO	LOLO	PV_XIC500.Inp_PV	>=

- **Tag Browser**
  - ✓ View alarm count on per-tag basis

- **Alarm Manager**
  - ✓ View status of each alarm in real time

Language Enhancements

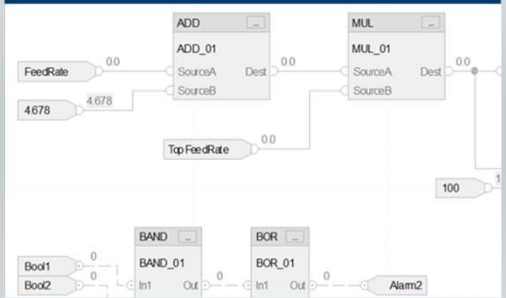
***Studio** 5000® **Logix Designer** FBD & ST Enhancements*

Studio 5000

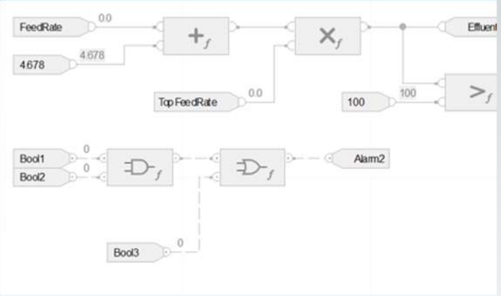
Studio 5000® Logix Designer

FBD Improved Functions

Before



After



Supported in CompactLogix 5380, CompactLogix 5480, and ControlLogix 5580 controllers only

Overview

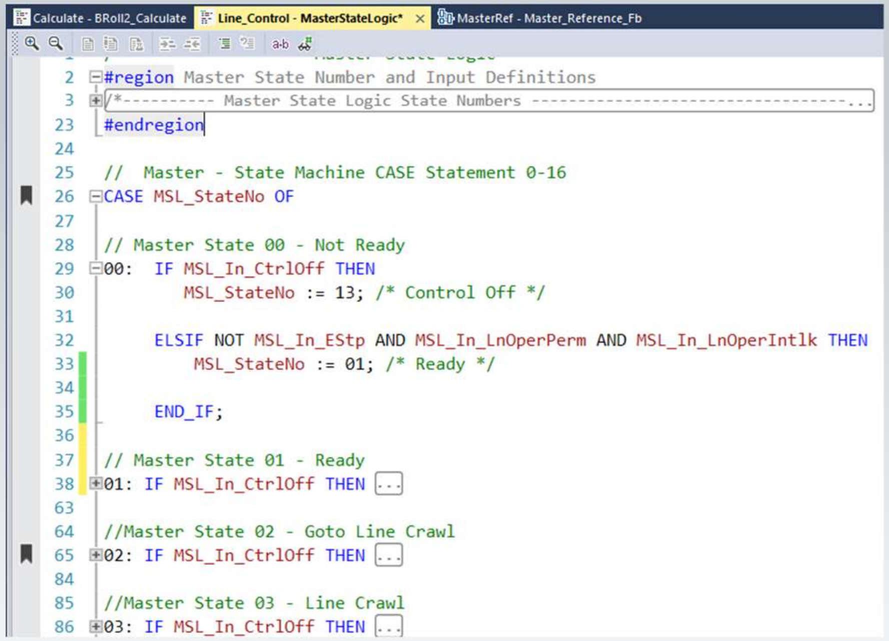
- ✓ Simpler function block construction mitigates how much configuration is needed, lowering overall design times

Benefits

- ✓ Faster troubleshooting with simpler user interface
- ✓ Drag and drop functions available without additional configuration
- ✓ Increased function performance leads to lower controller utilizations
- ✓ New functions to replace existing instructions
  - ✓ Smaller visual footprint
  - ✓ No backing tag
  - ✓ More intuitive symbol-based functions
  - ✓ Compare/Compute/Boolean Logic Instructions

Studio 5000® Logix Designer

ST Editor Improvements


Calculate - BRoll2_Calculate
Line_Control - MasterStateLogic*
MasterRef - Master_Reference_Fb

```
#region Master State Number and Input Definitions
/*---------- Master State Logic State Numbers ------------------------------------...
#endregion

//  Master - State Machine CASE Statement 0-16
CASE MSL_StateNo OF

// Master State 00 - Not Ready
00:  IF MSL_In_CtrlOff THEN
        MSL_StateNo := 13; /* Control Off */

     ELSIF NOT MSL_In_EStp AND MSL_In_LnOperPerm AND MSL_In_LnOperIntlk THEN
         MSL_StateNo := 01; /* Ready */

     END_IF;

// Master State 01 - Ready
01: IF MSL_In_CtrlOff THEN ...

//Master State 02 - Goto Line Crawl
02: IF MSL_In_CtrlOff THEN ...

//Master State 03 - Line Crawl
03: IF MSL_In_CtrlOff THEN ...
```

Overview

A modernized structured text editor packed with features for productive programming and editing

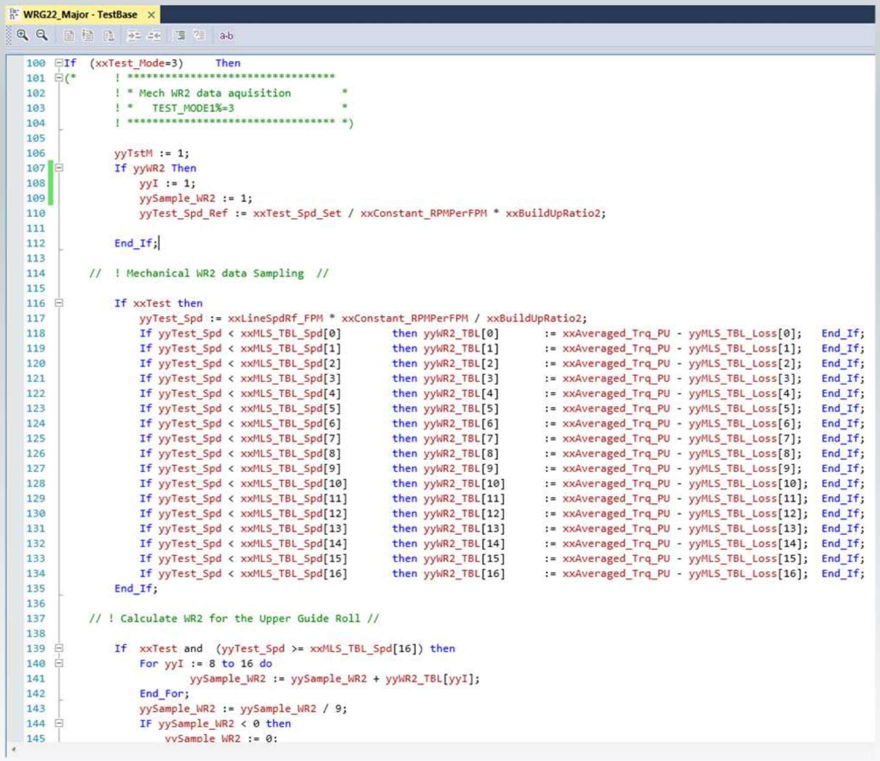
- ✓ Line numbers and bookmarks
- ✓ Descriptive tool tips and syntax highlighting
- ✓ Multi-line select and mouse scrolling
- ✓ Change and verify bars
- ✓ Collapsible code segments
- ✓ Inline value monitoring
- ✓ Code snippets and smart indent capabilities

Benefits

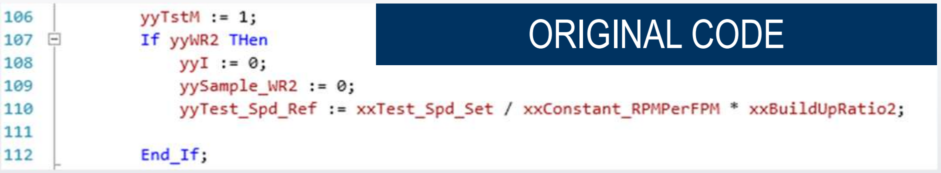
- ✓ Modern programming capabilities
- ✓ Increased productivity
- ✓ Efficient code development and editing

