

Background

MicroCODE Control is part of a new Windows 10 Industrial Internet of Things (IIOT) application to support System Acceptance Testing a GM Scalable Error Proofing (SEP) system:

- **Control** – Local control of SEP data and simulation

This MicroCODE Control application can support all the processes required without the need for an SEP Server, working independently with a connection to the SEP Cell Controller PLCs (GMP and GXP).

Hardware Requirements

The **Control** App requires/supports the following hardware for proper execution:

- Windows 10 PC
- Spec support for .NET 4.5+
- Logix 5000 L5x, L6x, L7x, L8x

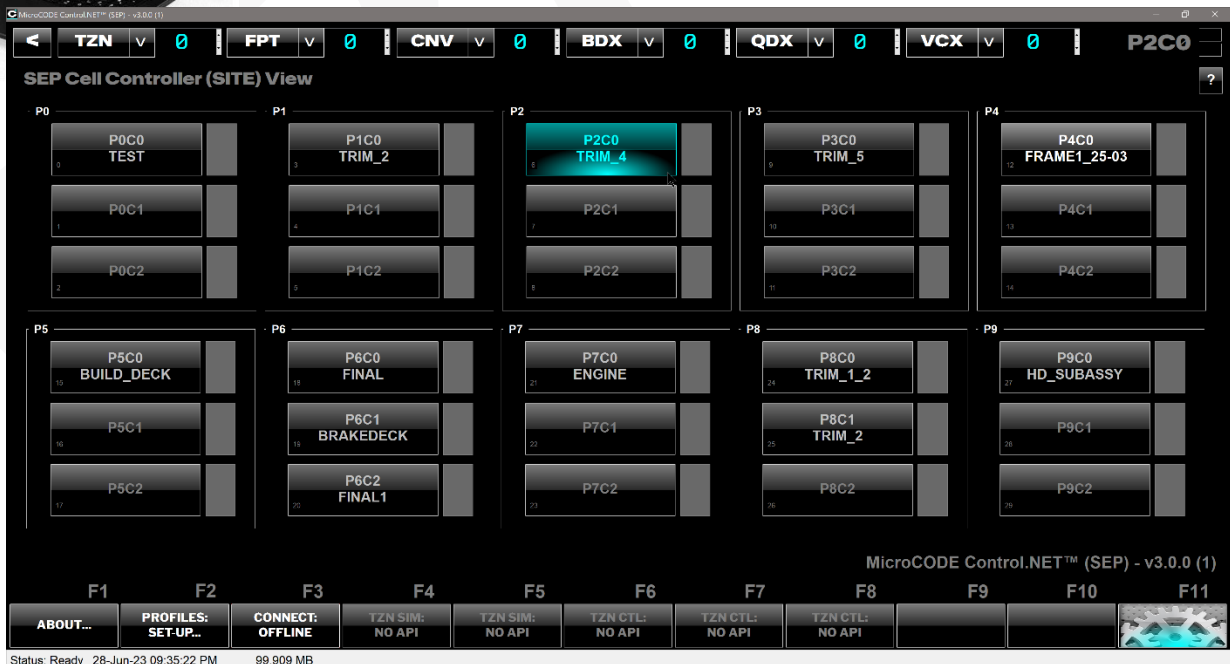
Software Requirements

The **Control** App requires the following software for proper execution:

- **Windows 10** – installed on the PC
- **Microsoft .NET 4.5 or higher** – installed on the PC
- MicroCODE **Control** Windows 10 .NET Application
- *Note: .NET Framework 4.* is always available on a Windows 10 PC*
- MicroCODE Application Interface PLC (**API**) program imported into the target SEP Cell Controller (**GMP**)

The Control App does ***not*** require any additional software licenses this is a MicroCODE owned application licensed directly to our end users:

- No Rockwell Automation RSLinx required
- No 3rd Party OPC Server
- No 3rd Party Communication Software





Control App Software

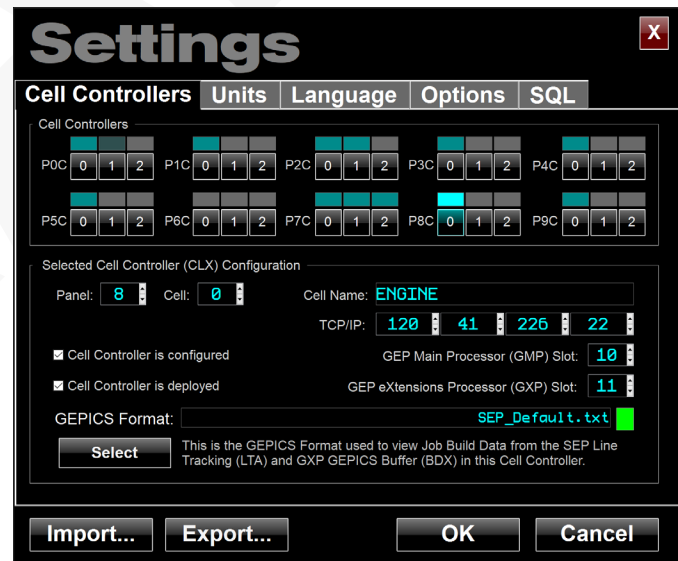
Application Framework

The underlying the Control App application framework supports the following requirements:

- **HMI Style Application** – graphical for ease of configuration and operation.
- **Multi-Threaded, Event Based Code** – this construct supports fast, very efficient code, allows a simple, inexpensive processor to handle control and interface tasks.
- **Extensive Logging** – supporting six (6) different logs the Control App can produce reports and data for all Production and Application support needs:

- 1) **Application Event Log** – internal application events of interest to Support Staff and Developers and operator related events of interest to Production Staff.
- 2) **Data Recorder** – record of every product that has passed through the Control App Station’s span of control.
- 3) **Console Log** – a separate ‘Developer’s Log’ for DEBUG messages that may be needed in the field. This capture ‘Console write to Standard Output (StdOut)’. This prevents developers from ‘flooding’ the Production Event LOG file which are used to diagnose Plant issues, *not* App issues. **New in v3.0**

- **HMI Integration** – built-in visualization for Operator control and monitoring.
- **SQL Integration** – auto-configures from a local or remote SEP SQL Database. **New in v2.0**
- **Programmable Logic Controller (PLC) Communication** – for the Control App can be directly extended by industry standard PLCs for larger systems.
- **Integrated Help and Tools** – all information required to configure the Control App devices is included in application screens where the actual work is performed.
- **Support Mode** – built into the Production application to allow for extensive data collection to help remotely trouble shoot problem.
- **JSON Data Storage** – all the MicroCODE Control.NET™ and Sequence.NET™ Apps now use JavaScript Object Notation (JSON) for all Configuration and Data storage files. **New in v3.0**





Control App Software

App Features

The **MicroCODE Control for SEP** App provides the following SEP control features:

- **Conveyor Simulator Control** – supports direct control of the Conveyor Simulation built into the SEP Cell Controller. Allows the user to run a single tool to control his entire System Acceptance Test (SAT).
- **Generation of Unique GEPICS Jobs** – allows the user to select a ‘seed’ Job from the GXP PLC Buffer and then alter it thru a configured Job Control Profiles (JCP). These Profiles allow the user to alter Model Code, RPO Codes, Process Tool Styles, etc. These can be set-up to rotate by CSN to simulate to flow of Jobs with different data and unique CSNs, VINs, and PVI.
- **Import / Export of Job Control Profiles** – the Profiles used to generate simulated GEPICS orders can be exported and re-imported to be shared between Users, PCs, and Launch Sites.
- **Build Data Viewer** – a built-in tool to examine any part of the GEPICS Build Data Packet from any location in the GMP or GXP PLCs; i.e.: The Line Tracking Image in GMP or the GEPICS Buffer in GXP.
- **GSIP Defect Queue Viewer** – a built-in tool to examine the Vehicle Defect Queue in the GMP PLC; i.e.: you can watch Defects being generated as you perform SATs.
- **GEPICS Trace Data Viewer** – a built-in tool to examine the Vehicle Trace Data Queue in the GMP PLC; i.e.: you can watch Trace Data being collected as you perform SATs*.
- **Requires no change to existing SEP PLC Code** – an interface program (**prgAPI**) is imported thru RSLogix 5000 and that program allows the Windows App to perform all required work. This program can then be deleted when the SAT Simulation work is completed.
- **Compatible with both SEP v1.0.5 SP1 and SEP v2.3.0** – the interface program (**prgAPI**) isolates the Windows App from differences within the SEP PLC programs.

* This only applies when the Trace Data is going thru the GMP PLC and not directly from an SEP Actions Station to GEPICS.

SEP Dependencies

The **MicroCODE Control for SEP** App requires the following items to remain unchanged within the SEP GEP Main Processor (**GMP**):

- **Conveyor Subsystem** – the “**CNV**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. Additions may be made, but all existing data items and their definitions must remain unchanged.
- **Line Tracking Subsystem** – the “**LTS_LTA**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. All existing data items and their definitions must remain unchanged. The maximum Build Data Packet size supported is the v1.0.5 SP1 size of 2,000 total bytes.
- **Vehicle Quality Subsystem** – the Defects Queue “**VQX_QDX_EventQueue**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. All existing data items and their definitions must remain unchanged.
- **Vehicle Quality Subsystem** – the Trace Queue “**VQX_VCX_EventQueue**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. All existing data items and their definitions must remain unchanged.

The **MicroCODE Control for SEP** App requires the following items to remain unchanged within the SEP GEP eXtensions Processor (**GXP**):

- **Build Data Subsystem** – the “**BDX_Data**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. All existing data items and their definitions must remain unchanged. The maximum Build Data Packet size supported is the v1.0.5 SP1 size of 2,000 total bytes.
- **Build Data Subsystem** – the “**BDX_SEQ**” Controller Level Tag, its User-Defined Datatype (UDT) and all sub-Datatypes must remain in their v1.0.5 SP1 form. All existing data items and their definitions must remain unchanged.

The **MicroCODE Control for SEP** App requires the following guidelines be followed in both GMP and GXP:

- No Controller Level tags should be created the start with “**API**”. These are created by the App Programming Interface and are removed after all simulation is completed.
- No program in GMP or GXP should be named “**prgAPI**” this is reserved for this App’s interface program.



App Overview

The main App Screen is used to both configure and monitor and control all GM SEP functions.

New in v2.0, this entire **SITE** layer was added to the Control App.

SEP SITE Summary Screen

SEP Panel

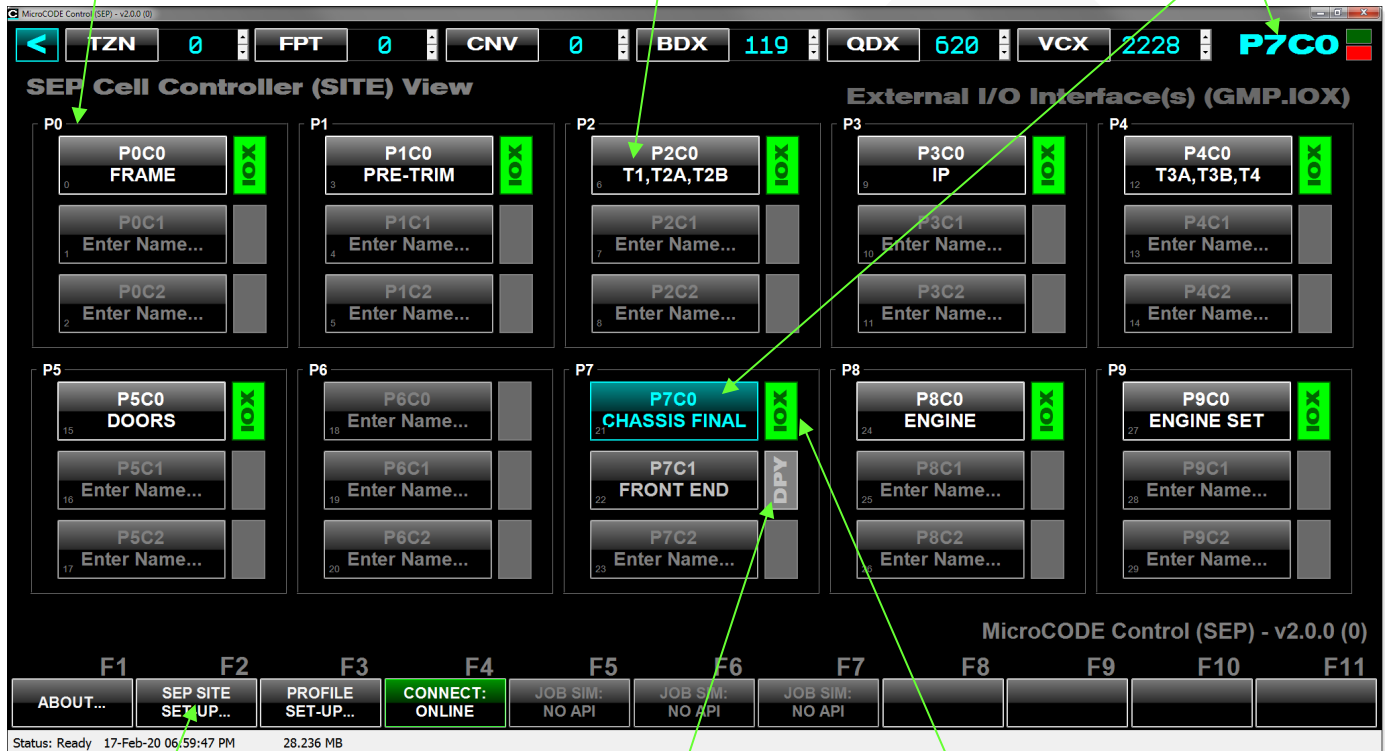
An Area covered by up to (3) Cell Controllers. SEP supports up to (10) Panels/Areas. All (30) possible Cells are shown on this screen.

SEP Cell Controller

A collection of up to (15) Track Zones, All controlled by a pair of PLCs: GMP+GXP.

Active Cell Controller

The Cell being controlled by the App.



SITE Set-Up

Allows configuration of all (30) Cell Controllers, including automatic extraction from the SEP SQL Database.

Subsystem Monitoring

Provides constant monitoring of SEP PLC Subsystem health, a simple substitute for GPM&C in the Strategic Supplier Site during validation. See details on the following pages...

Cell Status: "DPY"

Indicates a Cell still needs to be deployed.

SEP CELL Control Screen

- Track Zone Selection & Viewer
- Footprint / Job Selection & Viewer
- Conveyor Selection & Viewer
- Build Data 'Seed' Selection & Viewer
- GSIP Queue Defect Viewer
- VCVS Queue Trace Viewer

Live Conveyor Status for current Footprints

Live Job Status for current Footprints

Station Set-Up Communication
Set-up, PLC connections,

Job Control Profiles (JCP)
Controls the creation of simulated Jobs.

Connect
Connects the App to the SEP Cell Controller.

SIM: Start
Starts a new stream of simulated Jobs based on the active Profile.

SIM: Abort
Aborts the simulation and clears the active Track Zone.

CNV: Simulator Controls
Starts or Stops simulated Conveyor motion. Clears Andon Holds, Bypasses Andon, Changes Speed.



ControlLogix Integration

The MicroCODE Application Programming Interface (API) within the SEP Logix 5000 PLC provides a connection between the IPC C# App to the SEP subsystems:

The screenshot shows the Controller Organizer with the following structure:

- Controller GMSEP_PxCy_GMP
 - Controller Tags
 - Controller Fault Handler
 - Power-Up Handler
 - Tasks
 - tskContinuous
 - prgDGN
 - prgCLX
 - prgSEP
 - prgCNV
 - prgLTS
 - prgSCN
 - prgEPX
 - prgQAX
 - prgVQX
 - prgSIM
 - prgPVW
 - prgPMX
 - prgPWX
 - prgUIX
 - prgPSP
 - prgAPI
 - Parameters and Local Tags
 - ladAPI_Main
 - ladAPI_Fault
 - ladAPI_Init
 - ladAPI_Reset
 - rtnAPI_InitEnum
 - tskIO
 - prgIOX
 - tskIO_MIO
 - prgMIO
 - Unscheduled Programs
 - Motion Groups
 - Ungrouped Axes
 - Add-On Instructions
 - aoiSCN_NIO_SICK
 - aoiSCN_NIO_SICK_CCOM
 - Data Types
 - User-Defined
 - udtAPI
 - udtAPI_Event
 - udtAPI_EventData
 - udtAPI_FaultRecord
 - udtAPI_Flags
 - udtAPI_Semaphores
 - udtAPI_States
 - udtAPI_TimeStamp

The table below shows the details for the API tags:

Name	Data Type	Description
API	udtAPI	Control structure for the API Subsystem
API_ClearFault	udtAPI_FaultRecord	For all Fault Handlers, used to clear faults while maintaining record of fault data until logged.
API_Enum08	SINT[256]	8 Bit Numbers, starting at ZERO, in an array used to initialize other UDT based Tags that define numerations.
API_Enum16	INT[256]	16 Bit Numbers, starting at ZERO, in an array used to initialize other UDT based Tags that define numerations.
API_Enum32	DINT[256]	32 Bit Numbers, starting at ZERO, in an array used to initialize other UDT based Tags that define numerations.
API_EventData	udtAPI_EventData	Current Diagnostic Event being logged by a API Subsystem.
API_Semaphore	udtAPI_Semaphores	Enumeration defining all API Subsystem Semaphore states.
API_State	udtAPI_States	Enumeration defining API Subsystem Code States.

API Tags
Application Programming Interface data for C# App all begin with "API".

NOTE: Always use the prgAPI that was provide with the App you installed. These program's version must match the version of the Windows App you are using.

See the USER GUIDE for complete instructions.

API Program
Application Programming Interface for C# App.

API UDTs
User Defined Datatypes for C# App all begin with "udtAPI".

NOTE: The API program, tags, and UDTs can all be deleted after simulation SATs are completed with no effect on a running SEP Cell Controller.



Control System Acronyms

All the software and documentation for this MicroCODE Control System utilize a small set of acronyms:

Control Application Level...

CNV = Conveyor

LTA = Line Tracking Array

FPT = Footprint

Hardware Level...

PLC = Programmable Logic Controller

IPC = Industrial Personal Computer

Industry Software Level...

IoT = Internet of Things

IIoT = Industrial Internet of Things

MicroCODE Control Software Level...