Studio 5000® **Logix Designer**

ST Editor Improvements – Change Bars

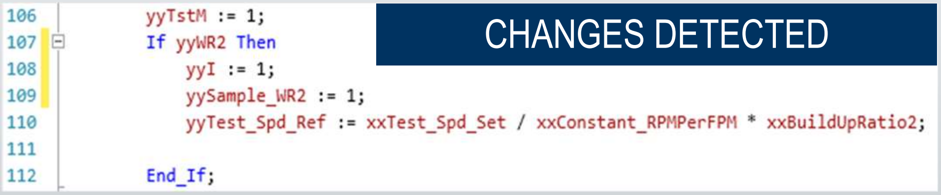


```
If  (xxTest_Mode=3)     Then
(*      ! *********************************
        ! * Mech WR2 data aquisition       *
        ! *   TEST_MODE1%=3                *
        ! ********************************* *)

        yyTstM := 1;
        If yyWR2 Then
            yyI := 1;
            yySample_WR2 := 1;
            yyTest_Spd_Ref := xxTest_Spd_Set / xxConstant_RPMPerFPM * xxBuildUpRatio2;

        End_If;

    //  ! Mechanical WR2 data Sampling  //

        If xxTest then
            yyTest_Spd := xxLineSpdRf_FPM * xxConstant_RPMPerFPM / xxBuildUpRatio2;
            If yyTest_Spd < xxMLS_TBL_Spd[0]        then yyWR2_TBL[0]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[0];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[1]        then yyWR2_TBL[1]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[1];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[2]        then yyWR2_TBL[2]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[2];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[3]        then yyWR2_TBL[3]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[3];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[4]        then yyWR2_TBL[4]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[4];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[5]        then yyWR2_TBL[5]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[5];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[6]        then yyWR2_TBL[6]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[6];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[7]        then yyWR2_TBL[7]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[7];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[8]        then yyWR2_TBL[8]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[8];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[9]        then yyWR2_TBL[9]       := xxAveraged_Trq_PU - yyMLS_TBL_Loss[9];   End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[10]       then yyWR2_TBL[10]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[10];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[11]       then yyWR2_TBL[11]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[11];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[12]       then yyWR2_TBL[12]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[12];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[13]       then yyWR2_TBL[13]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[13];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[14]       then yyWR2_TBL[14]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[14];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[15]       then yyWR2_TBL[15]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[15];  End_If;
            If yyTest_Spd < xxMLS_TBL_Spd[16]       then yyWR2_TBL[16]      := xxAveraged_Trq_PU - yyMLS_TBL_Loss[16];  End_If;
        End_If;

    // ! Calculate WR2 for the Upper Guide Roll //

        If  xxTest and  (yyTest_Spd >= xxMLS_TBL_Spd[16]) then
            For yyI := 8 to 16 do
                    yySample_WR2 := yySample_WR2 + yyWR2_TBL[yyI];
            End_For;
            yySample_WR2 := yySample_WR2 / 9;
            IF yySample_WR2 < 0 then
                yySample_WR2 := 0;
```



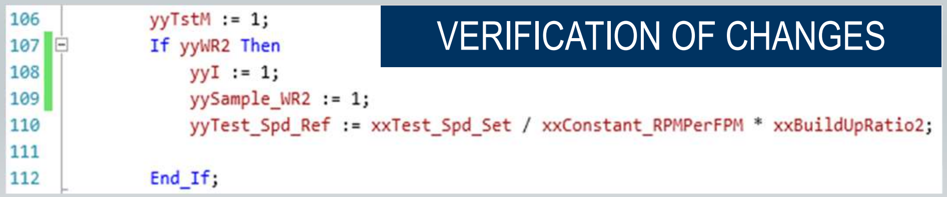
ORIGINAL CODE

```
yyTstM := 1;
If yyWR2 THen
    yyI := 0;
    yySample_WR2 := 0;
    yyTest_Spd_Ref := xxTest_Spd_Set / xxConstant_RPMPerFPM * xxBuildUpRatio2;

End_If;
```



CHANGES DETECTED

```
yyTstM := 1;
If yyWR2 Then
    yyI := 1;
    yySample_WR2 := 1;
    yyTest_Spd_Ref := xxTest_Spd_Set / xxConstant_RPMPerFPM * xxBuildUpRatio2;

End_If;
```



VERIFICATION OF CHANGES

```
yyTstM := 1;
If yyWR2 Then
    yyI := 1;
    yySample_WR2 := 1;
    yyTest_Spd_Ref := xxTest_Spd_Set / xxConstant_RPMPerFPM * xxBuildUpRatio2;

End_If;
```

Productivity

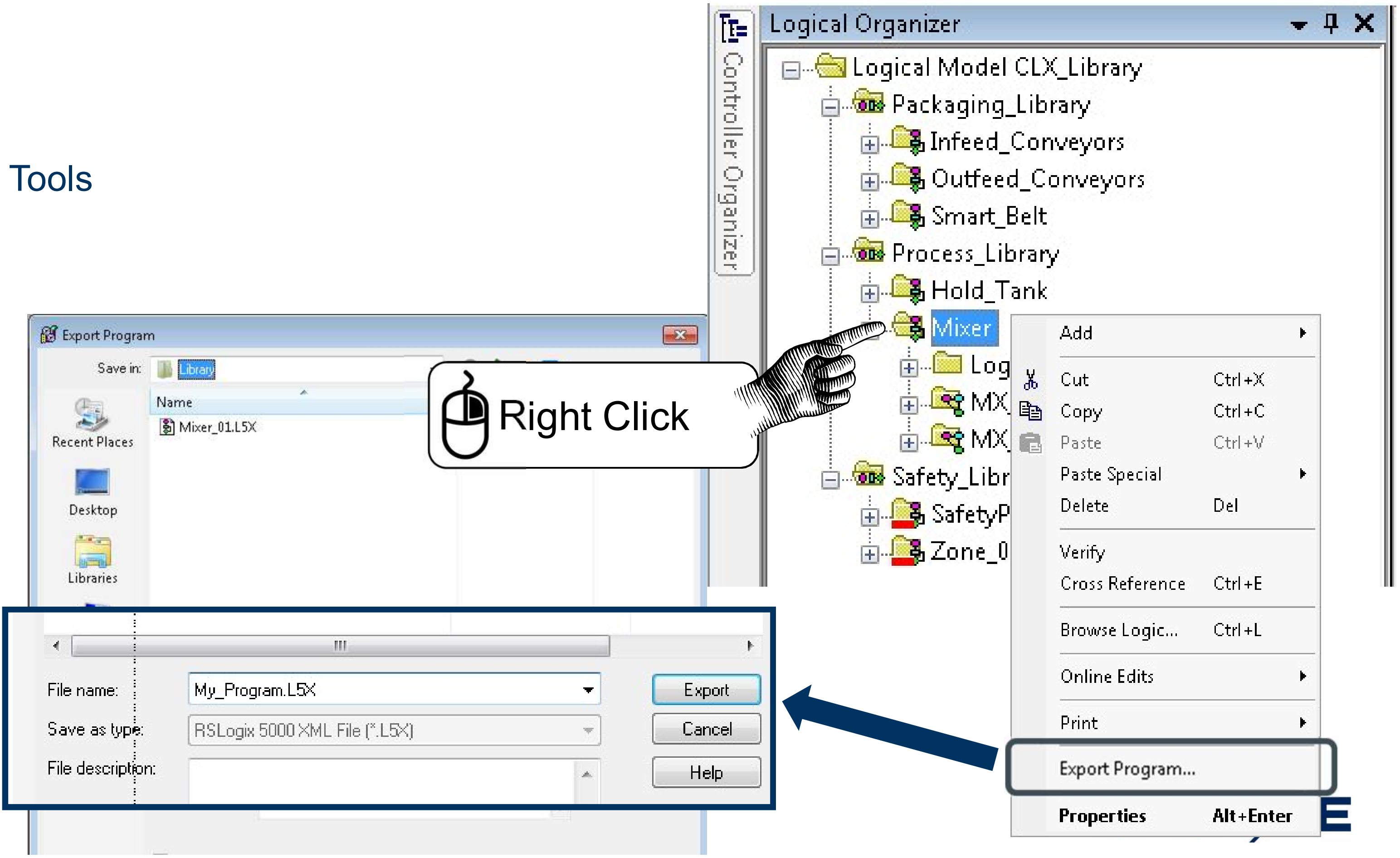
***Studio** 5000® **Logix Designer** Driving Efficiency During Design Time*

Studio 5000

Studio 5000® **Logix Designer -** Library Options

Utilizing .XML

- **Using XML Code**
  - ✓ Import/Export
  - ✓ Library Management Tools



Export Program

Save in: Library

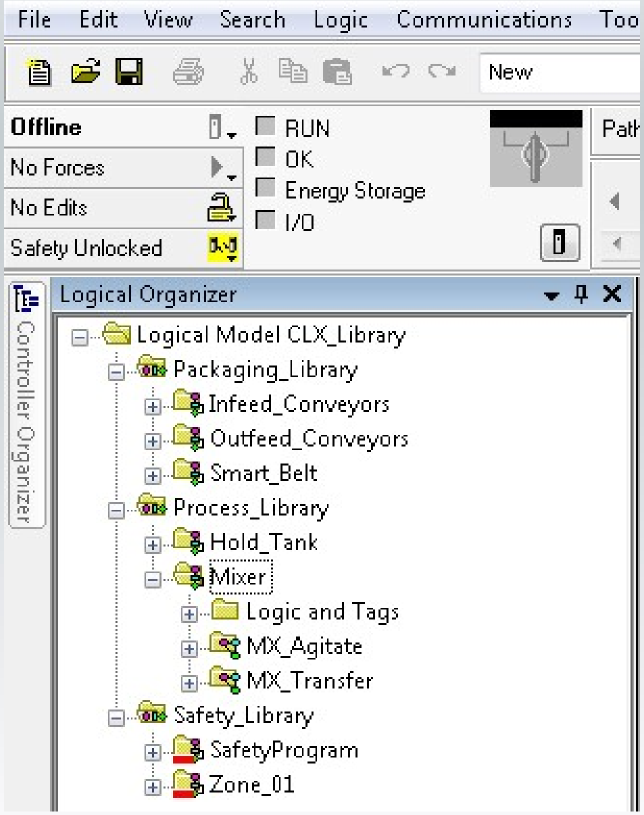
File name: My_Program.L5X

Save as type: RSLogix 5000 XML File (*.L5X)

File description:

Studio 5000® **Logix Designer -** Library Concept

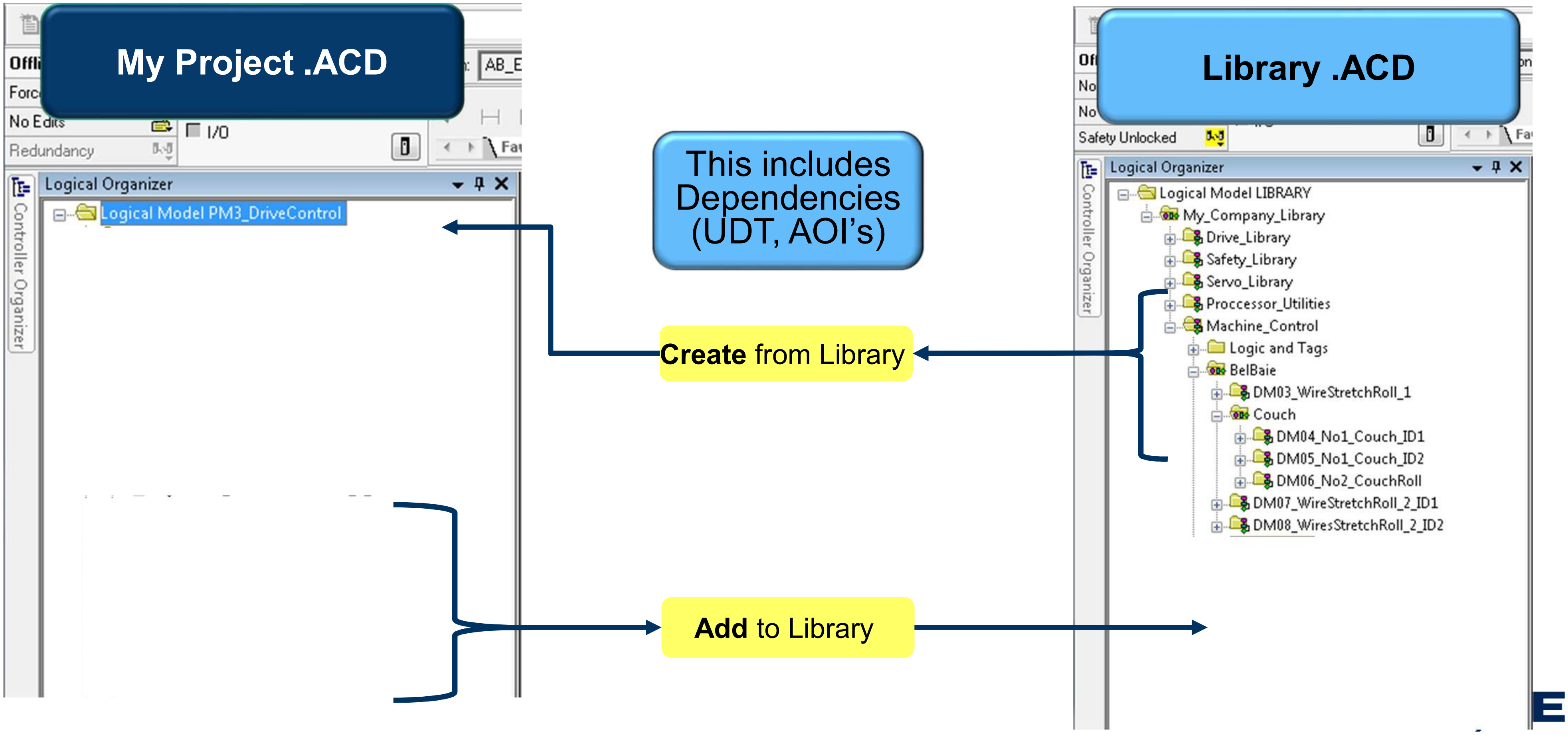
Utilizing an .ACD file as repository



- **Library File (.ACD File)**
  - ✓ Stored Repository for Code
- **Supported content**
  - ✓ Programs, Phases, UDT's, AOI's, & Routines
  - ✓ Basically anything that can be imported/exported or copy/pasted in Logix
- **Capability**
  - ✓ Dependencies & Collisions are managed
  - ✓ Edit entire library content
  - ✓ Organizational hierarchies supported

Studio 5000® **Logix Designer -** Library Options

Utilizing an .ACD file as repository



My Project .ACD

Library .ACD

This includes Dependencies (UDT, AOI's)