API = Application Program Interface

CTL = Control Data

JCP = Job Control Profile – Simulation Control



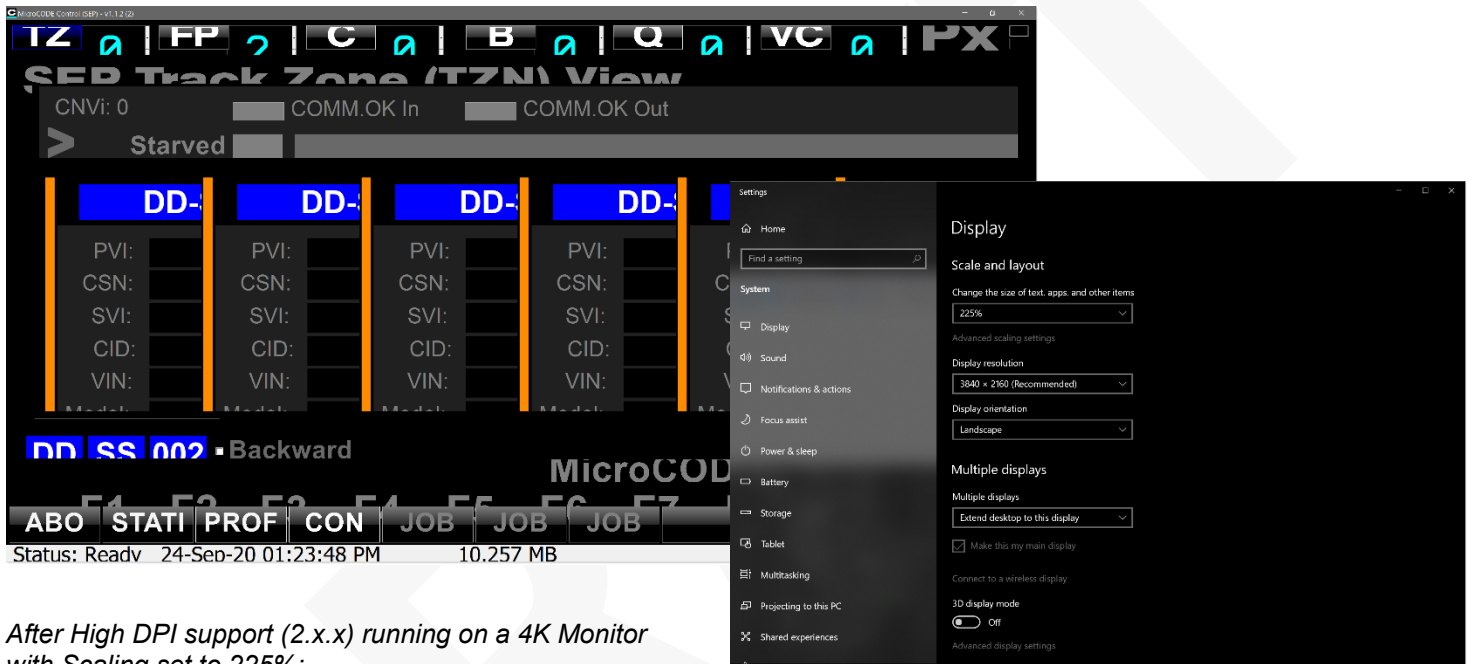
New in Version v2.0.0 a (0)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) High Resolution (DPI) Monitor and Windows Scaling Support

The App now supports 4K Monitors (greater than 96DPI) and User adjusted scaling (greater than 100%).

Before High DPI support (v1.x.x) running on a 4K Monitor with Scaling set to 225%:



After High DPI support (2.x.x) running on a 4K Monitor with Scaling set to 225%:





2) SITE Wide SEP Screen

The App now includes a screen showing all the SEP Cell Controllers in a Shop. A Shop / SEP Server can handle up to thirty (30) Cell Controllers.

These Cell Controllers are divided into ten (10) Panels, each capable of hold three (3) Cell Controllers.

In the original version of the GM Common Operator Support System (COSS) this corresponded to Main Control Panels (MCPs) that actually held three (3) Logix Chassis, two (2) Cell Controllers.

This was changed in GM Global Error Proofing (GEP) releases to treat all three Chassis as individual Cell Controllers. Late in GEP development and carried on into the GM Scalable Error Proofing (SEP) releases the 'Panel' became a representation of an 'Area' and no longer represents a physical MCP.

This new screen can be manually configured from the SEP SITE SET-UP dialog box, which replaces the STATION SET-UP dialog box in v1 of the Control App.

The new SEP SITE SET-UP also allows you to pull all the Cell Controller configuration from the SEP SQL Database if the SEP Server is available.

Each Panel's configuration holds:

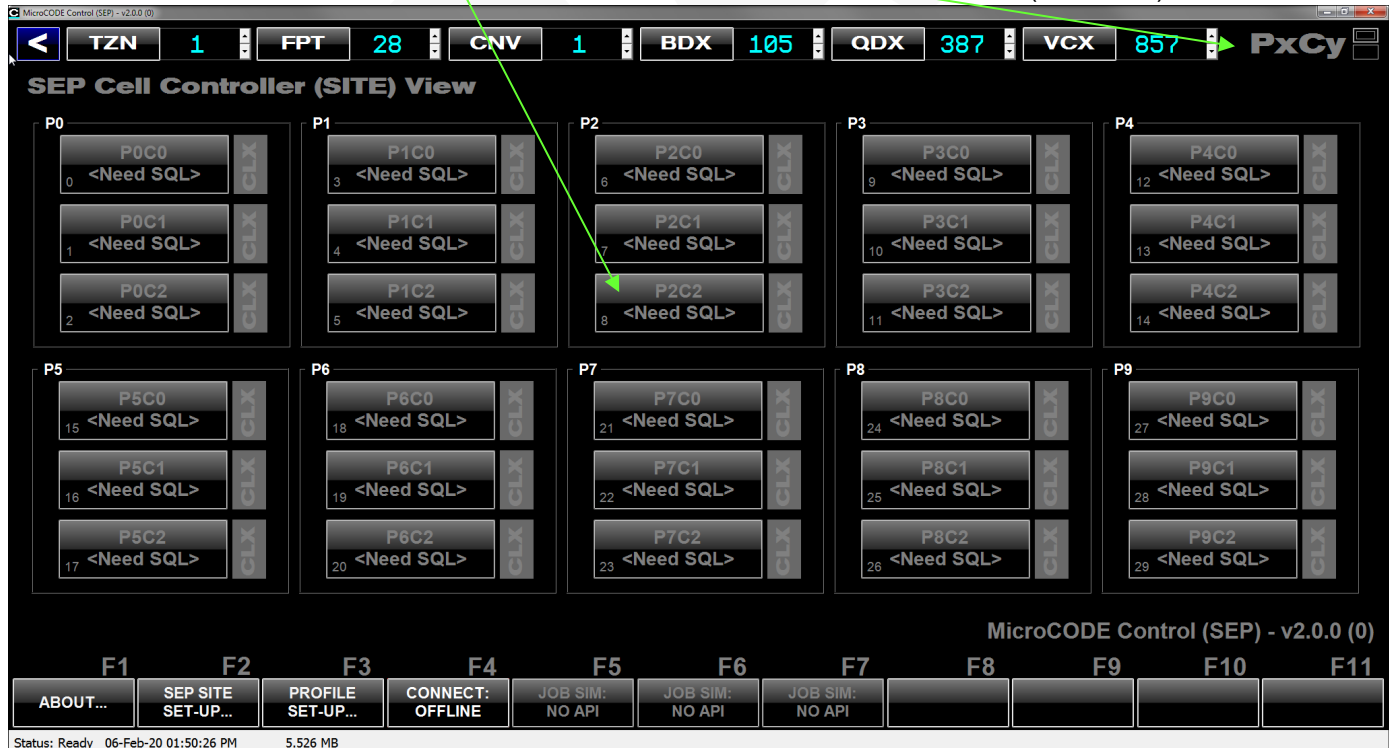
- Panel Index
- User configured Area Name
- Configured Status
- Deployed Status
- Area Type: TRIM, CHASSIS, ENGINE, FINAL

Each Cell Controller's configuration holds:

- Panel Index
- Cell Index
- User configured Cell Name
- Configured Status
- Deployed Status
- Area Type: TRIM, CHASSIS, ENGINE, FINAL
- IP Address of the 'GM IT' Ethernet Card (for App)
- Chassis Size (Slot count)

Active Cell Controller

New in v2.0, the Active Cell Controller is always displayed in the upper right, this is the only Controller that can be written to by the App. It is changed by simply clicking on a different Controller in the SITE View shown here.





3) SITE Wide Configuration Dialog box

This release replaces the STATION configuration dialog box with a SITE configuration. This allows all the SEP Cell Controllers in a Shop to be configured from a single dialog box. A Shop / SEP Server can handle up to thirty (30) Cell Controllers.

Cell Controllers
New in v2.0, users can now configure all possible Controllers in a Site once.

SQL Configuration
New in v2.0, users can now pull all required configuration from a GM Site's SEP SQL Database.

Cell Selection
New in v2.0, users can quickly change which Cell Controller they are configuring and see the configured/deployed status of each at-a-glance.

- Selected
- Configured & Deployed
- Configured only
- Not configured

Cell Deployment Status
New in v2.0, users can now indicate--at a Cell level--if the PLCs are configured, and/or physically deployed. The App will only attempt PLC communication after a Cell is marked 'deployed'.

GEPICS Format
User selected on a Cell Controller Level; every Cell can have a unique Format.

App Configuration Import
New in v2.0, Allows the user to import a previous App configuration file (.CFG)

4) SQL Database Configuration Extraction

In addition to being able to pull the configuration of all Cell Controllers—for communication—the App now extracts Track Zone and Footprint (DSOA) names from the SEP SQL DB.

SEP SQL Server

New in v2.0, configures READ-ONLY access to the SQL Configuration Database. The Username and Password of this account are fixed by GM IT.

SQL Configuration

New in v2.0, users can now pull all required configuration from a GM Site's SEP SQL Database.

Erase Track Zones

New in v2.0, clears all Track Zones (and Footprints) within the Control App, it has no effect on the SEP SQL Database.

Erase All Configuration

New in v2.0, clears all SEP configuration within the Control App, it has no effect on the SEP SQL Database.

Get Track Zones

New in v2.0, pulls all Track Zone (and Footprints) configuration from the SEP SQL Database into the Control App.

Get All Configuration

New in v2.0, pulls all required SEP configuration from the SEP SQL Database into the Control App.



5) Automatic DSOA Placards

The SQL Extraction allows the App to automatically display the DSOA Placards names with no user input and handles 'custom' DSOAs that do not follow the standard pattern.

- If the Track Zone configuration has been extracted into the App the Footprint Placards from the SEP SQL Database are used.
- If the SQL Database has not been queried, then the starting Placard—configured in this App—is used to generate all Placards for the Track Zone.

Active Cell Controller
New in v2.0, the Active Cell Controller is always displayed in the upper right, this is the only Controller that can be written to by the App.

The screenshot shows the MicroCODE Control (SEP) v2.0.0 (0) interface. At the top, there's a navigation bar with track zone numbers: TZN 4, FPT 19, CNV 7, BDX 125, QDX 771, VCX 3, and P8CO. Below this is the 'SEP Track Zone (TZN) View' section. It includes a 'CNV: 7' indicator, 'COMM.OK In' and 'COMM.OK Out' status, and a '@FPS' indicator. There are also status indicators for 'Blocked', 'Stopped', 'Production', 'EPx Stop(s)', 'Andon Stop(s)', 'Run Stop(s)', 'End-of-Travel(s)', 'SIM Stop(s)', and 'SIM Bypass(s)'. The main area displays a grid of placards for various track zones (e.g., 17-B2-020L to 17-B2-025R) and a B2_BOXLINE2 placard. Each placard shows fields for PVI, CSN, SVI, CID, VIN, and Model, along with status indicators like 'Carrier', 'Job Detect', 'No Read', 'No Build', 'Set-In', and 'Set-Out'. The interface also includes a bottom status bar with function keys (F1-F11) and system information.



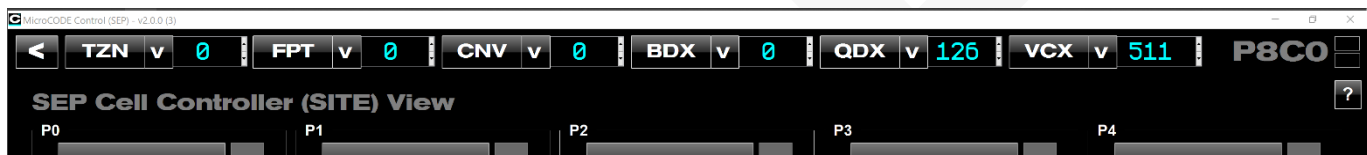
6) Track Zone Level Navigation

The SQL Extraction allows the App to handle Site Wide navigation by Track Zone name, automatically switching communication from one Cell Controller to another.

If the user moves between Cell Controllers—without using the new Track Zone navigation—the App instead returns the user to their last location in the Cell Controller.

The App remembers by Cell Controller:

- TRACK ZONE Index and Data Offset
- CONVEYOR Index
- FOOTPRINT Index
- GEPICS Buffer Index and Data Offset, and Filters
- GSIP Defects Index and Data Offset, and Filters
- TRACE Data Index and Data Offset, and Filters





7) Diagnostic Displays

New displays showing the health of the SEP Subsystems (PLC Programs) in both the GMP and GXP PLCs. These provide a 'Plant Monitoring' system in the Strategic Supplier Site when GPM&C is not available.

These displays are accessible by clicking on the Cell ID (PnCn) in the upper right corner of the SITE Screen, or by clicking on any of the Subsystem status indicators alongside the Cell selection buttons.

Cell Status: P7C0 CHASSIS FINAL

GEP Main Processor (GMP)		GEP eXtensions Processor (GXP)	
GMP GEP/SEP - Main Processor (GMP)	SIM Simulation Interface <Apps + Web> (GMP.SIM)	GXP GEP/SEP - eXtensions Processor (GXP)	SIM Simulation Interface <Apps + Web> (GXP.SIM)
DGN Internal SEP/GEP Diagnostics (GMP.DGN)	PVW PanelView Interface (GMP.PVW)	DGN Internal SEP/GEP Diagnostics (GXP.DGN)	PVW PanelView Interface (GXP.PVW)
CLX ControlLogix Hardware (GMP.CLX)	PMX Plant Monitoring Interface <GPM&C> (GMP.PMX)	CLX ControlLogix Hardware (GXP.CLX)	PMX Plant Monitoring Interface <GPM&C> (GXP.PMX)
GEP Main GEP/SEP Program (GMP.GEP)	UIX User Interface <Configuration> (GMP.UIX)	GEP Main GEP/SEP Program (GXP.GEP)	UIX User Interface <Configuration> (GXP.UIX)
	IOX External I/O Interface(s) (GMP.IOX)		IOX External I/O Interface(s) (GXP.IOX)
LTS Line Tracking Subsystem (GMP.LTS)	CNV Conveyor Interfaces (GMP.CNV)	BDX Conveyor Interfaces (GXP.CNV)	
SCN Scanner Interfaces (GMP.SCN)	QAX Quality Andon Interface <QAS 1,2,4> (GMP.QAX)		
EPX Error Proofing Actions (GMP.EPX)	VQX VQX Vehicle Quality Interface <GSIP> (GMP.VQX)		

GMP Summary			GXP Summary		
442	132	7%	37	67	7%
Controller Scan Time <Milliseconds> (GMP.CLX.CLS)	Controller I/O Connections (GMP.CLX.CLC)	Communication Time slice Remaining (GMP.CLX.TSR)	Controller Scan Time <Milliseconds> (GXP.CLX.CLS)	Controller I/O Connections (GXP.CLX.CLC)	Communication Time slice Remaining (GXP.CLX.TSR)

Key

SEP Subsystem States

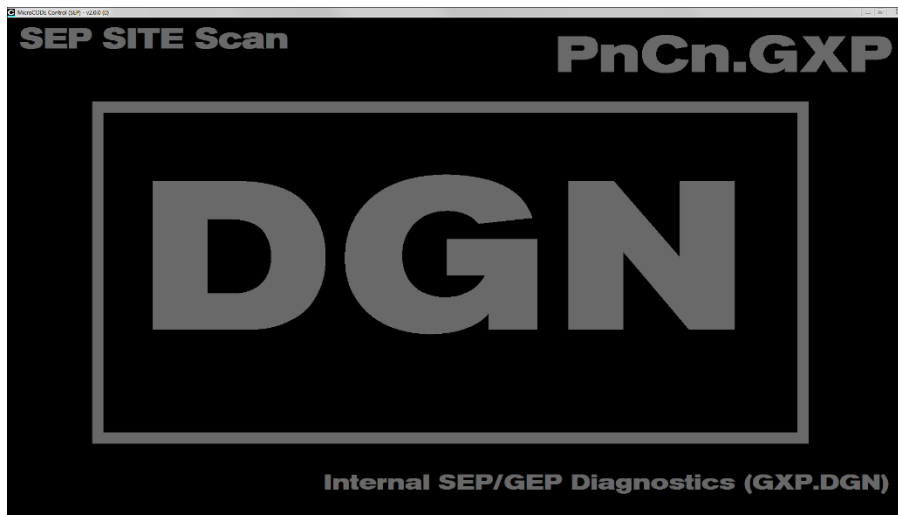
- AXP** UNCONFIGURED
- AXP** CONFIGURED
- AXP** INITIALIZED
- AXP** RUNNING - NORMAL
- AXP** RUNNING w/WARNINGS
- AXP** RUNNING w/ERRORS
- AXP** RUNNING w/FAULTS
- AXP** STOPPED

8) SEP Site Scan Display

A new display showing the health of the SEP Subsystems (PLC Programs) in both the GMP and GXP PLCs across all Cell Controllers configured in the Site.

This provides a 'Plant Monitoring' system in the Strategic Supplier Site when GPM&C is not available.

If all the SEP Subsystems in all Site Cell Controllers are fine—all Running with no Warnings, Errors, or Faults—the display rotates through them one at a time with a 'dim gray display', nice and boring.



If an abnormal condition is found the display shows the Cell Controller with the worst Subsystem state, shows the Cell Controller ID (PnCn.PLC) and shows the abnormal condition colorized by severity to draw attention to the issue.





9) Logix L8x Processor Support

This release includes a new ControlNet Interface Protocol (CIP) Driver capable of communicating with the new line of L8X PLCs. The cost of this new driver is built into the App upgrade or purchase price and provides a site-wide license allows as many clients (running this App) as required.

These processors feature 5x – 20x scan time improvements and a built in 1GB Ethernet Port.

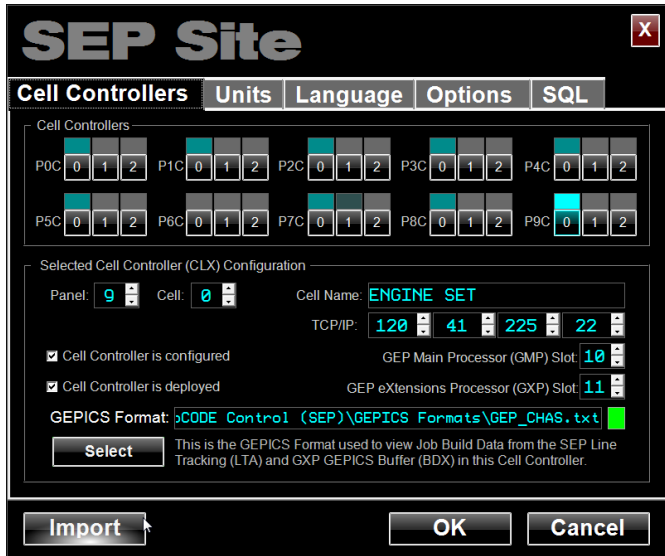
ControlLogix 5580 Controllers

Our ControlLogix® 5580 controllers provide increased performance, capacity, productivity, and security to help meet the growing demands of smart machines and equipment for manufacturing. All ControlLogix 5580 controllers use the Studio 5000® design environment as the standard framework that optimizes productivity, reduces time to commission. This framework manages Integrated Motion over EtherNet/IP for high-speed motion applications and SIL2/PLd and SIL3/PLe safety solutions. These controllers are ideal for applications that require high-performance communications, I/O, and motion control for up to 256 axes.



10) SITE Configuration IMPORT

This release allows the user to import a SITE Level Configuration file (.CFG) from a previous session or another User.



11) App Help Screens

There is now an App Help Screen available from the SITE and TRACK ZONE Views. It is accessed from the blue question mark button in the upper left.



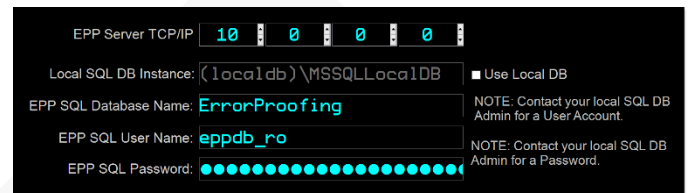
Corrected in Version v2.0.0 a (2)

The following defects were fixed in MicroCODE Control (SEP) in this Release:

1) SQL Connection Display

Issue: The dialog was not showing the 'Local DB' name dimmed/disabled when not in use.