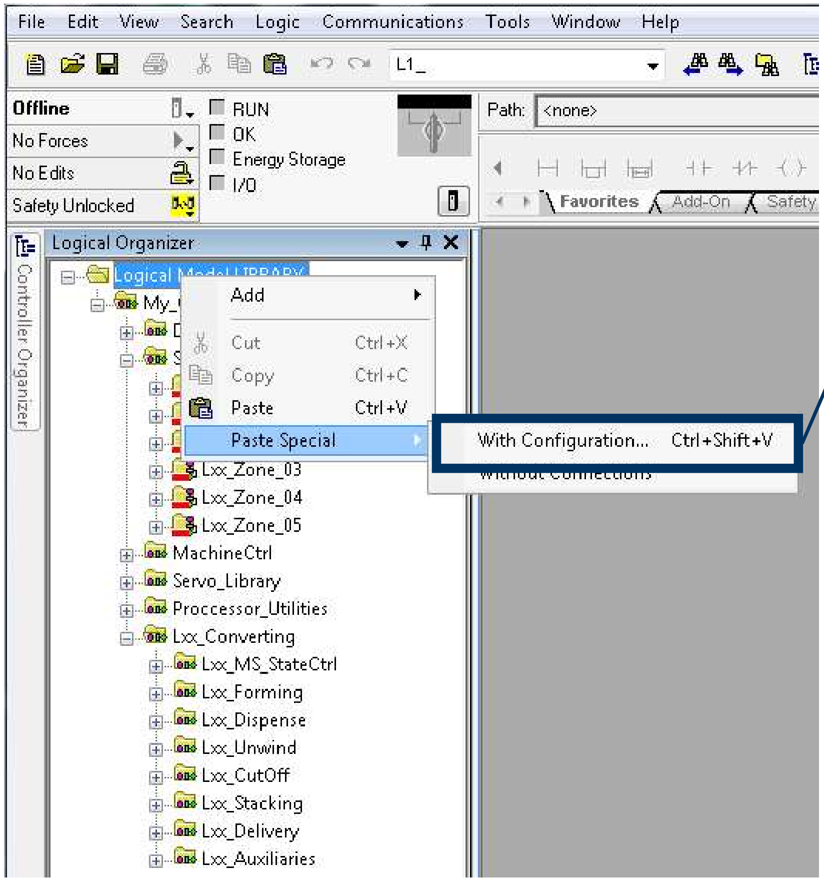
Create from Library

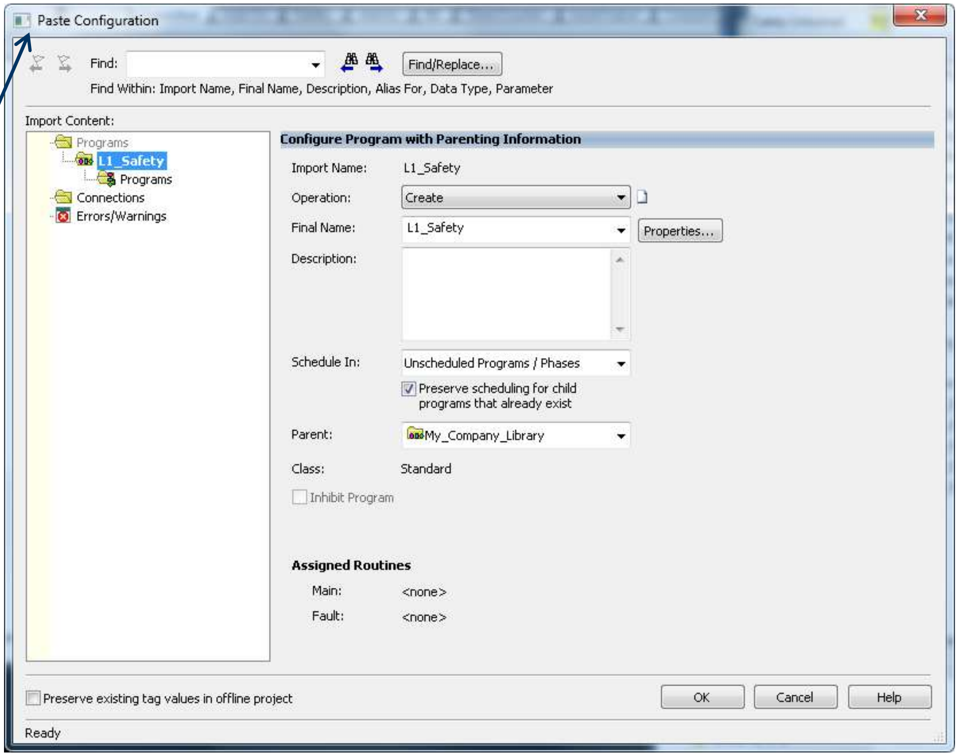
Add to Library

Studio 5000® **Logix Designer -** Partial Import Online/Offline

Enhancements Copy and Pasting Content

✓ Easily copy & paste content across projects or within project





Find Within: Import Name, Final Name, Description, Alias For, Data Type, Parameter

Configure Program with Parenting Information

Import Name: L1_Safety

Operation: Create

Final Name: L1_Safety

Description:

Schedule In: Unscheduled Programs / Phases

☑ Preserve scheduling for child programs that already exist

Parent: My_Company_Library

Class: Standard

□ Inhibit Program

Assigned Routines

Main: <none>

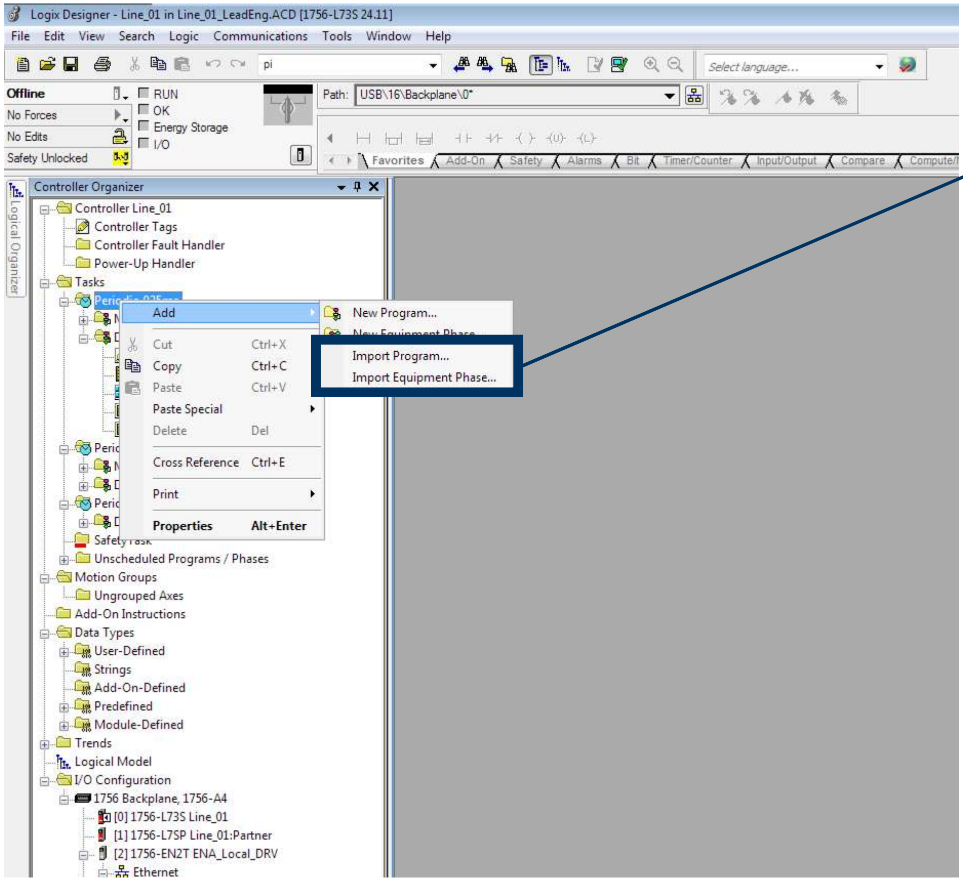
Fault: <none>

□ Preserve existing tag values in offline project

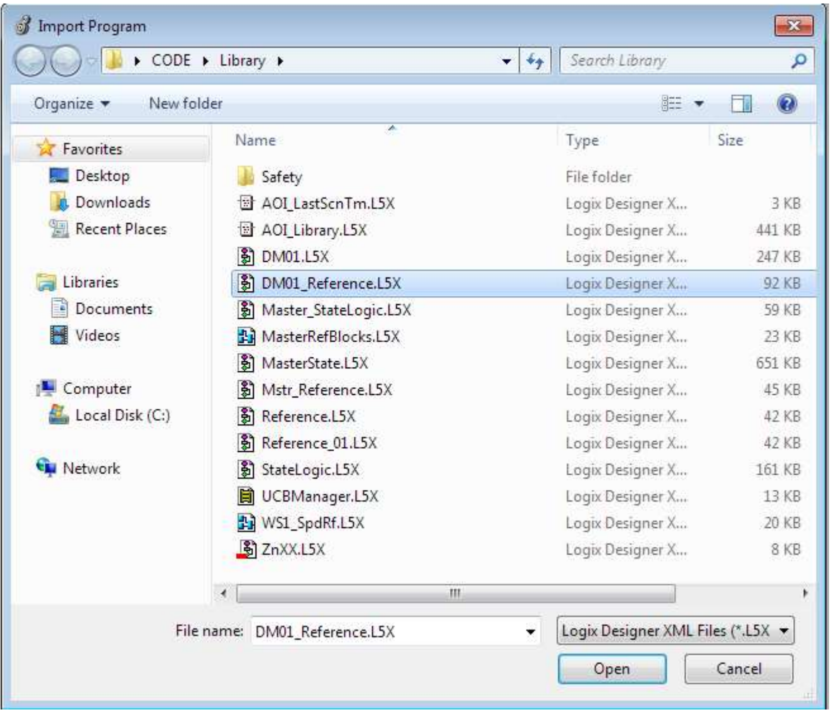
Ready

Studio 5000® **Logix Designer -** Partial Import Online/Offline

Enhancements in Importing Content



✓ Importing .L5X Content

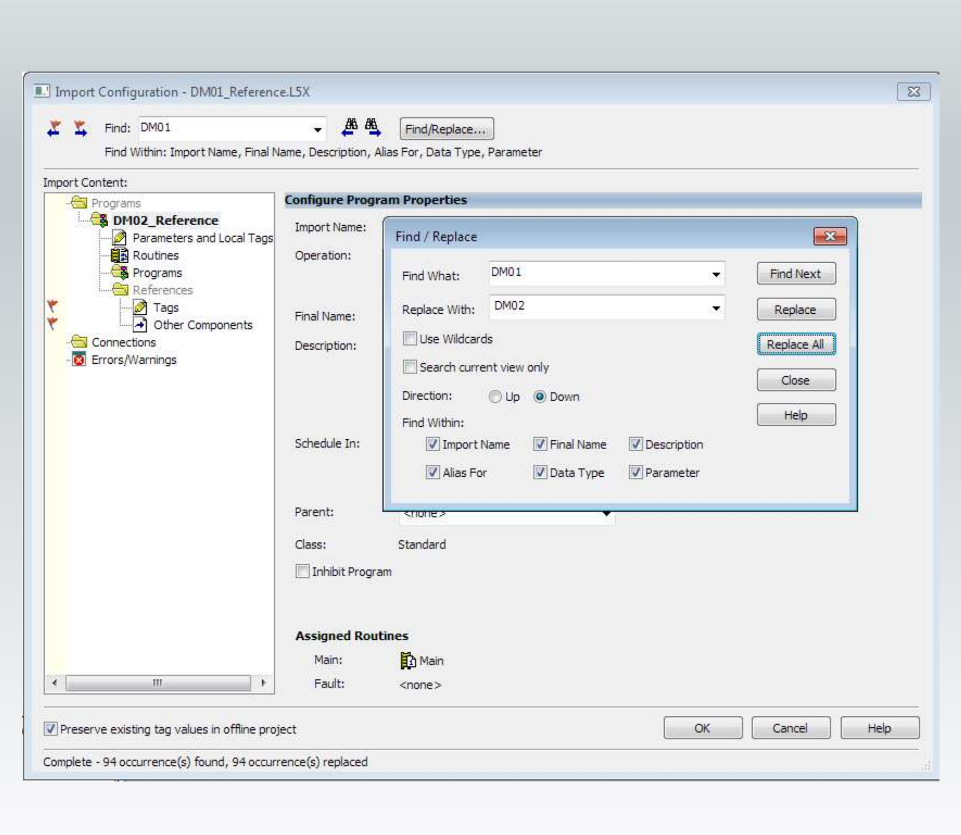


Name	Type	Size
Safety	File folder	
AOI_LastScnTm.L5X	Logix Designer X...	3 KB
AOI_Library.L5X	Logix Designer X...	441 KB
DM01.L5X	Logix Designer X...	247 KB
DM01_Reference.L5X	Logix Designer X...	92 KB
Master_StateLogic.L5X	Logix Designer X...	59 KB
MasterRefBlocks.L5X	Logix Designer X...	23 KB
MasterState.L5X	Logix Designer X...	651 KB
Mstr_Reference.L5X	Logix Designer X...	45 KB
Reference.L5X	Logix Designer X...	42 KB
Reference_01.L5X	Logix Designer X...	42 KB
StateLogic.L5X	Logix Designer X...	161 KB
UCBManager.L5X	Logix Designer X...	13 KB
WS1_SpdRf.L5X	Logix Designer X...	20 KB
ZnXX.L5X	Logix Designer X...	8 KB

File name: DM01_Reference.L5X

Studio 5000® **Logix Designer -** Partial Import Online/Offline

Enhancements in Importing Content



Find: DM01

Find Within: Import Name, Final Name, Description, Alias For, Data Type, Parameter

Import Name:

Operation:

Final Name:

Description:

Schedule In:

Parent: <none>

Class: Standard

□ Inhibit Program

Assigned Routines

Main: Main

Fault: <none>

Find What: DM01

Replace With: DM02

□ Use Wildcards

□ Search current view only

Direction: ○ Up ● Down

Find Within: ☑ Import Name ☑ Final Name ☑ Description ☑ Alias For ☑ Data Type ☑ Parameter

☑ Preserve existing tag values in offline project

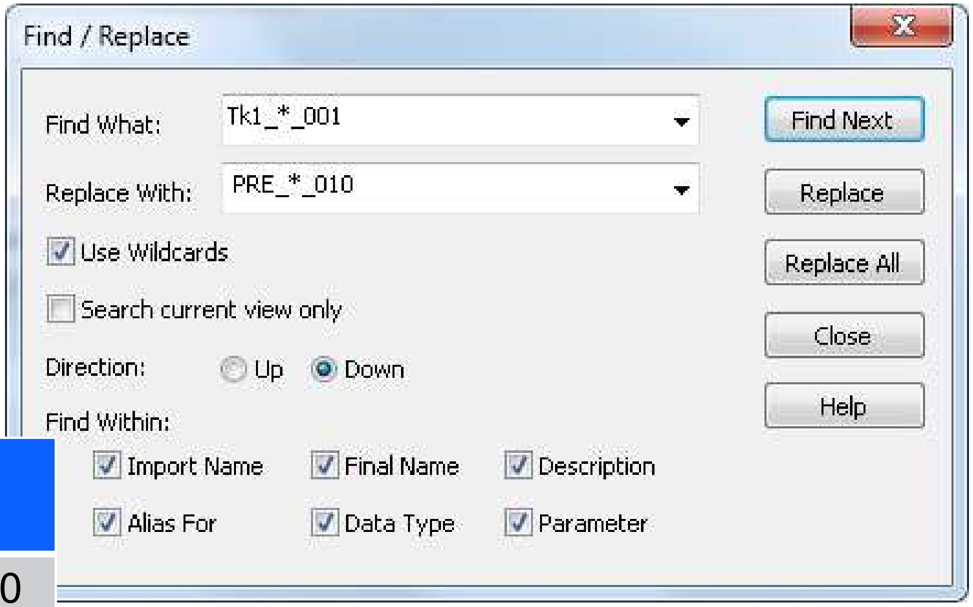
Complete - 94 occurrence(s) found, 94 occurrence(s) replaced

- **Easily Import Content**
  - ✓ Manage Conflicts
  - ✓ Create Or Update Existing
  - ✓ Manage references
  - ✓ (Tag Aliases, Program Connections)
  - ✓ Manage dependencies
  - ✓ (AOI’s UDT’s and Tags)
  - ✓ Optionally include/exclude data values
  - ✓ Supports online and offline workflows

Studio 5000® **Logix Designer -** Partial Import Online/Offline

Enhancements in Importing Content

- **Improved Search and Replace**
  - ✓ Anything you can change via PIO dialogs
  - ✓ Wildcards, Replace All



Find / Replace

Find What: Tk1_*_001

Replace With: PRE_*_010

☑ Use Wildcards

□ Search current view only

Direction: ○ Up ● Down

Find Within:

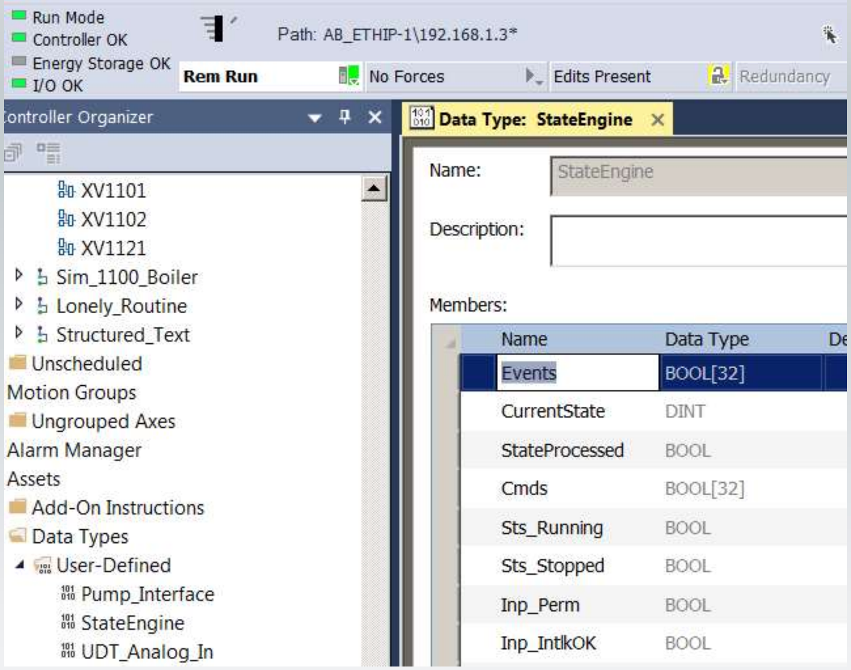
☑ Import Name ☑ Final Name ☑ Description

☑ Alias For ☑ Data Type ☑ Parameter

Find What	Replace With	Text Found	Result
Tk1_*_001	PRE_*_010	Tk1_me_001	PRE_me_010
Tk3_*	Tk4_*	Tk3_unit	Tk4_unit
*001	*002	Mi_2001	Mi_2002

Studio 5000® **Logix Designer –** Online Editing Enhancements

Rename User-defined Data Type Members and Tags on Scan



🟩 Run Mode
🟩 Controller OK
▨ Energy Storage OK
🟩 I/O OK

Path: AB_ETHIP-1\192.168.1.3*

Rem Run | No Forces | Edits Present | Redundancy

ontroller Organizer

- XV1101
- XV1102
- XV1121
- Sim_1100_Boiler
- Lonely_Routine
- Structured_Text
- Unscheduled
- Motion Groups
- Ungrouped Axes
- Alarm Manager
- Assets
- Add-On Instructions
- Data Types
  - User-Defined
    - Pump_Interface
    - StateEngine
    - UDT_Analog_In

Data Type: StateEngine

Name: StateEngine

Description:

Members:

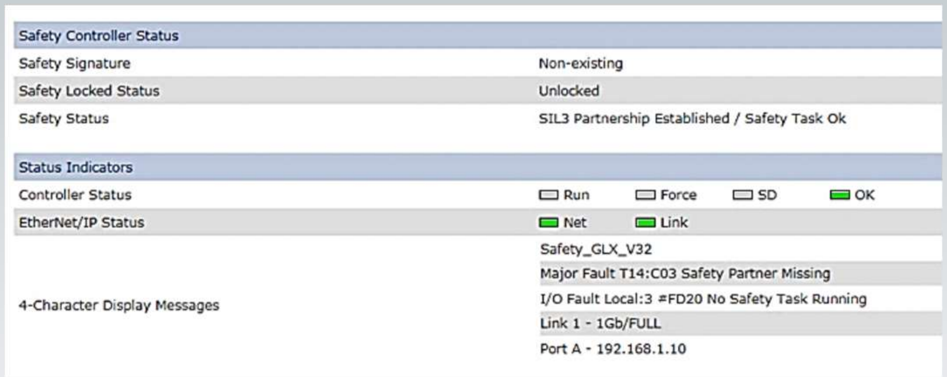
Name	Data Type	De
Events	BOOL[32]	
CurrentState	DINT	
StateProcessed	BOOL	
Cmds	BOOL[32]	
Sts_Running	BOOL	
Sts_Stopped	BOOL	
Inp_Perm	BOOL	
Inp_IntlkOK	BOOL	

- **User-Defined Data Type**
  - ✓ Design UDT with excess memory
  - ✓ Use spare members while editing online
  - ✓ Download not required
- **Tags on Scan**
  - ✓ Allow renaming of tags on scan
  - ✓ Supported by:
    - ✓ Studio 5000® Logix Designer software
    - ✓ FactoryTalk® Linx software (V6.10+)

Productivity Enhancements

Version 32 Updated Device Webpages

Updated Device Webpage Example



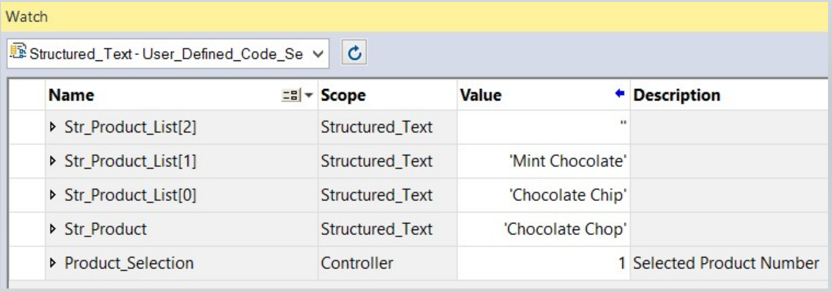
Safety Controller Status	
Safety Signature	Non-existing
Safety Locked Status	Unlocked
Safety Status	SIL3 Partnership Established / Safety Task Ok
Status Indicators	
Controller Status	□ Run □ Force □ SD 🟩 OK
EtherNet/IP Status	🟩 Net 🟩 Link
4-Character Display Messages	Safety_GLX_V32 Major Fault T14:C03 Safety Partner Missing I/O Fault Local:3 #FD20 No Safety Task Running Link 1 - 1Gb/FULL Port A - 192.168.1.10

Remote Controller Diagnostics

- ✓ Troubleshoot faster through upgraded device webpages that now display:
  - ✓ Safety Signature and Lock Status
  - ✓ Controller and Network Status Indicators
  - ✓ 4 Character Message Display Status
- ✓ Access status and fault information instantly wherever you are, for any standard or safety applications on high performance family of controllers

Studio 5000® **Logix Designer –** Quick Watch

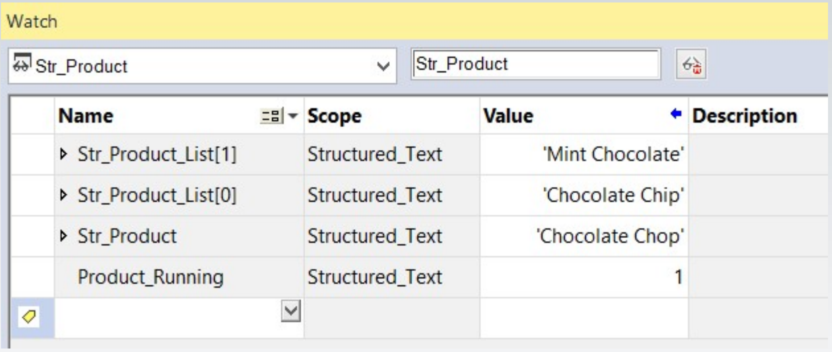
Quickly Monitor Tag Values



Watch

Structured_Text - User_Defined_Code_Se

Name	Scope	Value	Description
Str_Product_List[2]	Structured_Text	''	
Str_Product_List[1]	Structured_Text	'Mint Chocolate'	
Str_Product_List[0]	Structured_Text	'Chocolate Chip'	
Str_Product	Structured_Text	'Chocolate Chop'	
Product_Selection	Controller	1	Selected Product Number