Correction: The field is disabled when not in use.

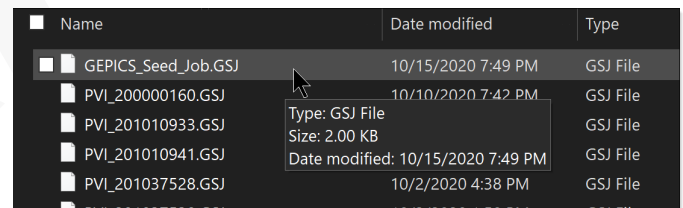


2) Cannot Import a GEPICS Seed Job

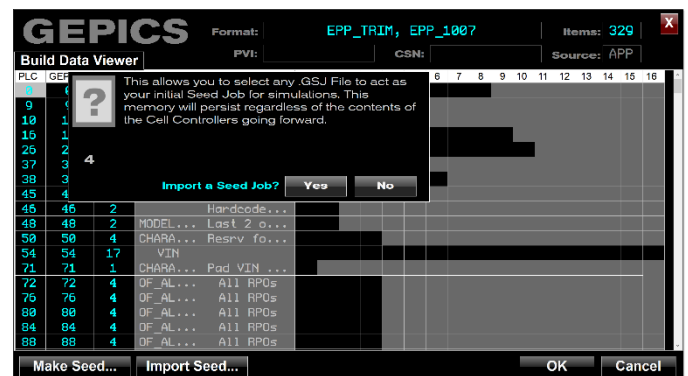
Issue: When the local PLC has no Build Data the U.I. does not have a feature to import a GEPICS Seed Job (.GSJ) from a file.

Work-around: Copy any existing GEPICS Seed Job (.GSJ) file to this path and rename:

C:\Users\<<Username>\Documents\MicroCODE Control (SEP)\GEPICS Seeds\



Correction: You can now Import any .GSJ file to become your Seed Job.





DRAFT



3) Required Fonts were not being delivered by the MSI Install file

Issue: Fonts required for proper App operations were not being delivered in the App folder.

Correction: The v1.0.0 installer was corrected to match the SEP v1.0.0 installer and deliver the Fonts as documented.

4) An unhandled exception occurs when opening the GEPICS Build Data Viewer without a Format configured

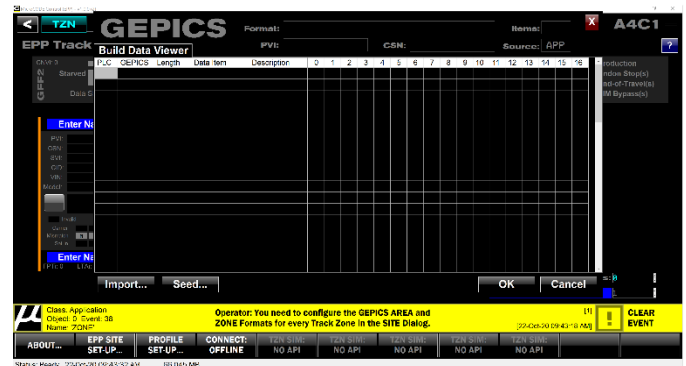
Issue: When opening the GEPICS Build Data Viewer if no GEPICS Format has been configured in the SITE: TRACK ZONE tab the App was displaying an unhandled exception. The user could continue working but it was annoying.



Work-around: Configure the GEPICS Formats for every Track Zone in SITE: TRACK ZONE tab.

PLC	Track Zones					
TZni	Name	Footprints	FPTsi	Conveyor	AREA Format	ZONE Format
0	GFF2T21	0	0	GFF2_FPS		N
1		0	0			N
2		0	0			N

Correction: The App now warns the User to configure the AREA and ZONE Formats instead of faulting.





New in Version 2.0.0 a (1)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) The target of the GEPICS Jobs can now move to any Footprint

In prior versions the App would only generate simulated Jobs into the first Footprint of a Track Zone (like an LTC). Now you can select any Footprint as the target of the simulation, and this can be changed on-the-fly.

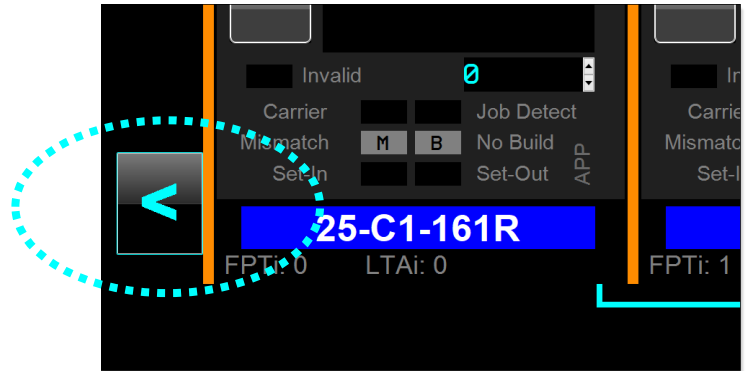
This is controlled by the new 'Job Focus' Left/Right Buttons in the Track Zone View.

NOTE: With this new feature the concept of "Simulation Owner" is deprecated because it is no longer required. Any user with the App can generate GAPs, EMPTY CARRIERS, or JOBS into any position on the Conveyor that they choose.

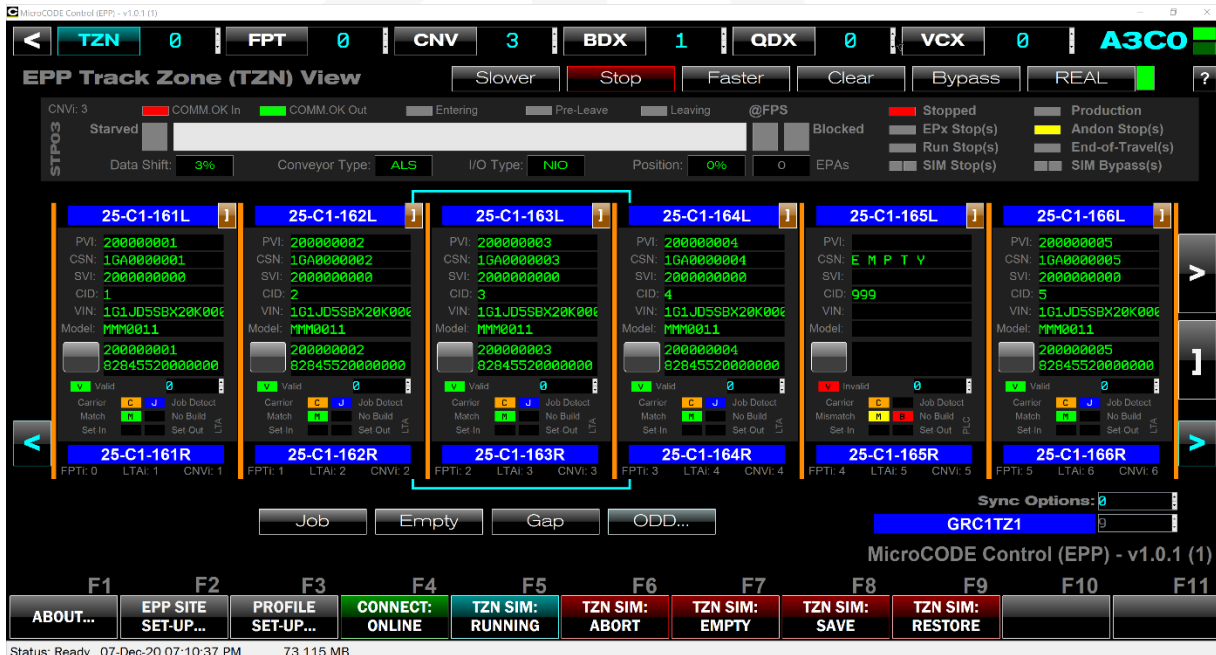
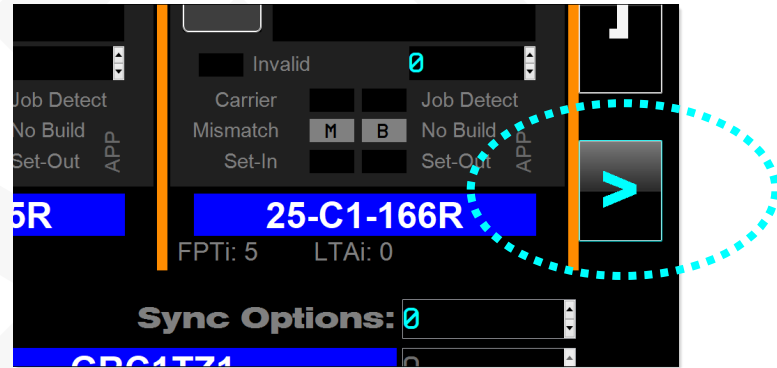
The **JOB**, **EMPTY**, and **GAP** commands follow the location of the 'Job Focus'.

The location of the 'Job Focus' is displayed with a Cyan Outline...

Move 'Job Focus' to the left one Footprint...



Move 'Job Focus' to the right one Footprint...



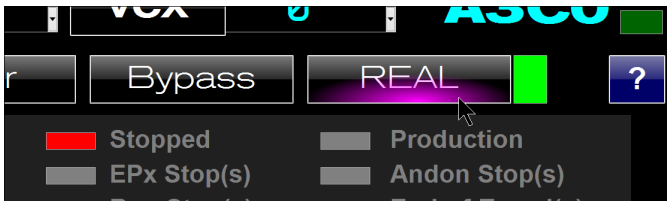
Status: Ready 07-Dec-20 07:10:37 PM 73.115 MB

2) You can release the Conveyor from simulation but continue generating Jobs

In prior versions the App would always place the Conveyor in simulation mode while generating simulated GEPICS Jobs.

When the GM Strategic Suppliers are at the end of the try outs, they want to run the real conveyor, but continue simulating GPEICS Jobs.

So, the two have been de-coupled with the 'REAL/SIM' button a toggle to go between the two,



3) App Events are now auto-cleared

After 7 seconds of no events, they are cleared off the App Screen.



Corrected in Version v2.0.0 a (1)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) The Profile Dialog loses modifications

Issue: When configuring modification to the Build Data Packet it is possible to lose modifications.

Correction: This is possible if you configure change after a 'NONE' and was noted in the User Guide.

*** Caution:** The App processes the changes from the top and the first row the has NONE for HEADER changes and NONE for OPTIONS changes is deemed the end of the entire change list.

Example: The **RPO** modification shown at position 300 below will be lost, because the **NONE** above it will be taken as the end of the packet.

Length	Data	OPTIONS	Position	Length	Data
9	NNN...	PVI	0	9	NNN...
16	NNN...	SKIP	---	---	---
11	NNN...	CSN	26	10	NNN...
17	NNN...	VIN	54	17	NNN...
8	MMM...	MODEL	38	7	MMM...
n/a	n/a	RPO	500	3	UM1
n/a	n/a	NONE	n/a	n/a	n/a
n/a	n/a	RPO	300	3	UM2
n/a	n/a	NONE	n/a	n/a	n/a



New in Version v2.0.0 a (2)

The following features were added to **MicroCODE Control (SEP)** in this Release:

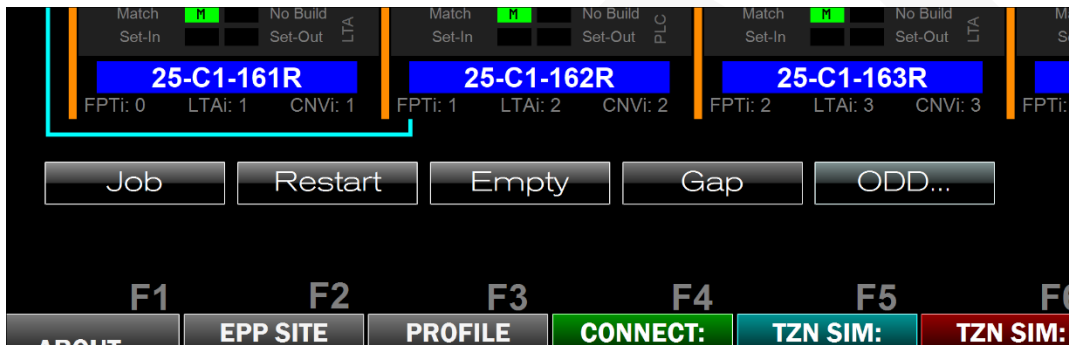
1) Complete SQL Integration

With the integration to the SEP SQL Database complete the feature where a user could configure temporary Footprint Placard names (DD-SS-OOOAs) has been deprecated.

This feature only appeared when there was no SQL data and was not visible in most cases anyway.

2) RESTART Job Work

A new command was added to specific restart error proofing work on the Job in the selected Footprint. Previously work was restarted with the JOB command, but this also gave the User a different PVI, a different Job. Users—at times—want to use a single PVI for several SAT cases.



3) Invalid Job data is now displayed

In previous versions the App did not read and display Job data from the Tracking image until the '.Finalized' bit was set by the SEP PLC code.

This could hide issues from the users, causing confusion.

The App now shows this data in dark green to aid in debugging App/PLC interaction issues.

See Chapter 9 'Common Issues' in the User Guide for examples.

Corrected in Version v2.0.0 a (2)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) SIM Commands can fail

Issue: Simulation commands at time failed to response properly.

Correction: the communication sequence between the App and the PLC have been updated to provide a more robust response.



New in Version v2.0.0 a (3)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) Display of API mismatch

While the App has always displayed the version of the API code in the PLC, it has not indicated if a mismatch is a problem.

In order for the App to work properly the version of the Windows App and the PLC API must be the same.



Requirement: Ensure the App user is aware that there is a version incompatibility between the PLC API and Windows APP.

Implementation: The App has draws attention to a mismatch by coloring the existing API Status Indicator.

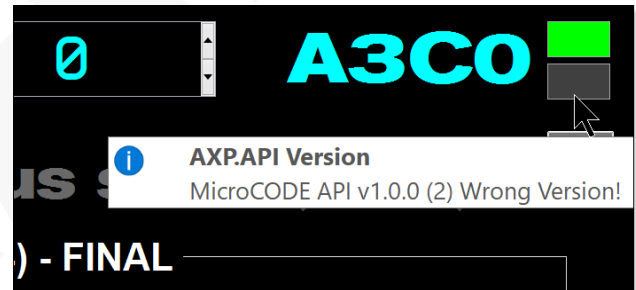
TEST CASES: This software tests were performed prior to release to ensure App functionality.

Test Case 1: Run the App with matching API version, the indicator should be green – **Passed**.

Test Case 2: Run the App with mis-matching API version, the indicator should be magenta and note “Wrong Version!” – **Passed**.



And with a warning: “Wrong Version!”

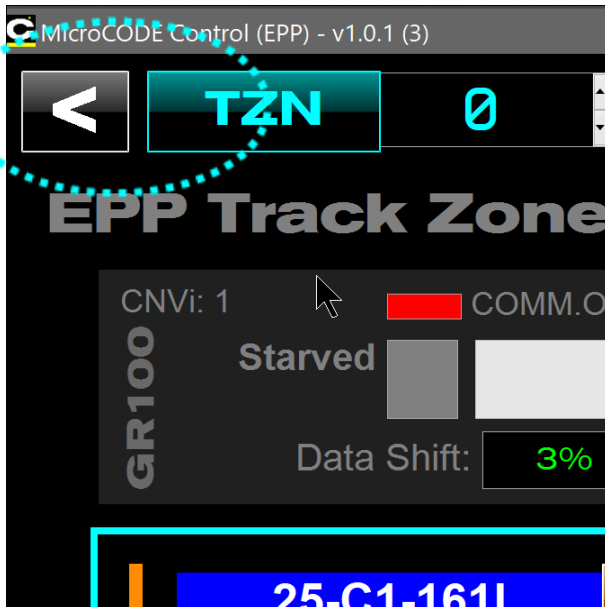


2) Improved 'Back' Function

In previous versions the 'Back' button always returned to the SITE level from any screen, and to the SITE Monitor from the SITE View.

Requirement: Treat the 'BACK' [<] button more like a browser 'Back' button and evaluate the user one level in the Site hierarchy.

Implementation: The 'Back' button now returns first to the Track Zone View, and then to the SITE View screen.



TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Clicking BACK from the CNV, FPT, BDX, QDX, and VCX Views should all return the user to the TZN View – **Passed**.

Test Case 2: Clicking BACK from TZN View should all return the user to the SITE View – **Passed**.

Corrected in Version v2.0.0 a (3)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) Last Footprint name is missing

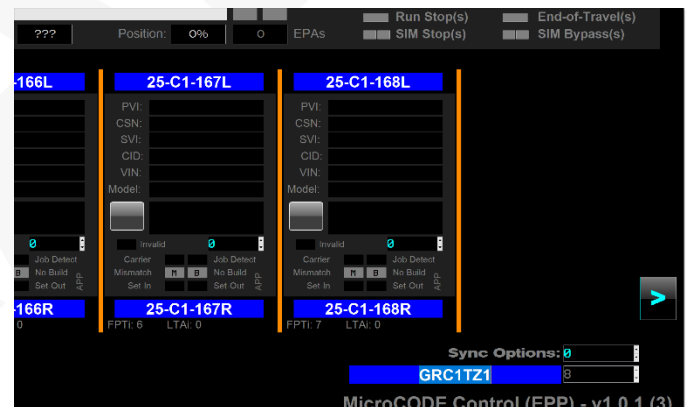
Issue: After pulling SEP configuration out of the SEP SQL Database the last Footprint names of each Track Zone.

Correction: The App was using the ending index of a Track Zone as the count and was corrected.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Clear the App configuration and re-query the KUKA and ESYS databases, all Footprint names should align right up to the last one – **Passed**.

Test Case 2: A database with a Track Zone with no Footprints should not accidentally create one in this App during the query – **Passed**.



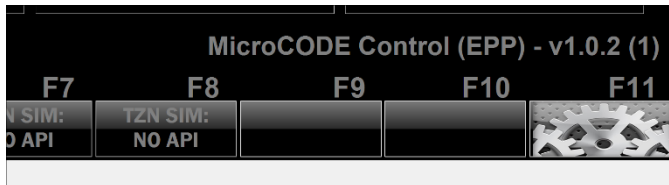


New in Version v2.0.0 a (4)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) Standardizes SETTINGS dialog box in all MicroCODE Control Apps

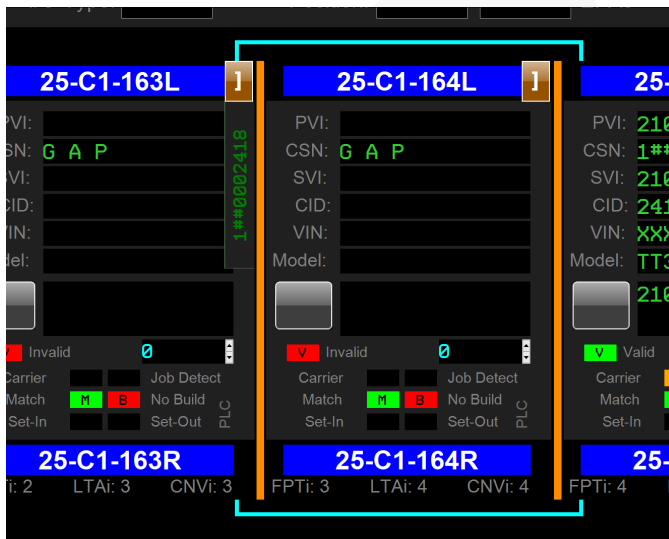
All Control Apps now have the 'SETTINGS' dialog boxes moved to the F11 function key and now show a typical 'Gear' image to aid in discoverability.



1) SEP Shift Buffer Positions are now visible on Stop Station Conveyors

To aid in debugging Stop Station tracking issues the Control App now displays the CSN of any Jobs in the PLC's Buffer Positions.