Watch

Str_Product

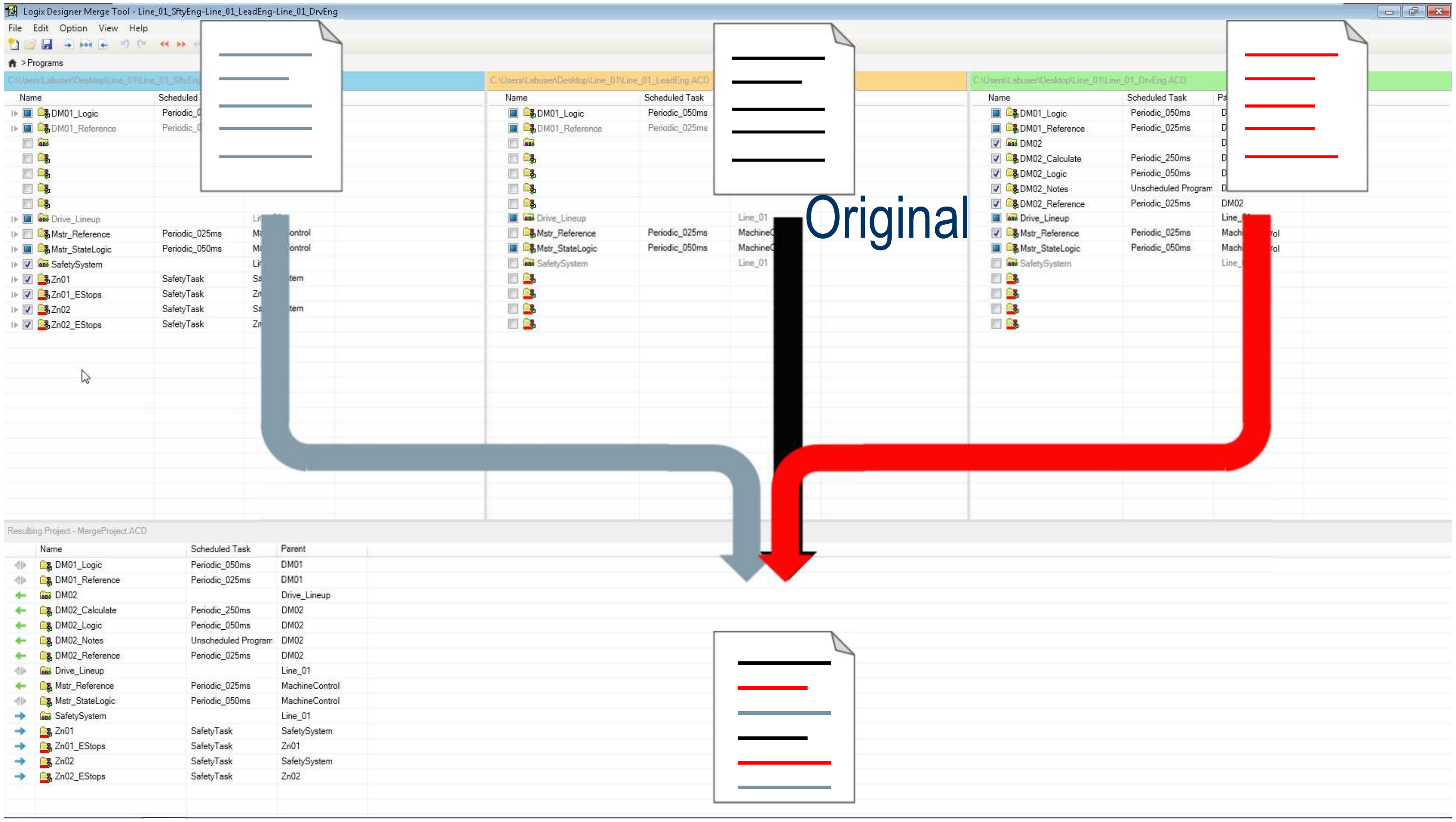
Str_Product

Name	Scope	Value	Description
Str_Product_List[1]	Structured_Text	'Mint Chocolate'	
Str_Product_List[0]	Structured_Text	'Chocolate Chip'	
Str_Product	Structured_Text	'Chocolate Chop'	
Product_Running	Structured_Text	1	

- **Quickly monitor tags based on an open routine**
  - ✓ Select the routine to monitor from the pull-down menu
  - ✓ Routines are listed at the bottom
  - ✓ Watch pane tags are updated based on the tags that are used in the selected routine
- **Add tags to create a custom tag monitor view**
  - ✓ Select Quick Watch from the pull-down menu
  - ✓ Add your tags to the list below
  - ✓ You can use the box to the right of the pull-down menu to name the quick watch

Studio 5000® **Logix Designer –** Compare & Merge Tool

Merge Selected Components Against The Original


Logix Designer Merge Tool - Line_01_SftyEng-Line_01_LeadEng-Line_01_DrvEng
File
Edit
Option
View
Help
> Programs
Original

C:\Users\Labuser\Desktop\Line_01\Line_01_SftyEng

	Name	Scheduled	
■	DM01_Logic	Periodic_0	
■	DM01_Reference	Periodic_0	
☐			
☐			
☐			
☐			
☐			
■	Drive_Lineup		Li
☐	Mstr_Reference	Periodic_025ms	M ontrol
■	Mstr_StateLogic	Periodic_050ms	M ontrol
☑	SafetySystem		Li
☑	Zn01	SafetyTask	S tem
☑	Zn01_EStops	SafetyTask	Z
☑	Zn02	SafetyTask	S tem
☑	Zn02_EStops	SafetyTask	Z

C:\Users\Labuser\Desktop\Line_01\Line_01_LeadEng.ACD

	Name	Scheduled Task	
■	DM01_Logic	Periodic_050ms	
■	DM01_Reference	Periodic_025ms	
☐			
☐			
☐			
☐			
☐			
■	Drive_Lineup		Line_01
☐	Mstr_Reference	Periodic_025ms	MachineC
■	Mstr_StateLogic	Periodic_050ms	MachineC
☐	SafetySystem		Line_01
☐			
☐			
☐			
☐			

C:\Users\Labuser\Desktop\Line_01\Line_01_DrvEng.ACD

	Name	Scheduled Task	Pa
■	DM01_Logic	Periodic_050ms	D
■	DM01_Reference	Periodic_025ms	D
☑	DM02		D
☑	DM02_Calculate	Periodic_250ms	D
☑	DM02_Logic	Periodic_050ms	D
☑	DM02_Notes	Unscheduled Program	D
☑	DM02_Reference	Periodic_025ms	DM02
■	Drive_Lineup		Line_
☑	Mstr_Reference	Periodic_025ms	Mach rol
■	Mstr_StateLogic	Periodic_050ms	Mach rol
☐	SafetySystem		Line_
☐			
☐			
☐			
☐			

Resulting Project - MergeProject.ACD

	Name	Scheduled Task	Parent
⇔	DM01_Logic	Periodic_050ms	DM01
⇔	DM01_Reference	Periodic_025ms	DM01
←	DM02		Drive_Lineup
←	DM02_Calculate	Periodic_250ms	DM02
←	DM02_Logic	Periodic_050ms	DM02
←	DM02_Notes	Unscheduled Program	DM02
←	DM02_Reference	Periodic_025ms	DM02
⇔	Drive_Lineup		Line_01
←	Mstr_Reference	Periodic_025ms	MachineControl
⇔	Mstr_StateLogic	Periodic_050ms	MachineControl
→	SafetySystem		Line_01
→	Zn01	SafetyTask	SafetySystem
→	Zn01_EStops	SafetyTask	Zn01
→	Zn02	SafetyTask	SafetySystem
→	Zn02_EStops	SafetyTask	Zn02

Studio 5000® **Logix Designer –** Automatic Device Configuration

ADC Enable Seamless Device Replacement

Field Device

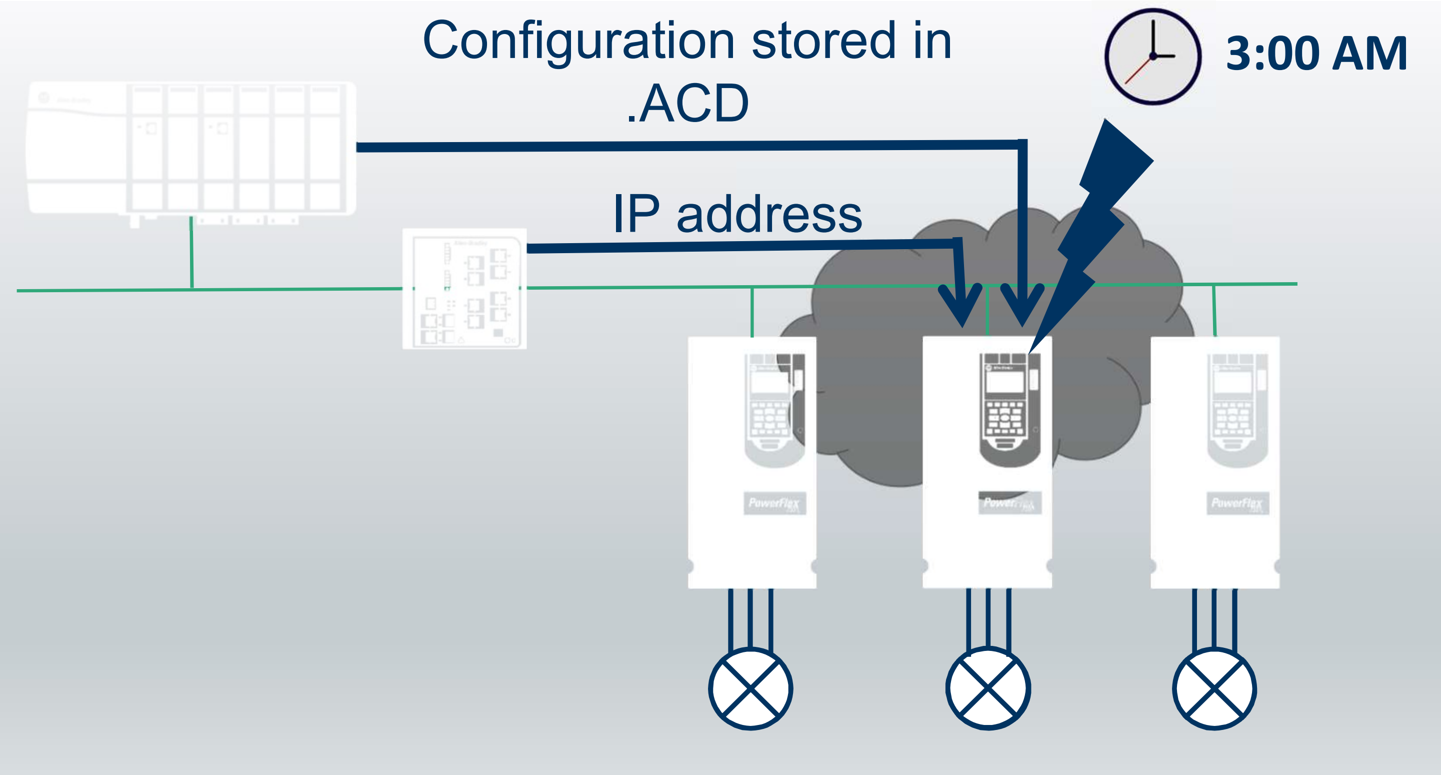
+

Controller

+

EtherNet/IP

LOWER TIME TO REPAIR



Studio 5000® **Logix Designer –** Automatic Device Configuration

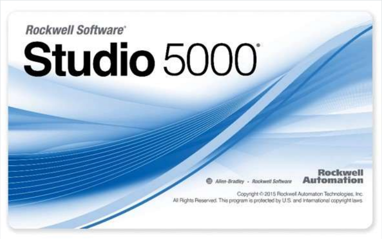
ADC Enable Seamless Device Replacement

DHCP Port Persistence

- ✓ Stratix switch acts as a DHCP server
- ✓ Automatically assigns a specific IP address to a particular port

Firmware Supervisor

- ✓ Local and remote modules can be flashed in Program / Run mode
- ✓ Firmware images stored on controller SD card
- ✓ Modules must have Electronic key parameter set to "Exact Match"



Automatic Device Configuration

- ✓ ADC downloads project drive parameters to new drive once firmware update completes
- ✓ Drive configuration settings stored in project .ACD file
- ✓ No software required to commission new drive

Studio 5000® Logix Designer

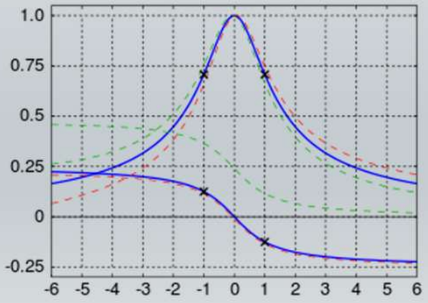
Smart Machine Drives Should Tune Themselves

Load Observer

- ✓ Operates in real time as the machine runs
- ✓ Dynamically estimates the load torque & provides a feedback signal to cancel its effect
- ✓ Works well not only for rigid loads, but also compliant loads
- ✓ Great for systems in which the load varies either predictably or unpredictably

Adaptive Tuning with Tracking Notch Filter

- ✓ Compensates for unknown & changing load conditions on the fly
- ✓ Operates in real time as the machine runs
- ✓ Automatically adjusts torque loop notch and low pass filter parameters to suppress resonances
- ✓ Automatically de-tunes control loop gains to avoid instabilities if detected



Studio 5000® Design Environment

Studio 5000® Architect Environment	Simplified System Design and Data Exchange
Studio 5000® Logix Designer Software	Collaborative System Programing and Configuration
Studio 5000® View Designer Environment	Integrated HMI with Logix
Studio 5000® Application Code Manager Tool	System Reuse and Project Builds
Studio 5000® Logix Emulate™ Software	Virtual Design and Operator Training Systems
Studio 5000® Simulation Interface	Logic Connection to Simulation Models

Studio 5000

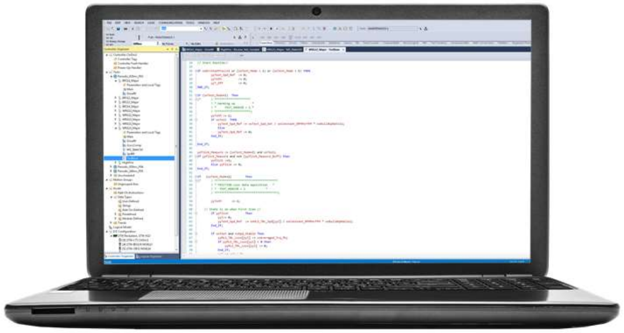
Kontakta WEXOE AB för en individuell genomgång!

Studio 5000
Logix Designer

Tony Annell
Divisionschef
M: 070-609 26 10
E: tan@wexoe.se

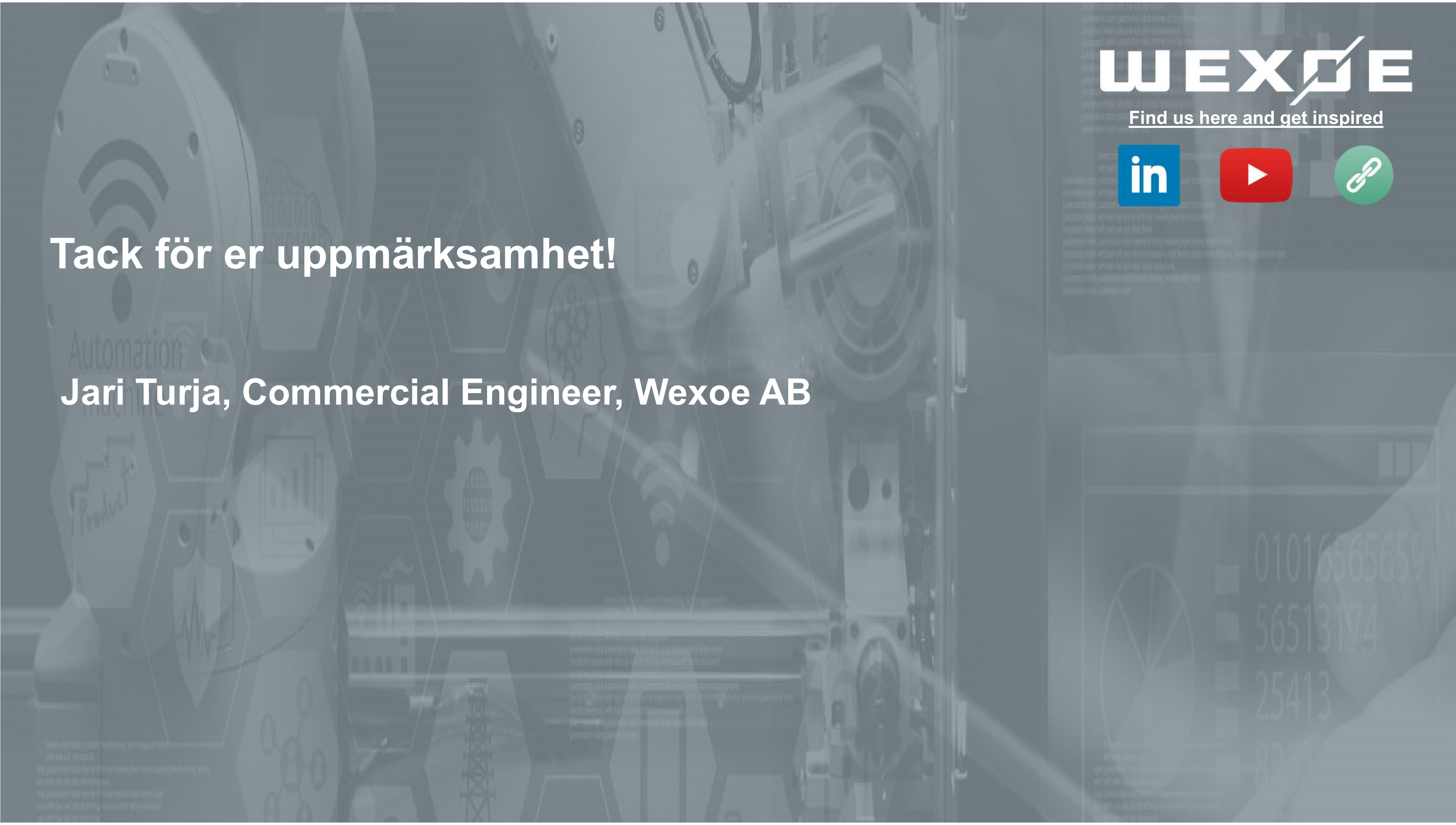
Andreas Grünerwald
Account Manager
M: 070-609 26 38
E: agr@wexoe.se

Jari Turja
Commercial Engineer
M: 072-991 81 91
E: jtu@wexoe.se



WEXOE AB
Scheelevägen 17
223 70 LUND
www.wexoe.se

WEXOE



WEXOE

Find us here and get inspired

Tack för er uppmärksamhet!

Jari Turja, Commercial Engineer, Wexoe AB