Below CSN '2418' is moving between -163 and -164:



Corrected in Version v2.0.0 a (4)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) Header Changes are being discarded from Job PROFILES

Issue: When configuring a change to the GEPICS HEADER in a Job PROFILE changes to the MODEL CODE were being discarded.

User changes MODEL to "TTX6545"...

HEADER	Position	Length	Data	OPTIONS
PVI	0	9	NNNN...	PVI
SVI	16	16	NNNN...	SKIP
CSN	26	11	NNNN...	CSN
VIN	54	17	NNNN...	VIN
MODEL	38	8	TTX6545	MODEL
NONE	n/a	n/a	n/a	NONE
NONE	n/a	n/a	n/a	NONE

But it immediately reverts to the template "MMM0000"...

HEADER	Position	Length	Data	OPTIO
PVI	0	9	NNNN...	PVI
SVI	16	16	NNNN...	SKIP
CSN	26	11	NNNN...	CSN
VIN	54	17	NNNN...	VIN
MODEL	38	8	MMM0000	MODE
NONE	n/a	n/a	n/a	NONE

Resolved: This is 'as designed' behavior, the left side panel is used to show legal Header formats.

Changes are made on the 'Data' side, which is also able to modify the Header.



New in Version v2.0.0 a (5)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) Treat all Conveyor Interfaces as one thing for SIM / REAL toggle

Requirement: When toggling an ALS/AGV/VAC Conveyor in or out of Simulation Mode the user had to toggle each Conveyor interface, one for each Stop Station individually. This was time consuming and error prone.

Implementation: Toggle any Stop Station into or out of SIM Mode now does all Stop Stations in the Track Zone.

Corrected in Version v2.0.0 a (5)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) Corrected App interpretation of the SEP SQL DB relationships

Issue: Track Zones were being extracted from the SEP SQL DB as PLCs due to a query issue.

Solution: The APP queries were corrected.

2) Connected the App to multiple SEP PLCs simultaneously was produced 'cross-talk'

Issue: The User could see a Track Zone momentarily display data from a different SEP PLC.

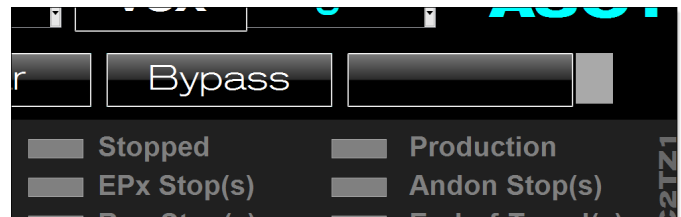
Solution: The API for the SEP PLCs required AREA and CELL be added to distinguish one from another.

The App only deals with one SEP PLC at a time but maintains a 'status connection' to all those marked **deployed** in the SETTINGS dialog box.

3) Clarify SIM/REAL Display on Stop Station Conveyor Interfaces

Issue: To the User it appeared that ALS/AGV Stop Stations were toggling in and out of SIM Mode as they moved from Station to Station.

Solution: The App was defaulting to REAL mode in the display while waiting for updated data from the PLC to show the actual Mode. This waiting period—usually less than a second—now shows a blank display for Mode to avoid confusion.





New in Version v2.0.0 a (6)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) New App Events were added to clarify why a Simulation command is rejected

Requirement: When a Simulation Command is rejected, it should be clear to the User exactly why it will not execute.

Implementation: Three (3) Events were added:

- Simulation is BUSY
- Simulation MODE is not enabled
- Controller is not CONNECTED

2) The Footprint Simulation commands are now multi-threaded for speed and overlap

Requirement: Users had to wait on some commands that were being executed in real-time.

Implementation: These commands are now executed as background threads and can be overlapped:

- ENTERING – ALS/AGV/AGC/VAC
- LEAVING – ALS/AGV/AGC/VAC
- Generate NEW JOB

Corrected in Version v2.0.0 a (6)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) Simulated JOBS always start at 00001 even if there are higher CSNs in Track Zone

Issue: When generating new Jobs, the simulation always seems to start at CSN 1GA0000001, unless a complete simulation is done in one session.

Solution: The APP now reexamines the CSNs in the Track Zone whenever you go ONLINE and starts subsequent Jobs with the next CSN.

2) Corrected ability to place SVI in the App generated GEPICS Orders

Issue: If the GEPICS Format did not mention “SVI” in the DATA ITEM name, it was set to SKIP.

Solution: The APP now looks for “SVI” in the DATA ITEM Name and Description.

0	PVI	0	9	NNNNNNNN0	PVI	0	9	NNNNNNNN0
1	SVI	16	16	NNNNNNNN0	SKIP	---	---	---
2	CSN	26	11	NNNNNNNN0	CSN	26	10	NNNNNNNN0
3	VIN	54	17	NNNNNNNN...	VIN	54	17	NNNNNNNN...
4	MODEL	20	0	NNNNNNNN	MODEL	20	7	NNNNNNNN

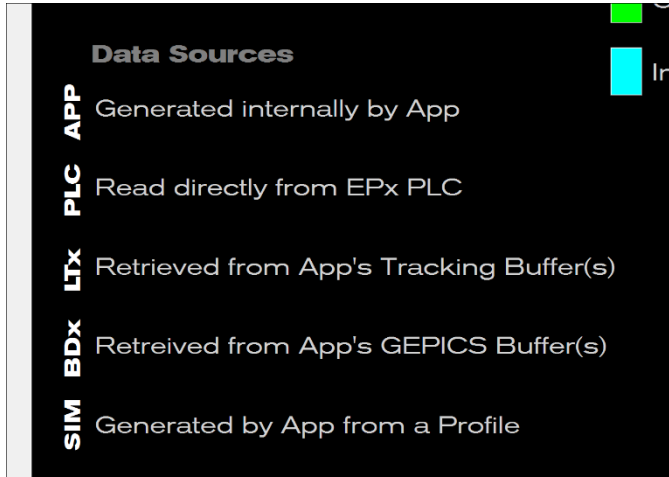
0	PVI	0	9	NNNNNNNN0	PVI	0	9	NNNNNNNN0
1	SVI	16	16	NNNNNNNN0	SVI	16	9	NNNNNNNN0
2	CSN	26	11	NNNNNNNN0	CSN	26	10	NNNNNNNN0
3	VIN	54	17	NNNNNNNN	VIN	54	17	NNNNNNNN

Data Item	Description	0
PVI		2
CHARACTER_SPACES	Pad PVI to Length of 10	
BSSEQNUM	Last 6 of BSSN	
CHARACTER_SPACES	Pad SVI to length of 16	
CSN		1

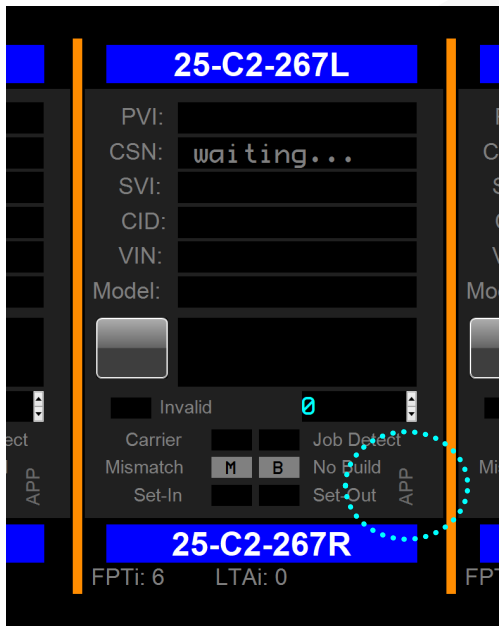
3) Additional Data Source Help Definitions

Requirement: The source of some data on the App Screens was not clear.

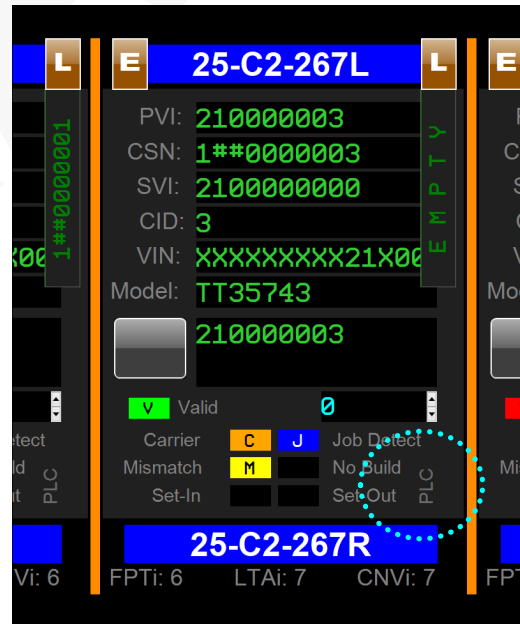
Implementation: Key App Objects now display the source of their data and the Help Screen was updated with the definitions of these sources.



Footprint display initialized by App, but not populated from PLC... Source: **APP**.



Footprint display filled by App, directly from PLC reads... Source: **PLC**.

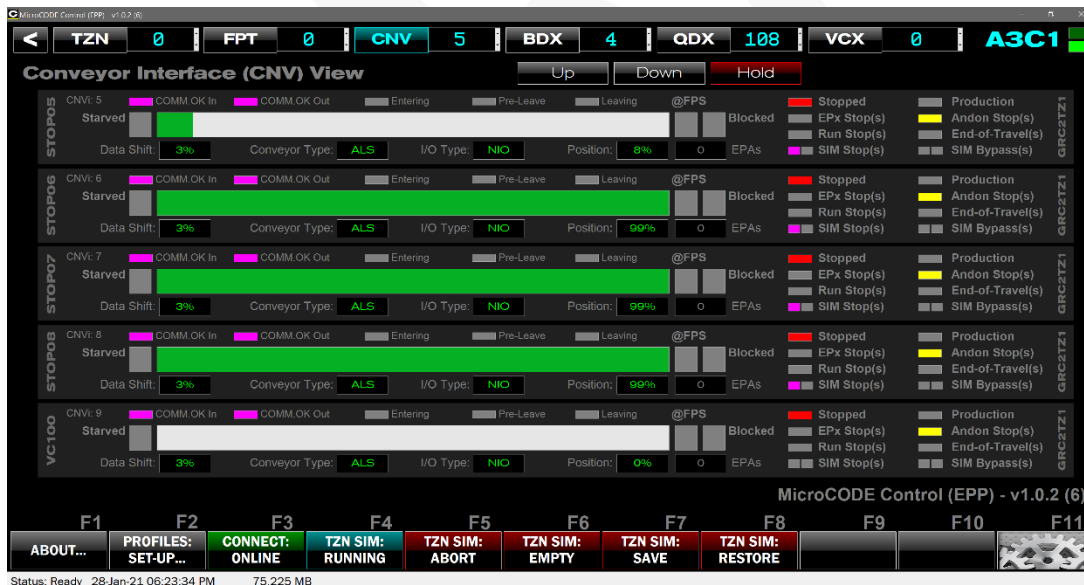
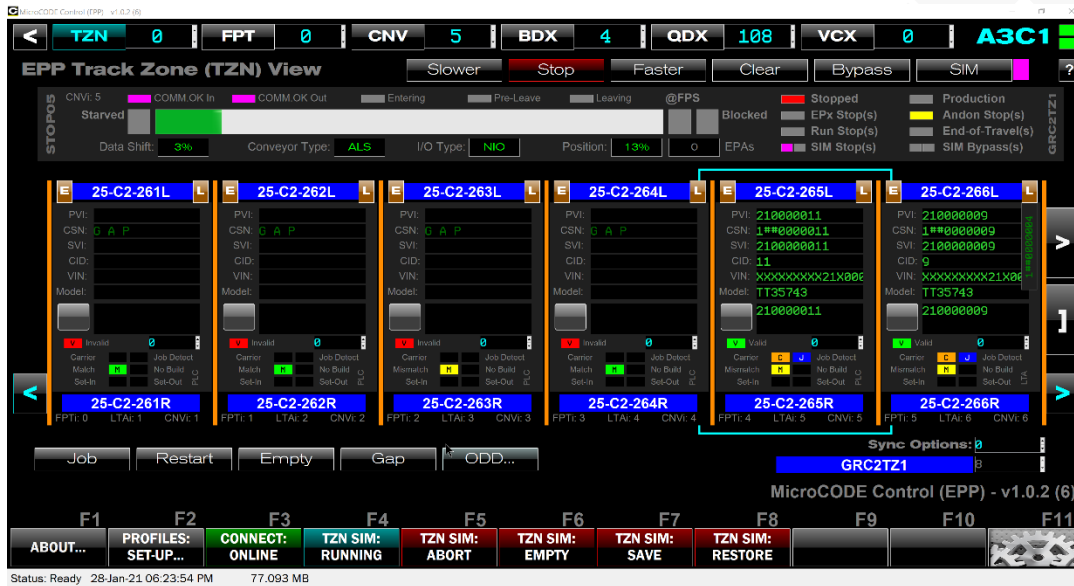




3) Stop Station TAKT Time is not displayed when a Footprint is selected

Issue: For ALS/AGV/AGC/VAC Conveyors the % Travel (TAKT Time) does not come from the Conveyors; it is generated on the SEP side in the Footprint Programs.

Solution: The APP now translates Conveyor Index into a Footprint Program references and reads the TAKT time from the Program level Tag `"ConvStatus.PercentTravel"`.



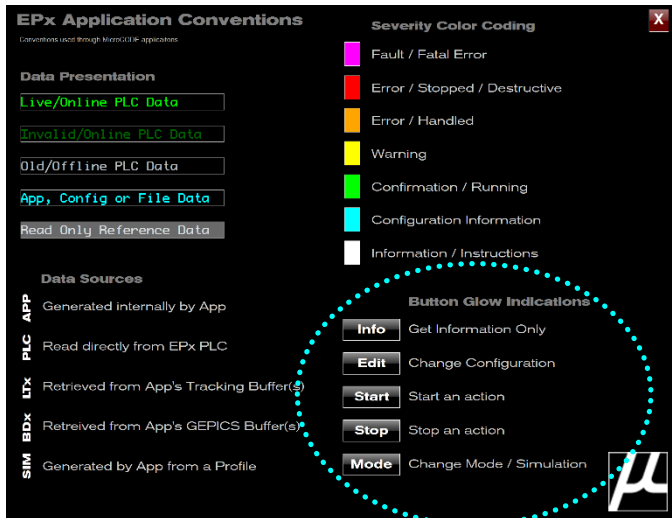
New in Version v2.0.0 a (7)

The following features were added to **MicroCODE Control (SEP)** in this Release:

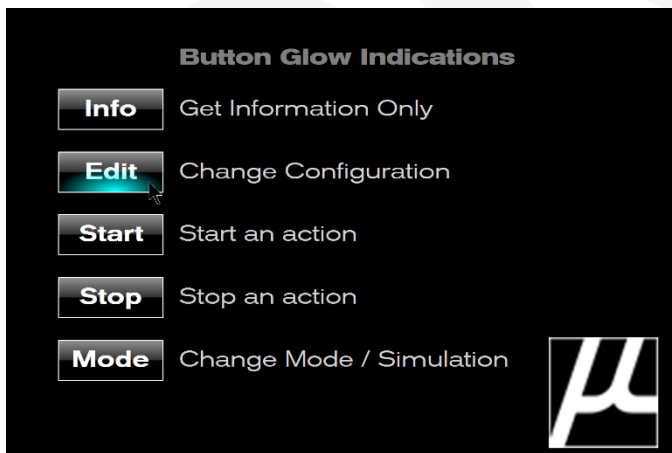
1) Help Screen – additional information

Requirement: Ensure Users understand the App conventions to ease use.

Implementation: Added definition of button ‘Glow’ indications and reviewed App for consistent use of these indications. This release also added the [?] help button to every major screen.



Hovering over these buttons reveals the color of the ‘glow’ indications.



2) Support for variable size GEPICS Formats

Requirement: Some GEP/SEP Plants implemented Build Data Packets size other than the standard 2,000 bytes. These would not display properly in the App.

Implementation: The App now uses the GEPICS Format associated with the Cell Controller as the definition of the maximum packet size. This controls how much is read, written, and displayed.

The LDT v2.2 SEP PLCs have the Build Data Packet set to **3000** bytes, **2765** Options bytes.

Address	Format	ASCII	SINT
+ LTS_LTA[0].BD.Byte[2754]	'\$F4'		SINT
+ LTS_LTA[0].BD.Byte[2755]	'\$01'		SINT
+ LTS_LTA[0].BD.Byte[2756]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2757]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2758]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2759]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2760]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2761]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2762]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2763]	'\$00'		SINT
+ LTS_LTA[0].BD.Byte[2764]	'\$00'		SINT
+ LTS_LTA[1]	{...}		udtLTS_...
+ LTS_LTA[2]	{...}		udtLTS_...

But their GEPICS Format only defines **1835** bytes, **1761** Options bytes so that is all that is read or written.

SEP	GEPICS	Length	Data Item	Description	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1500	1580	14	CIVX BB/...		2	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	X
1520	1594	8	E03 CIVX BB/...		8	4	7	3	1	5	1	8									
1528	1602	8	CHARA...	spaces f...																	
1536	1610	14	CIVX Sun...		6	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	X
1538	1624	8	A13 CIVX Sun...		-	-	-	-	*	*	*	*									
1538	1632	8	CHARA...	spaces f...																	
1556	1640	14	BUICK EL...		9	2	2	0	2	0	4	0	0	0	0	0	0	0	0	0	X
1588	1654	8	G02A BUICK EL...																		
1588	1662	30	CHARA...																		
1618	1692	4	M30A CIVX Keyfob		2	7	5	1													
1622	1696	4	P05H Prop 05 ...																		
1626	1700	89	CHARA...																		
1715	1789	44	CHARA...																		
1759	1833	3	End of D...	E N D																	



3) Show EPA ID and Description in the Defect Queue

Requirement: To aid in SATs the user needs GSIP Machine Codes translated to SEP or SEP EPA/Task IDs along with their descriptions.

Implementation: The SQL query of the EPx SQL Databases (SEP and EPP) now pulls:

- EPA / TASK ID
- Machine Code
- Footprint location (DSOA)
- Description

These are kept in the App's .CFG file and are now used to translate Machine Code to EPA ID, Description and Location.

Example – SEP TASK Shown:

The screenshot shows the 'GSIP Defect' application window. At the top, it displays 'Machine Code: 25327', 'Defects: 2', 'PVI: 210000006', and 'CSN: <Not Queued>'. Below this, it shows 'C2-BC-004R' and 'TASK: 10242'. A large 'BC' logo is visible. The main area is titled 'Defect Data Viewer' and contains a table with the following data:

PLC	GSIP RLN	Defect	Description
0	25327002	002	RELEASED
1	25327005	005	SC - MISSED SCAN or PT.03 - NO CYCLE

If the EPA does not have a Machine Code in the SQL Database, you will get this display... which typically means you needs to re-query your EPx database from the SETTINGS dialog box.

The screenshot shows a message box with a question mark icon on the left. The text inside the box reads: '< Can't locate via Machine Code >' followed by 'DD-SS-000A' in a blue box on the right.



Example – SEP TASK Shown:

GSIP Defect

Machine Code: 25403 Defects: 7

PVI: 221999007 CSN: <Not Queued>

PT C1-PT-002 ASFL012 TASK: 10340 O.I.: 129

AIR SUSPENSION FILL 25-C1-166L

PLC	GSIP RLN	Defect	Description
0	25403002	002	RELEASED
1	25403006	006	SC - DUPLICATE VID or PT.04 - CYCLE ABORTED
2	25403008	008	SC - INVALID S/N or PT.06 - TOOL SPECIFIC #2
3	25403009	009	SC - UNCOLLECTED or PT.07 - TOOL SPECIFIC #3
4	25403010	010	PT.08 - TOOL SPECIFIC #4
5	25403011	011	PT.09 - TOOL SPECIFIC #5
6	25403012	012	PT.10 - TOOL SPECIFIC #6

OK Cancel

TASK Type
 PP, TT, BC, PT,
 VC, CE, etc.

TASK Name
 As entered in the error
 proofing system.

TASK ID
 As entered in the error
 proofing system.

TASK Description
 As entered in the error
 proofing system.

TASK Location
 DD-SS-000A
 Footprint Placard.

TASK O.I.
 Unique ID within
 this Cell Controller
 for the Light Stack /
 Keyswitch Device.



4) Show EPA ID and Description in the Trace Queue

Requirement: To aid in SATs the user needs GSIP Machine Codes translated to SEP or SEP EPA/Task IDs along with their descriptions.

Implementation: The SQL query of the EPx SQL Databases (SEP and EPP) now pulls:

- EPA / TASK ID
- Machine Code
- Footprint location (DSOA)
- Description

These are kept in the App's .CFG file and are now used to translate Machine Code to EPA ID, Description and Location.

Example – SEP EPA shown:

GEPICS Trace

Station Name: 25C1089R System Code: VCVSTRAC
 Machine Code: 25142 Record Key: C1FUELTANK
 EPA Id: 255 Parts: 3
 PVI: 201463360 CSN: 0

EPA: 14 SC C1YX VCVS FUEL TANK 25-C1-089R

SEP	VPPS / Component	Part No.	DUNS / Vendor	Trace Data	Payload Data
0	Y9510000000000X	P84847392	12V840208362	TA121028CFFSA1278	10S00001...
1	Y9529700000000X	P84847394	12V812751138	T6221012000038134	...
2	Y9511200000000X	P13274188	12V588176862	T1M16044@R2655620	...

OK Cancel

EPA ID
 Unique ID within this Cell Controller.

EPA Type
 PP, TT, SC, PT, VC, CE, etc.

EPA Name
 As entered in the error proofing system.

EPA Location
 DD-SS-OOOA
 Footprint Placard.



5) The JOB FOCUS now follows the PVI

Requirement: While following Builds the User is usually focused on a specific Job / PVI. It would be nice if the 'Job Focus' followed the Job automatically.

Implementation: After the User moves the Job Focus to a Footprint with a PVI, the focus ten will move along with that Job until it reaches the end of the Track Zone or the User selects a different Job.

Focus is also kept with Track Zone boundaries as part of this change.

The screenshot displays the 'EPx Track Zone (TZN) View' interface. At the top, there are navigation buttons for TZN, FPT, CNV, BDX, QDX, and VCX, along with a 'A3CO' indicator. Below this, there are status indicators for 'COMM.OK In', 'COMM.OK Out', and '@FPS'. The main area shows a sequence of footprints (FPT) and conveyor units (CNV) with associated PVI, CSN, SVI, CID, and VIN information. A blue arrow indicates the current focus moving from footprint 25-C1-163L to 25-C1-164R. The bottom of the screen features a function key bar with buttons for ABOUT..., PROFILES: SET-UP..., CONNECT: ONLINE, TZN SIM: START, TZN SIM: ABORTED, TZN SIM: EMPTY, TZN SIM: SAVE, TZN SIM: RESTORE, and a camera icon. The status bar at the bottom left shows 'Status: Ready 03-Feb-21 12:53:58 PM 80.359 MB'.



6) Maintain GEPICS Formats, PLC IP Addresses and 'Deployed' State

Requirement: While re-querying the SQL Database to get current Tracking and Action / Task configuration the App was losing any information that is not held in the SQL Database.

This includes for EPP:

- GEPICS AREA Format for each Track Zone
- GEPICS ZONE Formats for each Track Zone
- PLC Deployed State

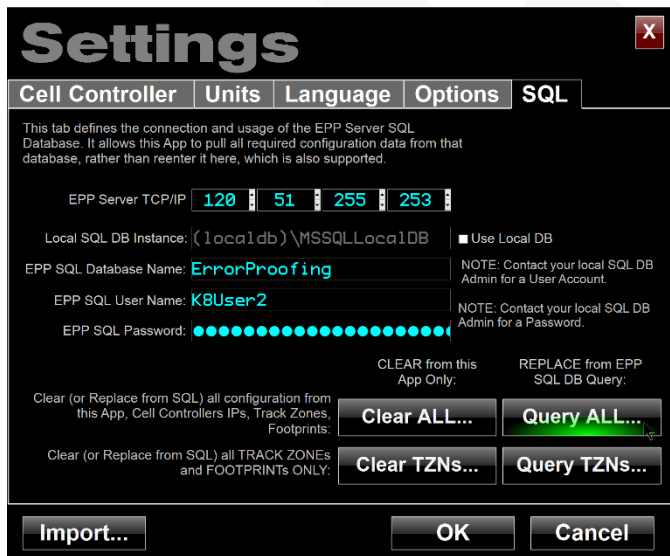
For SEP this includes:

- GEPICS Format for each Cell Controller

Implementation: When the App re-queries the SEP or SEP SQL Database from the SETTINGS Dialog box it now saves a copy of the current User Settings.

Then after the query, where the AREA/PANEL and CELL/PLC Names match, the GEPICS Format(s), PLC IP Addresses, and the User's Deployed state are restored.

This allows you to freely re-query the EPx SQL DB at anytime to get tracking and action/task changes without disturbing your ability to immediately go online and run SATs.





7) Display 'Kick-Out' Jobs

Requirement: SEP has the ability to KICK-OUT a Job.
This action stops work on the Job but does not open GSIP Defects.

Implementation: The NO BUILD indicator now show KICK-OUT in Magenta when this is present in the Tracking image.



8) Quick Exit without Configuration Changes

Requirement: How that the App carries EPA / TASK configuration data there is a noticeable delay on exit.

Implementation: The App now skips re-writing the configuration data on exit—the source of the delay—if no changes have been made to it since it was last saved.



Corrected in Version v2.0.0 a (7)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) The Asynchronous Entering, Leaving, and creating Working Job Commands announce failure prematurely.

Issue: Now that the commands execute in background threads, they are, by definition, out-of-sync with the User UI. So, the User UI cannot be the thread announcing the success or failure of the action.

Solution: The App Events for success or failure were moved into the background threads with the execution code.

2) Cell Controllers with more than (16) Conveyor Interfaces had issues.

Issue: There were accidentally limitations in the App when dealing with more than (16) Conveyor interfaces.

Solution: This has been expanded everywhere to support (160) interfaces, the SEP maximum.

3) SIM Mode is left on for different Stop Stations on ALS/AGV/VAC Conveyors

Issue: When going in and out of Simulation Mode some individual Conveyor Interfaces were left in SIM mode or not place in SIM mode.

Solution: Corrections were made to the PLC API code to ensure all Stop Stations follow the Track Zone mode (SIM/REAL) or they all follow the toggling of any specific Conveyor interface.





New in Version v2.0.0 a (8)

The following features were added to **MicroCODE Control (SEP)** in this Release:

1) Display TASK Information on SEP Trace Queue Events

Requirement: The SEP TASK information is not displaying on TRACE Queue Events like it does on GSIP Defect Queue Events.

Implementation: In the SEP DEFECT and TRACE Code the Action (TASK) GSIP Machine Code is carried in the entries as a unique identifier on the GM IT side of the world.

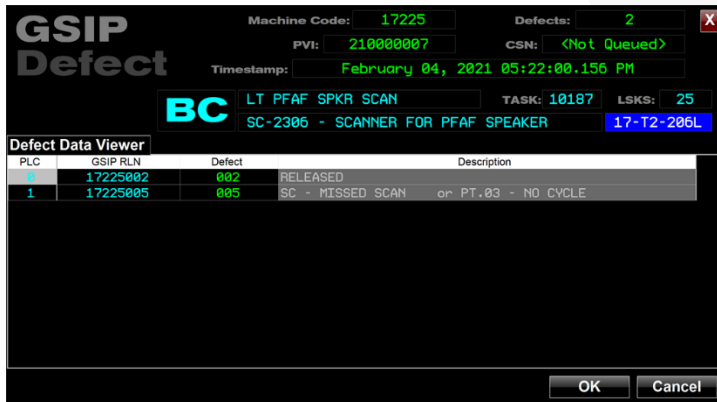
This allows for a backward link to the EPA that generated the TRACE or DEFECT.

In the SEP GSIP Queue the Machine Code is there, and the TASK lookup can be completed...

But, in EPP, the Machine Code was eliminated from the TRACE Queue, preventing this backward link, so when the Machine Code is missing the Control App uses the **Station Name** to locate the TASK that mostly likely generated the TRACE Data.

This is **not** a unique relationship, as you can have multiple SCANS/TRACE Action in the same Footprint Operation.

But without a unique identifier in the SEP TRACE Queue there is nothing available to backward link to a unique TASK.



NOTE: We tried to use the VPPS but found the SEP Server is not populating the TASK ID in the Attribute table, again preventing a backward link to the TASK.

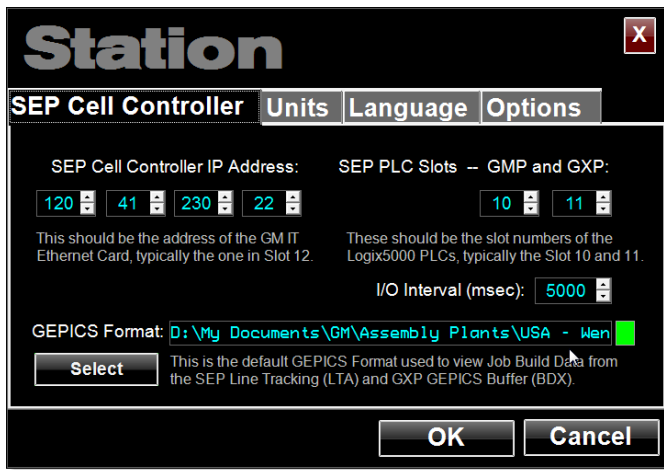


Corrected in Version v2.0.0 a (8)

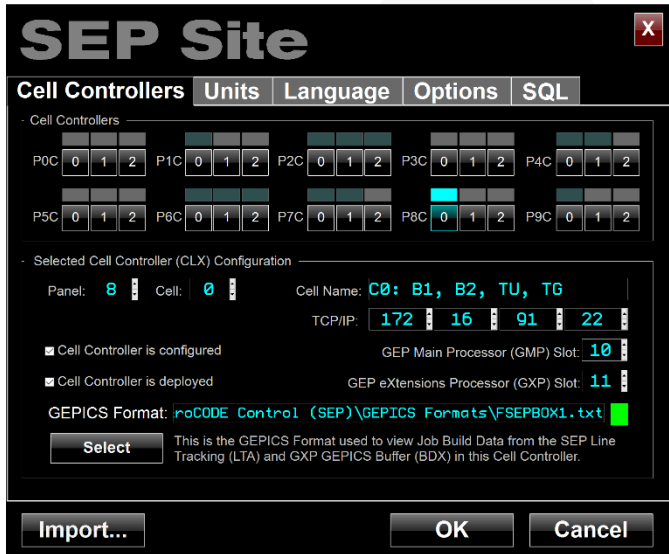
The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) Can't see name of the GEPICS Format

Issue: In prior releases you could not see the name of the GEPICS Format you selected if it had a long path. (This despite the App using 'right-justified' text).



Correction: The App now used programmatic right-justify so you can always see the file name...



New in Version v2.0.0 a (9)

The following features were added to **MicroCODE Control (EPP)** in this Release:

1) Pulldown Menus are now associated with all App Views.

Requirement: User are asking for extensions to the App's functionality, in order to add functionality, the App needs a common method to extend each View.

The current Views are:

- SITE
- TRACK ZONE
- FOOTPRINT
- CONVEYOR
- BUILD DATA
- DEFECTS
- TRACE DATA

Implementation: To accommodate this pulldown menus have been added to each View Button.

The Footprint Pulldown is shown below, this is used for quick navigation to any Footprint by its DSOA (Department, Section, Operation DD-SS-000).



By view the menus are:

- SITE – no pulldown
- TRACK ZONE – Navigate the Site by Track Zone
- FOOTPRINT – Navigate a Cell by Footprint
- CONVEYOR – Navigate a Cell by Conveyor
- BUILD DATA – Export and Print Orders
- DEFECTS – Export and Print Defects
- TRACE DATA – Export and Print Trace Data

NOTE: The **Print** and **Export** features for the BDX, QDX, and VCX Views will be implemented in a future release.



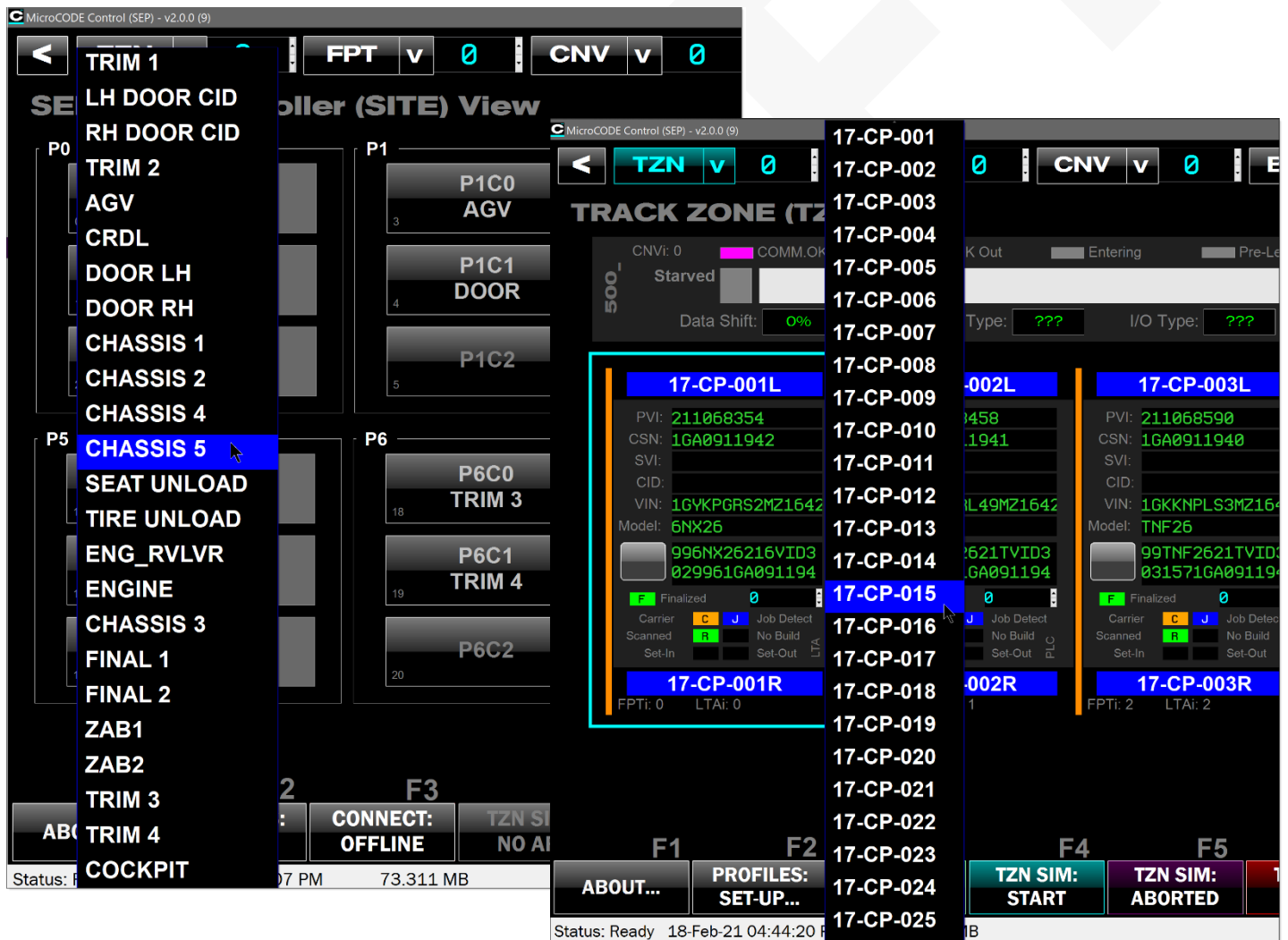
2) The new Pulldown Menus are be used to navigate to any Track Zone and Footprint from the TRACK ZONE View

Requirement: Quick navigation around an entire SITE.

Implementation: The TZN Pulldown can be used to directly jump to any Track Zone in the SITE regardless of which Cell Controller is in housed in.

This saves the step of going back to the SITE View, picking a different Cell Controller, and then navigating to a specific Track Zone.

You also do not have to know which Cell Controller holds the Track Zone you are interested in watching.



2) Additional TRACK ZONE Pulldown jump control

Requirement: Simplify / Clarify / Control where a User can jump to in the Track Zone Pulldown menu.

Having all Track Zones available in the TZN Pulldown was misleading, as the User could not get to them all while ONLINE with the EPP PLCs present.

Implementation: The TZN Pulldown has been modified as follows:

- In the SITE VIEW you can see and selected all Track Zones in all EPP PLCs.
- In the TRACK ZONE VIEW you only see the Track Zones in the currently selected EPP PLC.
- While the APP is running / online you can only jump to Track Zones in connected EPP PLCs.
- When the APP is configuring / offline you can jump to any Track Zone that is configured

If you attempt to jump to a Track Zone that is in a disconnected EPP PCL—while the APP is running—you will see the following warning:

The screenshot shows the MicroCODE Control interface for an EPP Cell Controller (SITE) View. The top navigation bar includes buttons for TZN, FPT, CNV, BDX, QDX, and VCX, along with a status indicator A3C2. The main area is divided into sections for EPP Cell Controller (SITE) View and ControlLogix Hardware (EMP.CLX). The SITE View section contains a grid of track zone buttons (A0C0-A0C5, A1C0-A1C5, A2C0-A2C5, A3C0-A3C5, A4C0-A4C5). The ControlLogix Hardware section shows a list of track zones (EP_GTT1TZ1, EP_GVD1TZ1, EP_GVR1TZ1, A3C3, A3C4, A3C5) with corresponding DPY buttons. A yellow warning banner at the bottom reads: "Operator: The selected TRACK ZONE is UNAVAILABLE because the selected EPP PLC is OFFLINE." Below the warning is a row of buttons: ABOUT..., PROFILES: SET-UP..., CONNECT: ONLINE, TZN SIM: RUNNING, TZN SIM: ABORT, TZN CTL: CLEAR, TZN CTL: SAVE, TZN CTL: RESTORE, and CLEAR EVENT. The status bar at the bottom indicates "Status: Ready 17-Feb-21 11:06:08 AM 69.165 MB".



3) The scope of the TZN Functional Keys

Requirement: The scope—which Track Zone will be affected—of the TZN Function Keys was unclear.

Implementation: The “TZN” related Function Keys are now disabled unless the User is looking at the TRACK ZONE View, where they can see the tracking that will be affected by the command.

The “TZN” Function Keys always affect a single-Track Zone, the current one seen in the TRACK ZONE View.

BUILD DATA (BDX) View

BDX: / BDB: / PLC	PVI:	CSN:	Model:	Options:
BDX: 74 / BDB: 26 / PLC	211065554	1GA0908975	TNF26	99TNF2621TVID3000981G/0
BDX: 75 / BDB: 25 / PLC	211065434	1GA0908974	6NH26	996NH26215VID3001411G/0
BDX: 76 / BDB: 24 / PLC	211065921	1GA0908973	6NW26	996NW26216VID3001711G/0
BDX: 77 / BDB: 23 / PLC	211065483	1GA0908972	TNF26	99TNF2621TVID3000951G/0
BDX: 78 / BDB: 22 / PLC	211065471	1GA0908971	TNB26	99TNB2621TVID3001841G/0
BDX: 79 / BDB: 21 / PLC	211065834	1GA0908970	6NW26	996NW26216VID3000761G/0
BDX: 80 / BDB: 20 / PLC	211065396	1GA0908969	TNB26	99TNB2621TVID3001041G/0
BDX: 81 / BDB: 19 / PLC	211065860	1GA0908968	TND26	99TND2621TVID3001151G/0
BDX: 82 / BDB: 18 / PLC	211065574	1GA0908967	6NW26	996NW26216VID2993361G/0
BDX: 83 / BDB: 17 / PLC	211065235	1GA0908966	TNB26	99TNB2621TVID2993171G/0
BDX: 84 / BDB: 16 / PLC	211065757	1GA0908965	TNF26	99TNF2621TVID3001241G/0
BDX: 85 / BDB: 9 / PLC	211065355	1GA0908964	6NY26	996NY26216VID3001351G/0

MicroCODE Control (SEP) - v2.0.0 (9)

F1 ABOUT... F2 PROFILES: SET-UP... F3 CONNECT: ONLINE F4 TZN SIM: NO TZN F5 TZN SIM: NO TZN F6 TZN SIM: NO TZN F7 TZN SIM: NO TZN F8 TZN SIM: NO TZN F9 F10 F11

Status: Ready 18-Feb-21 04:49:55 PM 77.831 MB

4) Add option to clear a TRACK ZONE to GAPS or EMPTY CARRIERS

Requirement: On a STOP STATION line the preferred starting state is all GAPS, i.e.: nothing to block simulated Carriers as they are created. On an FPS Conveyor the preferred stating state is a line full of EMPTY CARRIERS. So, the option is needed to select which the User wants when clearing the Track Zone.

Implementation: The “TZN CTL: CLEAR [F6]” command now gives the option to clear to GAPS or EMPTYs. To do neither select [X] or hit ESCAPE.





New in Version v2.0.0 a (10)

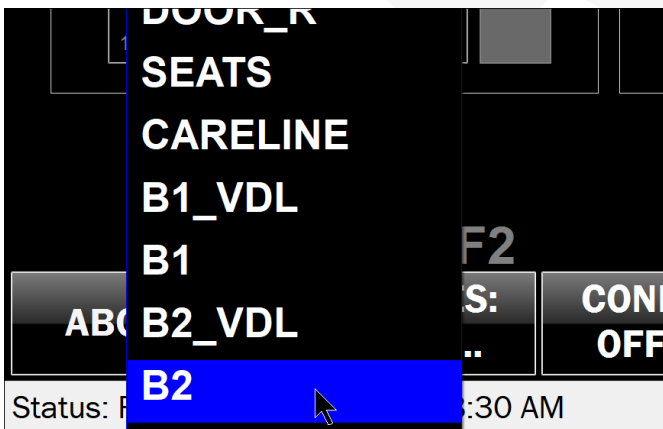
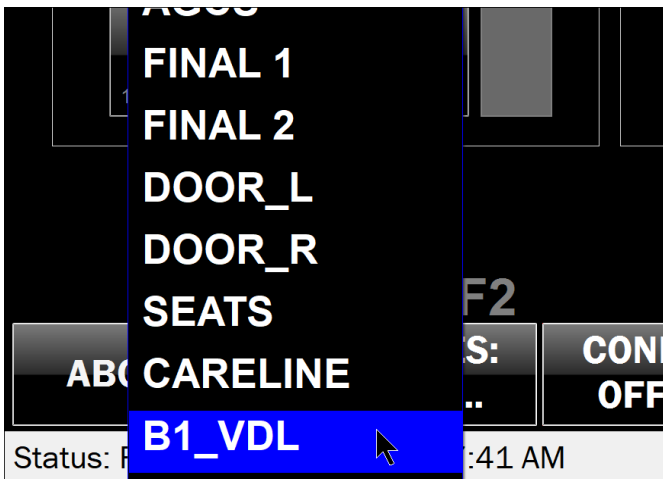
The following features were added to **MicroCODE Control (EPP)** in this Release:

1) App now supports scrollable pull-down menus

Large facilities have more Track Zones, Footprints, or Conveyors than can be shown in the TZN, FPT, and CNV Pulldown menus all-at-once

Requirement: Allow the User to access menu items that are off screen.

Implementation: With any Pulldown menu active—mouse in the menu—where the menu is outlined in blue, you can use the Mouse Wheel or the Keyboard Arrow Keys to scroll the menu and access all items.





Corrected in Version v2.0.0 a (10)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) All Conveyors were not shown at times in the CONVEYOR View

Issue: In Multi-Track Zone PLCs it seemed random as to the number of Conveyor interfaces that were being shown in the CONVEYOR View.

Solution: The App was corrected to properly calculate which Conveyor Interfaces should be shown from a given starting point (interface).

The screenshot displays the MicroCODE Control (SEP) v2.0.0 (10) interface. The top navigation bar includes buttons for TZN (1), FPT (31), BDX (0), QDX (0), VCX (0), and P1C2. The main area is divided into two panes. The left pane, titled 'CONVEYOR Interface (CNV) View', lists several conveyor interfaces (CNV) with their respective status indicators (Starved, COMM.OK In, COMM.OK Out) and data shift percentages (0%). The right pane shows a detailed view of a selected CNV (C107), displaying its status (Starved), position (0%), and various control buttons (Blocked, EPx Stop(s), Run Stop(s), SIM Stop(s), Production, Andon Stop(s), End-of-Travel(s), SIM Bypass(s)). The bottom status bar indicates 'Status: Ready' and '21-Feb-21 09:30:14 AM'.



New in Version v2.0.1 b (1)

The following features were added to **MicroCODE Control (EPP)** in this Release:

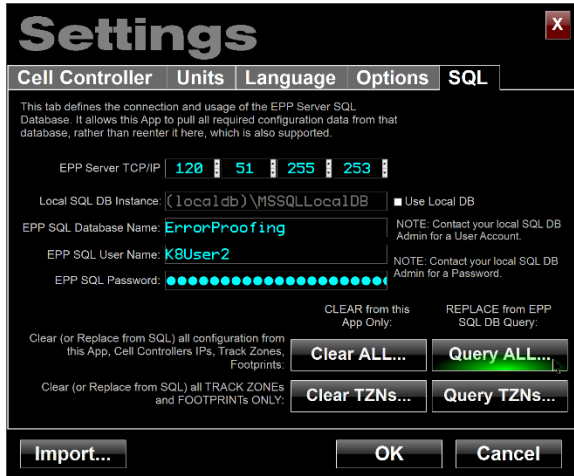
1) SQL Interface has been simplified.

There was some User confusion about when to use REPLACE ALL or REPLACE TRACK ZONES.

Requirement: Originally the idea was to limit the time it took to update the App from SQL but allowing selective updates, like just the Tracking information.

Implementation: In practice replacing everything only takes about 8 seconds. So, the 'TRACK ZONES' only option was deprecated for simplicity's sake.

OLD:

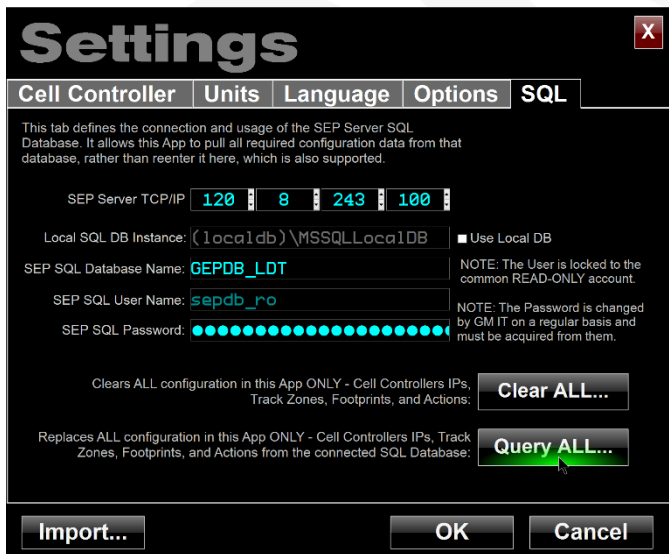


NOTE: The interface to the SQL Database now makes every effort to maintain:

- **PLC IP Addresses**
- **PLC Deployed States**

This, in order to make periodic updates of the App from the connected SQL DB completely painless, i.e.: removing the need to reconfigure IPs for each PLC and resetting the 'deployed' checkbox to indicate which PLCs are reachable in your environment.

NEW:



2) EPP 'Tracking Buffer' is now displayed for FPS Conveyor

Due to code issues in the EPP PLC Jobs can enter the 'Tracking Buffer' and get stuck there. To the User they appear to just 'disappear' and never come back.

It turns out they are actually copied into the background 'Tracking Buffer' and are not shifted back out.

Requirement: Allow the User to see the 'Tracking Buffer' position on an FPS, just like an ALS/AGV Conveyor' to help diagnose issues.

Implementation: Any time there is Job data in a Buffer position it will be visible in the App regardless of Conveyor Type.



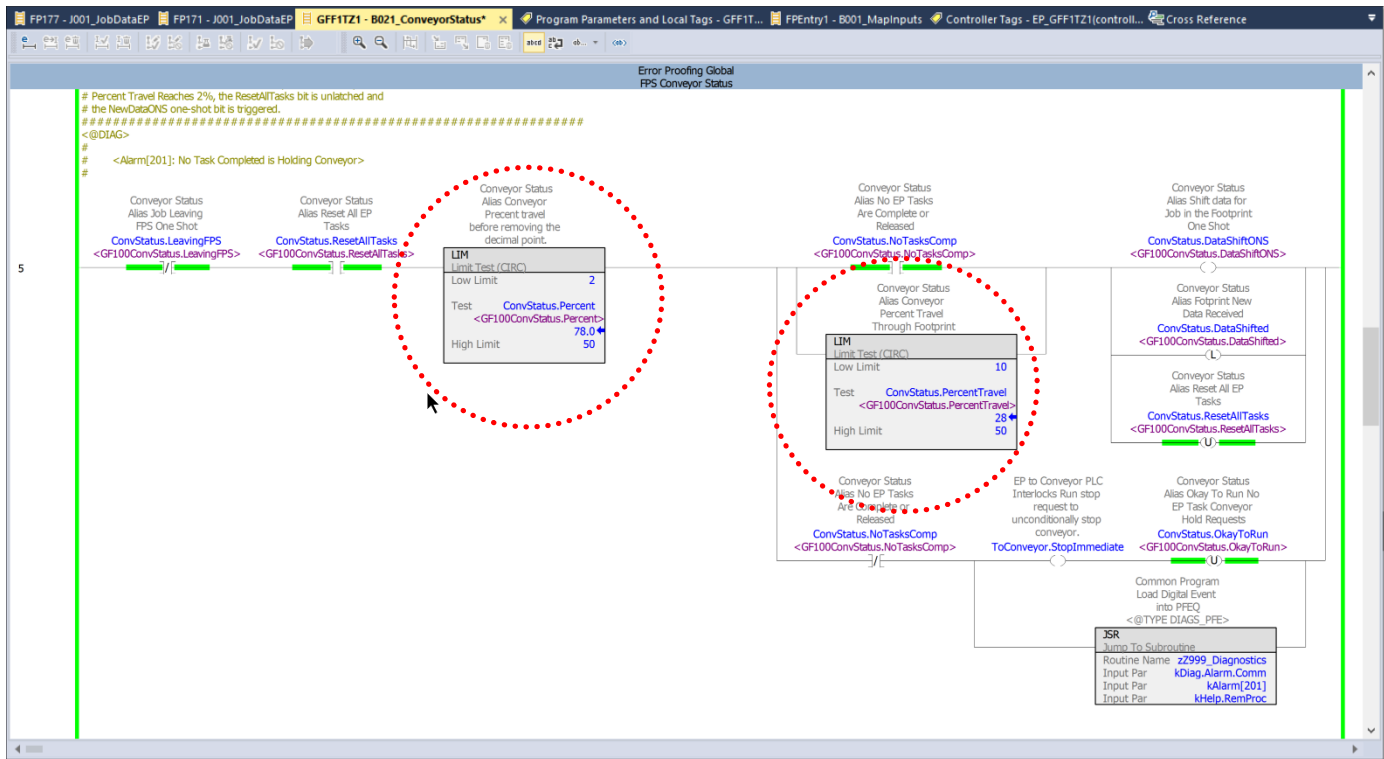


Control.NET™ App for GM SEP

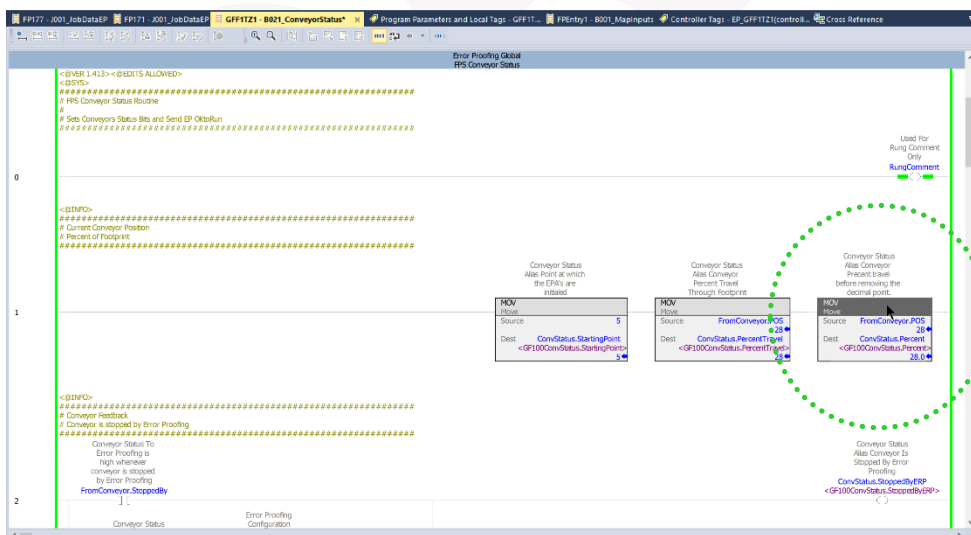
Release Notes Alpha Release: 3.0.0 a (1)



The root cause was the EPP code reference Conveyor % Travel from two (2) different sources in the same rung, one updated from the Conveyor consumed tag and one that wasn't...



The simple correction was to duplicate the consumed % Travel into both places for the EPP code in **B021_ConveyorStatus**.... Normal tracking resumes after this change.



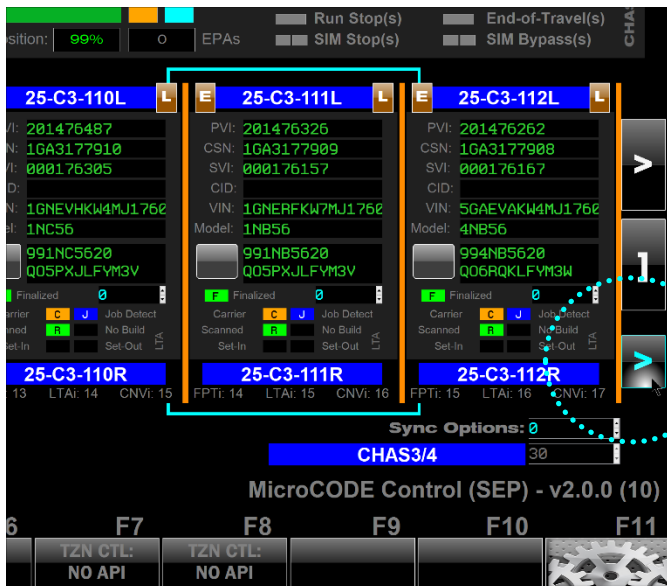
Corrected in Version v2.0.1 b (1)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) JOB FOCUS Left/Right Buttons are not following the Job off-screen.

Issue: When using the Left/Right Focus buttons vs. eth Keyboard Arrows keys the App was not auto-panning to follow the Job focus.

Solution: Corrected to match the behavior of the Arrow Keys and keep the Job with focus on the screen.



The App also follows the JOB FOCUS by PVI. So once you have focused on a Job the cyan highlight to automatically pan the view to keep that Job on the screen as it moved down the line.



2) Issues running Simulation of FPS Conveyors.

Issue: After all the changes in the App and EPP PLC code the good old FPS Conveyor simulation was not working.

- Job weren't being generated
- Conveyor wasn't moving
- Data wasn't shifting
- Etc.

Solution: All of these issues were resolved in v1.0.3b6.

Be sure to update both the **APP** and the **API** in the PLC and tie the API.CNV simulation into the EPP PLC Code as documented in the **User Guide – Appendix A1**.

The screenshot shows the MicroCODE Control (EPP) - v1.0.3 (6) interface. At the top, there are navigation buttons for different zones: TZN, FPT, CNV, BDX, QDX, VCX, and A4CO. Below this is the 'TRACK ZONE (TZN) View' section, which includes a 'Slower' button, a 'Stop' button, a 'Faster' button, and other controls like 'Clear', 'Bypass', and 'SIM'. The main area displays a grid of conveyor zones, each with its own set of data fields and status indicators. The zones shown are 27-F1-171L, 27-F1-172L, 27-F1-173L, 27-F1-174L, 27-F1-175L, and 27-F1-176L. Each zone has fields for PVI, CSN, SVI, CID, VIN, and Model. Below these fields are status indicators for 'Valid', 'Carrier', 'Job Detect', 'Mismatch', 'No Build', and 'Set-Out'. At the bottom of the interface, there are buttons for 'Job', 'Restart', 'Empty', 'Gap', and 'ODD...'. The status bar at the very bottom indicates 'Status: Ready 24-Feb-21 11:36:38 AM 82.137 MB'.



New in Version v2.0.1 b (2)

The following features were added to **MicroCODE Control (EPP)** in this Release:

1) The App now memorizes the last Track Zone you were on along with Area and Cell

The App strives to keep your 'Context' consistent from one session to the next, so you can "pick up where you left off". The Track Zone was missing from this.

Requirement: Maintain context across App sessions.

Implementation: Memorize Track Zone.

2) The Tracking and Conveyor data display is now event based and much faster

The App now displays tracking changes from the Logix 5000 background thread directly into the UI thread on change.

Requirement: Improve App responsiveness to UI Simulation Commands.

Implementation: The periodic display update of tracking information has been replaced with display-on-change. As soon as the background thread talking to the PLC has new data it shown immediately on the screen, Footprint-by-Footprint, Conveyor-by-Conveyor.



3) The configuration of the App's API is now fully automatic

The goal of this App is an API program that can be loaded into an EPP PLC and never touched, like the SEP API program.

Requirement: EPP's program construction make this very difficult. Where SEP uses indexed arrays of common objects, EPP has named references. Example...

SEP

Track Zone #1's Tracking Image is in...
LTS_LTA[X] where X is a configured index.

Track Zone #2's Tracking Image is in...
LTS_LTA[Y] where Y is a configured index.

EPP

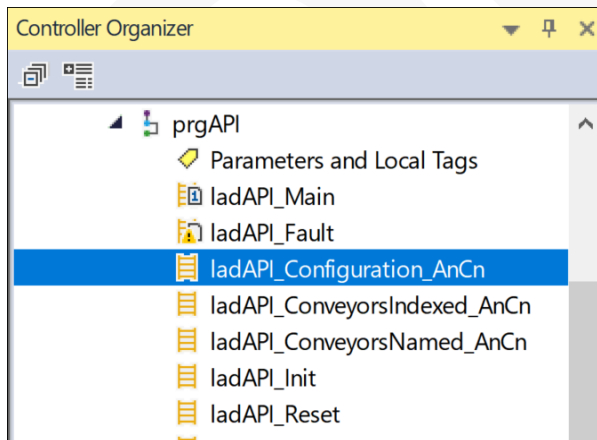
Track Zone #1's Tracking Image is in...
TrackingImage1 *and* **TrackingBuffer1**.

Track Zone #2's Tracking Image is in...
TrackingImage2 *and* **TrackingBuffer2**.

To hide these differences and make for a common API the essential data is placed in the API Tags API.TZN...

In SEP it's automatic, in EPP (until the Release) it was manually configured by the Controls Engineer

Implementation: With this release of the App, it now configures the API automatically from the SQL DB configuration. The routine **ladAPI_Configuration_AnCn** has been removed.



This release also removed the need for the routines **ladAPI_ConveyorsIndexed_AnCn** and **ladAPI_ConveyorsNamed_AnCn** to map Conveyor I/O into the API Tags.

The App now does this itself when a Conveyor is in SIM mode.

When a Conveyor is in REAL mode the App is reading the Conveyor I/O from:

FPnnn: fromConveyor and
FPnnn: toConveyor

When a Conveyor is in SIM mode the App is transfer the Simulated Conveyor I/O...

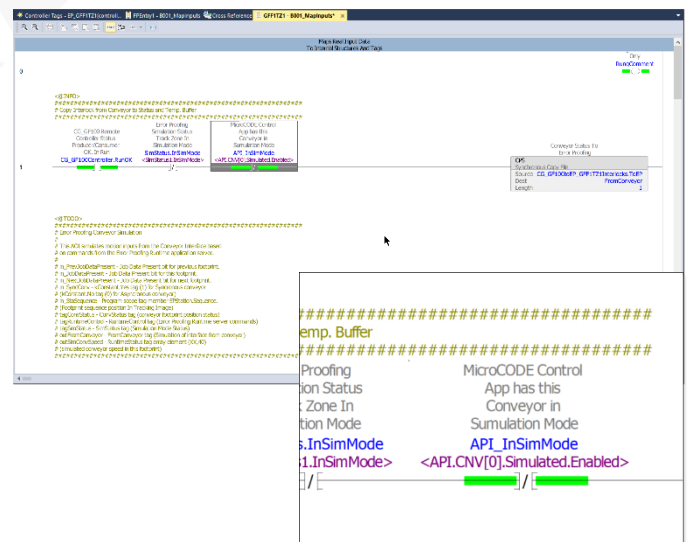
From:

API.CNV[CNV].I and
API.CNV[CNV].O

To:

FPnnn: fromConveyor and
FPnnn: toConveyor

So, this still required that the Control App SIM Mode contact be added to each MAP INPUTS routine to block the loading of 'fromConveyor' by the Consumed Tag.



4) The App now starts faster, exists faster

The App could take a while to startup.

Requirement: Speed is a feature.

Implementation: App entrance was rewritten to eliminate redundant operations and data loads.

5) The App now announces the loss of communication to the PLC(s) more clearly

Once communication is lost the App continuously tried to continue working, filled the App Event log with errors.

Requirement: Show the User he has no network connection very clearly and go OFFLINE to stop faulting.

Implementation: And so...

The screenshot displays the MicroCODE Control (EPP) v2.0.1.12 interface. At the top, there are control buttons for various zones: TZN (v 0), FPT (v 0), CNV (v 1), BDX (v 0), QDX (v 0), and VCX (v 0). The main display area is titled "TRACK ZONE (TZN) View" and shows a "STOP!" warning: "The App has lost communication with its API program. The App will now go OFFLINE, check connections to the PLC." Below this, there are four job cards for zones 25-P1-181R, 25-P1-182R, 25-P1-183R, and 25-P1-184R. A yellow banner at the bottom reads: "Maintenance: Communication Error with the AXP-API Interface, ensure MicroCODE 'prgAPI' is installed in Cell Controller's EPP PLC." The status bar at the bottom shows "Status: Ready 01-Mar-21 02:36:49 PM 71.550 MB".



Corrected in Version v2.0.1 b (2)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) Conveyor Type was not being saved from session to session

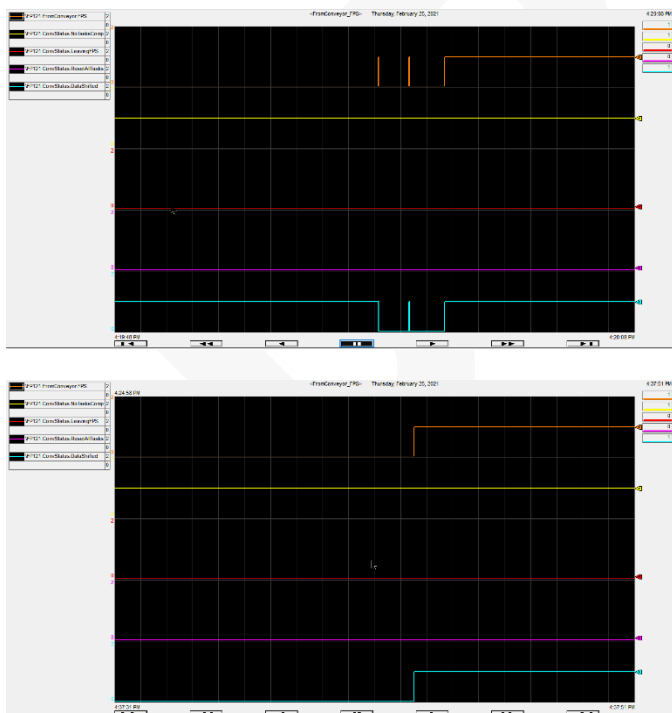
Issue: The App strives to keep your 'Context' consistent from one session to the next, so you can "pick up where you left off". The Conveyor Types were missing from this.

Solution: Maintain context across App sessions, memorize Conveyor Types.

2) In Sub-Assembly Footprints, Jobs generated by the App disappear

Issue: Jobs generated by the App sometimes appear momentarily and then disappear or never appear when expected.

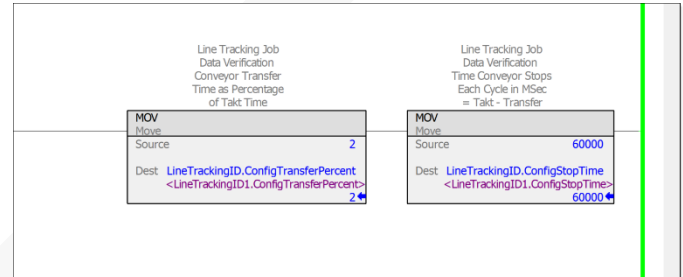
Root Cause: The Jobs were being erased by the EPP PLC code. This was happening in the first Footprint of an AGV/ALS/SAI Track Zone. The App was accidentally given the EPP PLC a 'pulse' in the FPS interlock and this caused two data shifts. Before and After correction are shown below...



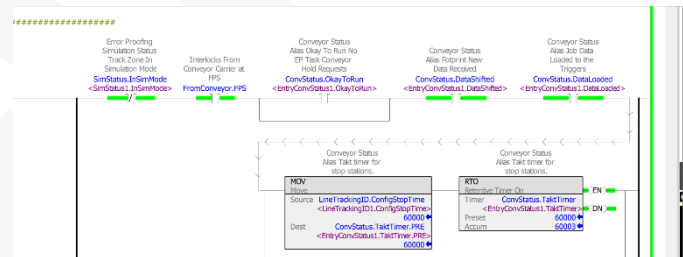
3) TAKT Timer is not running for Footprint

Issue: After the App moves a Job into a Footprint the EPP TAKT Timer (% Travel) never increases.

Root Cause: The EPP PLC code needs to be configured properly.



And the EPP PLC code does not accumulate time until all Tasks are 'OK to Run'. This needs to be corrected in the EPP PLC code. Time waits for no man, nor any Task, it must start when the Job arrives.



TAKT Time moving...



3) User context was being lost when saving or refreshing SQL configuration

Issue: The Track Zone, Footprint, and Conveyor the User is looking at was getting reset and causing display issues immediately after using the SETTINGS dialog box.

After using the SETTING dialog box, context lost...



Solution: Internal App corrections related to the loading of User preferences.



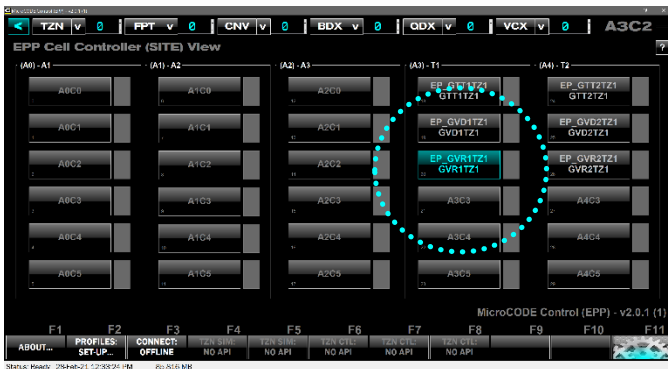


4) Old Controller Reference non-existent after a SQL Import

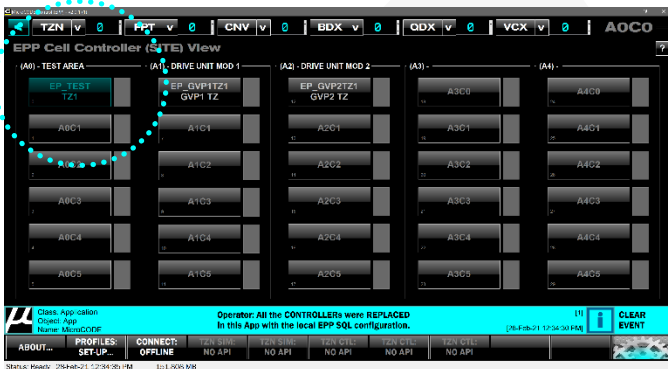
Issue: when switching between EPP SQL DBs the App could be left pointing to a Cell Controller that does not exist in the new SQL DB. This was easily fixed by reselecting an existing Cell but caused confusion.

Solution: The App now checks for the Cell Controller being configured in the new SQL DB and if not resets to the first one that is, it also resets to A0C0 if you clear all configuration data.

BEFORE – ESYS SQL:



AFTER – HTI SQL:





5) App needed to translate EPP Conveyor Types from SQL to APP Types

Issue: After importing SQL DB Conveyor types are not always translating properly.

EPP SQL:

- 1 Fixed Position Stop
- 2 Accumulating Line Stop
- 3 Accumulating Stop with Carrier ID
- 4 AGV ALS Controlled Position
- 5 FPS 70 80 90
- 6 AGV ALS Timed Position
- 7 ALS w/Position

MicroCODE APP:

- 1 = Fixed Position Stop (FPS) conveyor
- 2 = Accumulating Lane Stop (ALS) conveyor
- 3 = Automated Guided Vehicles (AGVs)
- 4 = Sub-Assembly Interface (SAIs)

Solution: The App now translated as follows:

- 1 FPS
- 2 ALS
- 3 ALS
- 4 AGV
- 5 FPS
- 6 AGV
- 7 ALS

NOTE: Sub-Assembly Interface (SAI) is not supported by the EPP Server Configuration

6) App was not displaying Conveyor and I/O Type from SQL when offline

Issue: The App was showing "???" for both Conveyor Type and I/O Type when Offline, when both are known from the EPP SQL DB.

Solution: The configured values are now shown when offline.

7) GEPICS Build Data Viewer was overloading with BLANK LINES

Issue: With Blank Lines loaded all the sort function in the Data Viewer were basically destroyed.

Solution: The GEPICS Data View is not terminated at the last loaded row and sorting is back to normal.

8) App was loading slow

Issue: With a large SITE configuration there a noticeable delay after starting the .EXE before the App 'Splash Screen' appeared.

Solution: The 'Splash Screen' now appears immediately to show a response to starting the .EXE. Then the configuration loads in the background.



New in Version v2.0.1 b (4)

The following features were added to **MicroCODE Control (EPP)** in this Release:

1) Conveyor display is now customized for EPP Interlocks

SAT activities reveal that its more important to see 'OK TO RUN' than the 'COMM OK Out' (which does not actually exist in the EPP CNV Interface).

Requirement: Display 'OK to RUN' for EPP instead of 'COMM.OK Out'.

Implementation: And so...

MicroCODE Control (EPP) - v2.0.1 (3)

CONVEYOR Interface (CNV) View

Up Down Hold

Unit	COMM.OK In	OK to RUN	Entering	Pre-Leave	Leaving	@FPS	Blocked	EPAs	Stopped	Production
CNV: 0	Green	Grey	Green	Grey	Grey	0	Yellow	0	Red	Grey
CNV: 1	Green	Grey	Green	Grey	Grey	0	Yellow	0	Red	Grey
CNV: 2	Green	Grey	Green	Grey	Grey	0	Yellow	0	Red	Grey
CNV: 3	Green	Green	Green	Grey	Grey	99%	Yellow	0	Red	Grey
CNV: 4	Green	Green	Green	Grey	Grey	99%	Yellow	0	Red	Grey

Legend:

- Stopped
- EPx Stop(s)
- Run Stop(s)
- SIM Stop(s)
- Production
- Andon Stop(s)
- End-of-Travel(s)
- SIM Bypass(s)

MicroCODE Control (EPP) - v2.0.1 (3)

F1 ABOUT... F2 PROFILES: SET-UP... F3 CONNECT: ONLINE F4 TZN SIM: NO TZN F5 TZN SIM: NO TZN F6 TZN SIM: NO TZN F7 TZN SIM: NO TZN F8 TZN SIM: NO TZN F9 F10 F11

Status: Ready 04-Mar-21 04:09:30 PM 70.889 MB

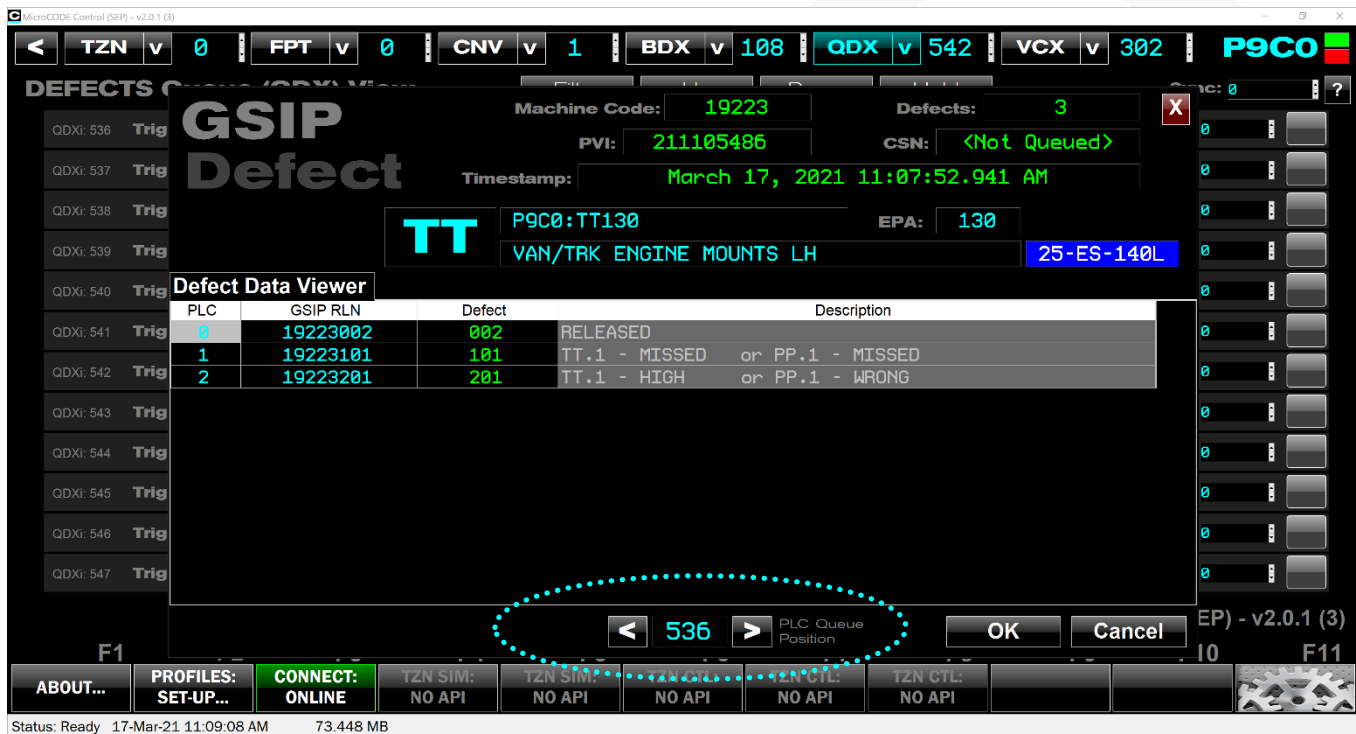


2) You can now move thru the DEFECT, TRACE, and GEPICS Queues from within the Viewers

SAT activities reveal that it's a nice convenience to be able to scan back and forth thru these queue data items without having to exit, select a different queue position, and re-open the Viewer.

Requirement: Allow the user to scan thru the PLC Data Queue without leaving an open Viewer.

Implementation: And so... you can now see the PLC Queue position in the Viewer, and each Viewer has a NEXT and PREVIOUS button that support scanning the PLC Queues (with 'wrap-around').



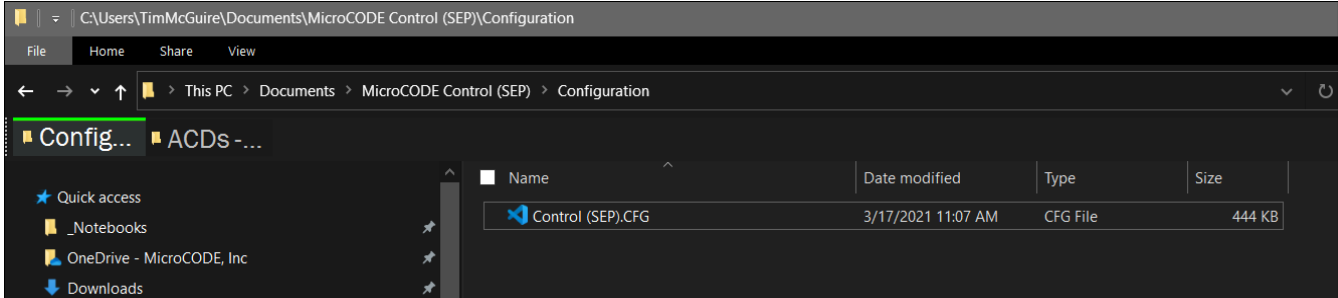
NOTE: The current implementation is limited to walking thru the DEFECTs (or TRACE ITEMS) currently on the underlying screen (and wraps within those items).

Future versions will allow unlimited walking thru the PLC Queue. Until then close the Viewer and Page UP/DOWN to get the Defects/Traced Components of interest on the Screen first.

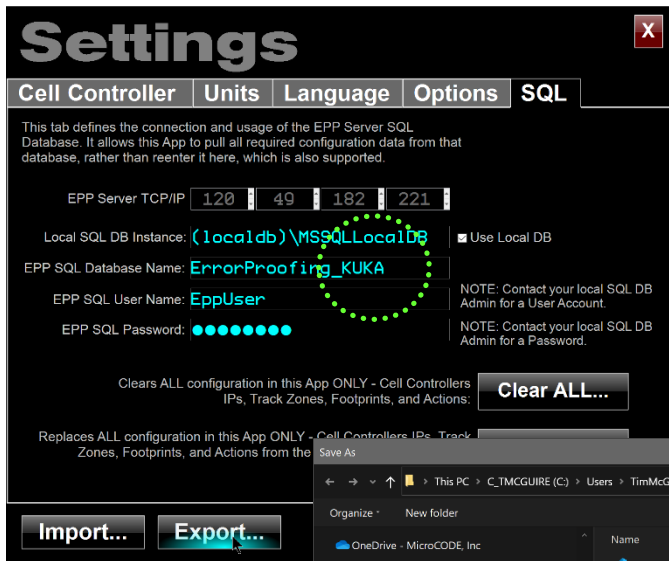


3) You can now EXPORT App Configuration

You were always able to copy and import the App Settings from the file from your **Documents** folder:



But it was not intuitive. So, we have added an explicit button to do so and that also allows you to give it a specific name in the process for reference:

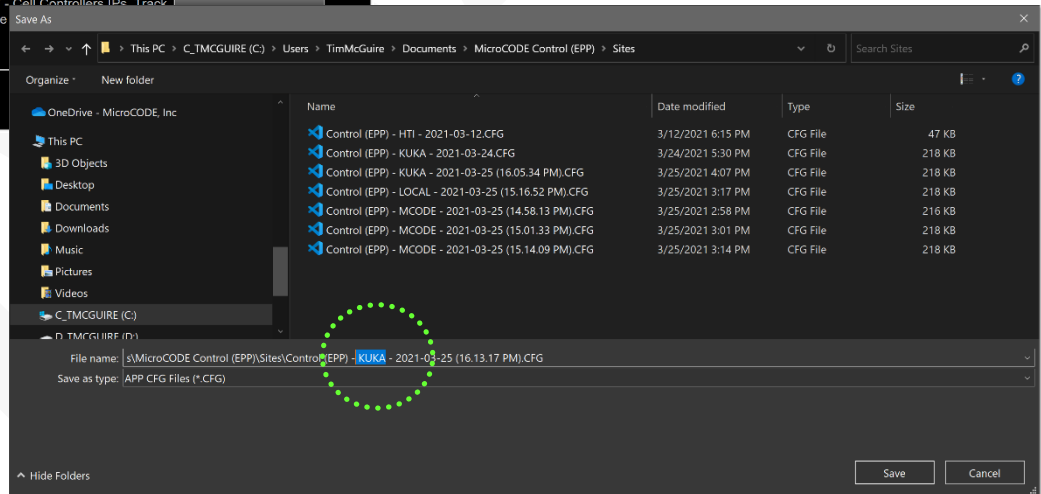


NOTE: A new folder is created to support Configuration Import/Export called "Sites".

This is there to facilitate switching between customer Sites and is keyed to the name of the SQL DB you last installed. (But the names can be anything you desire).

This 'Sites' folder will be used in a future Release to enable a new Site Switching feature... stay tuned.

Naming Convention:
Control (APP) – SITE – YYYY-MM-DD...



4) You can jump to any Conveyor in your SITE directly from the SITE Screen

Supporting Plant Startups reveal that “everything has three (3) names”. (Just like Father-Son-Holy Ghost).

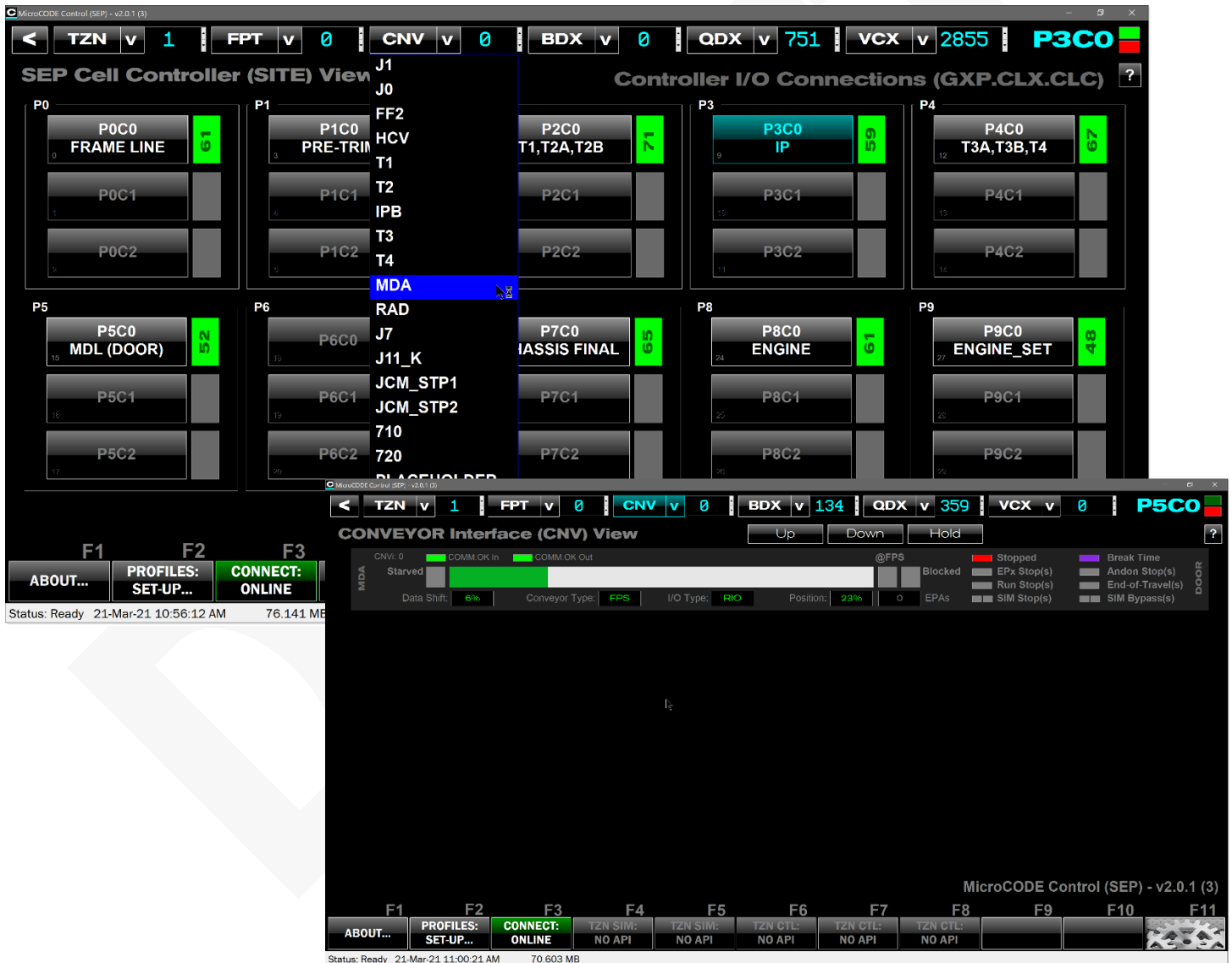
On a Plant radio mechanic people refer to Conveyor names (J0, J1, K11, etc.).

IEs and Error Proofing people refer to Track Zone names (TRIM 1, TRIM 2, CHASSIS 3, etc.).

And Group and Team Leaders refer to Footprint names (17-T1-123L, or just 123 Left).

Requirement: When working support for a large Site allow the App user to quick navigate to anything Production refers to by the names they use, Conveyor, Track Zone, or Footprint.

Implementation: And so... you can now select any Footprint group directly and the App will jump you directly into the appropriate Cell Controller and Track Zone.





5) You can jump to any Footprint group in your SITE directly from the SITE Screen

SAT activities reveal that it's a nice convenience to be able to scan back and forth thru these queue data items without having to exit, select a different queue position, and re-open the Viewer.

Requirement: When working support for a large Site Production often calls for support by Conveyor name vs. Track Zone and the SEP/EPP Staff must translate this navigate.

Implementation: And so... you can now select any Conveyor directly and the App will jump you directly into the appropriate Cell Controller.

The screenshot displays the MicroCODE Control (SEP) v2.0.1 (3) interface. The top section, 'SEP Cell Control', shows a grid of conveyor cells (P0 to P9) with various components like TRIM, CHAS, ENGINE, and COCKPIT. A list on the left shows conveyor names and IDs, with '25-C2-123' highlighted. The bottom section, 'TRACK ZONE (TZN) View', shows a detailed view of a specific conveyor (P2C1) with a grid of footprint groups (25-C2-123L to 25-C2-128R). Each footprint group has fields for PVI, CSN, SVI, CID, VIN, and Model, along with status indicators like 'Incomplete', 'Carrier', 'Job Detect', 'No Read', 'Set-In', and 'Set-Out'. The interface includes various navigation buttons and a status bar at the bottom.

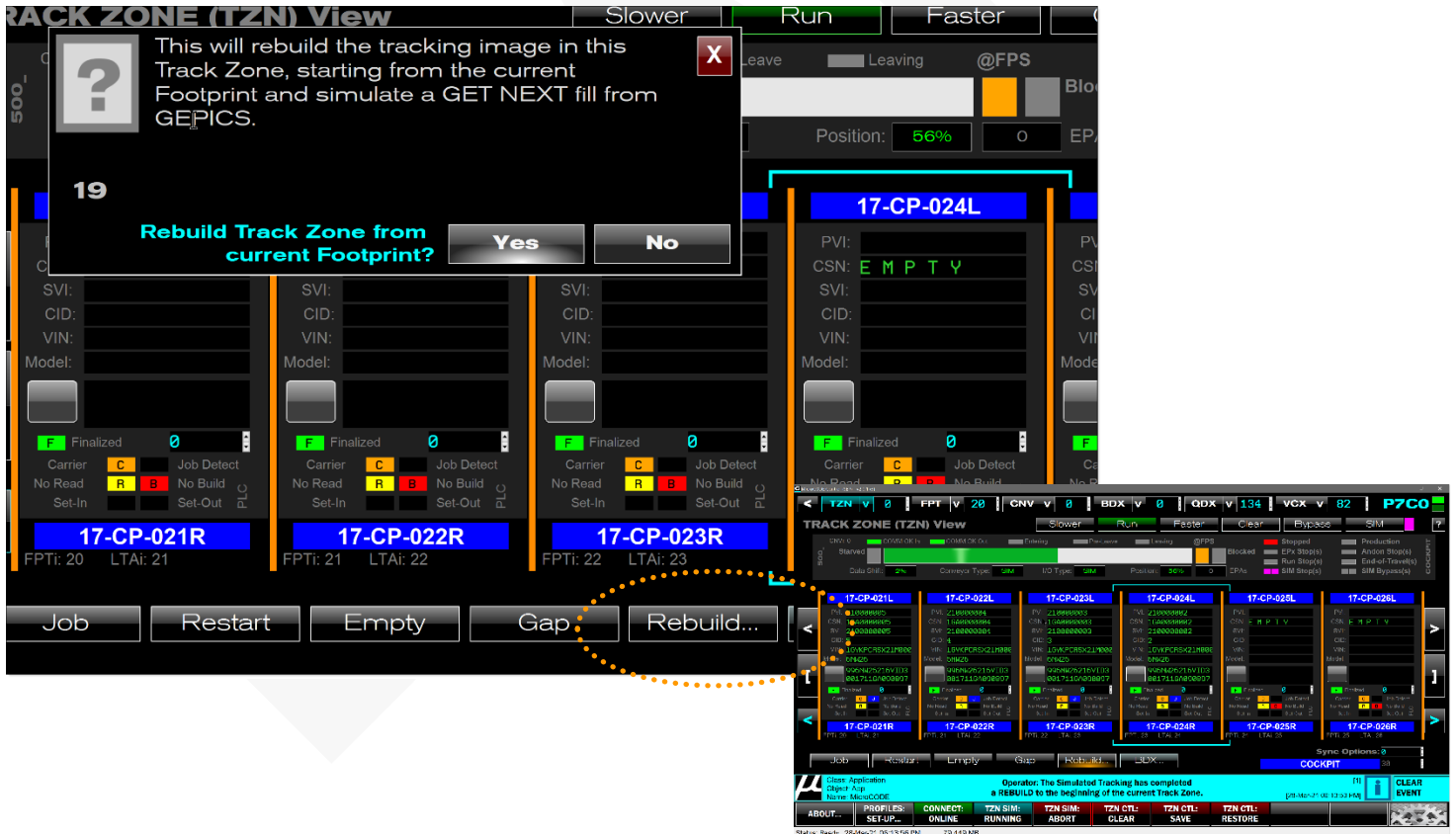
6) ESCAPE KEY is now supported as a BACK button or Screen Closure Command

This is just a User convenience to fill out the standard supported commands in the U.I.

7) REBUILD Track Zone has been implemented

You can now fill a Track Zone with simulated GEPICS Jobs built from your JOB PROFILES.

- 1) Place the 'Job Focus' (the Cyan Outline) on the Footprint where you want to start.
- 2) Click the **Rebuild...** button. (Only available in Simulation Mode).
- 3) Click "Yes" when ready.
- 4) The App will place simulated Jobs in every Footprint going back toward the beginning of the Track Zone. These will be in CSN order starting with eth highest CSN in the Track Zone at the time of the Rebuild +1.



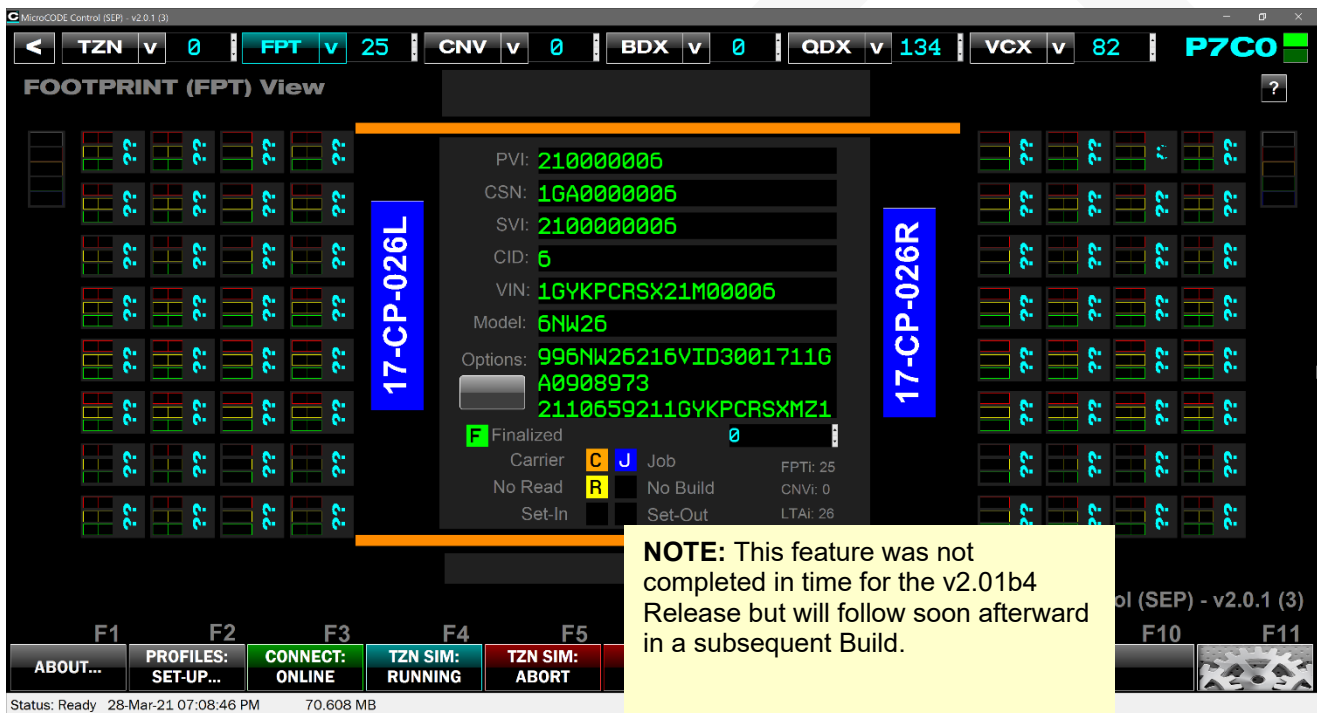


8) Task / Actions Status display in FOOTPRINT VIEW

During SATs the User's require quick access to the state of a Footprints Tasks (EPP) or Actions (SEP). This new feature will give the that with a simple display and click on the MicroCODE KeyStack (a new Screen Control representing the EPP Lighted keyswitch and SEP Lightstack).

Requirement: When executing SATs in a Footprint give the Users a quick overall of all the Tasks / Actions in that Footprint with direct access to the status details.

Implementation: And so... now you see up to sixty-four (64) Tasks or Actions in Footprint, all on one screen.





Corrected in Version v2.0.1 b (4)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) Issues with simulated FPS Conveyors

After all the App revisions to support the changing EPP PC code for AGV and ALC Conveyors issues crept into the already completed FPS Conveyor support.

Issue: APP simulation no longer worked for stand FPS Conveyors in either the SEP or EPP versions of the App.

Solution: Conflicts between FPS Conveyor and ALS/AGV Conveyors were resolved in the App.

NOTE: Be sure to use the most recent version of the APP for both SEP and EPP and the matching version of the PLC API.

The PLC API for EPP is not like the SEP version, you import it and you do NOT modify it at all.

You do have to add *not* simulated contacts into the consumption of the Conveyor Tag into **FromConveyor** and around any RFID Tab verification on a Footprint-by-Footprint basis.

See the update **User Guide** for completed directions.

MicroCODE Control – for the GM Scalable Error Proofing (EPP) System

STEP 2: Add an API simulation to block consuming real Tags in Footprint Programs.

This step is required because the EPP PLC code directly uses the Conveyor Inputs from a Consumed Tag in the Footprint level code, rather than an intermediate generic object.

In every FPnnn Program's **B001_MapInputs** add NOT API.CNV[n].Simulated.Enabled to block being the Consumed tag into the temporary Tag "fromConveyor".



New in Version v2.0.1 b (5-7)

The following features were added to **MicroCODE Control (EPP)** in this Release:

1) The 'Job Focus' Cursor now displays PVI

To make the 'Job Focus' more intuitive the PVI the User last selected is now displayed in the Cyan colored wire frame. This focus follows the job as it moves in the Track Zone.

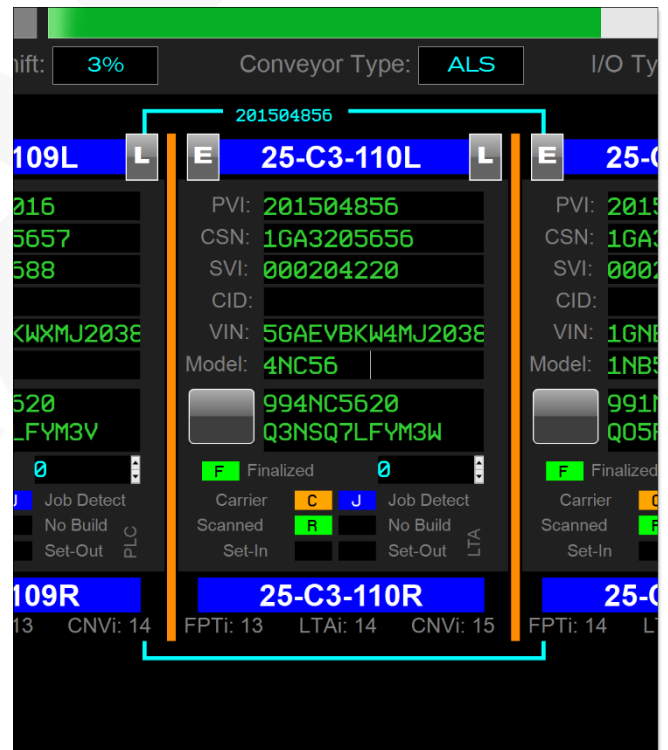
Requirement: make it clear to the User when he has a Job in 'focus' and when he does not.

Implementation: The 'Job Focus' wire frame is now gray when unfocused, and cyan when focused on the specific PVI, and that PVI is displayed in the wire frame.

UNFOCUSED, stationary cursor:



FOCUSED, follows PVI:

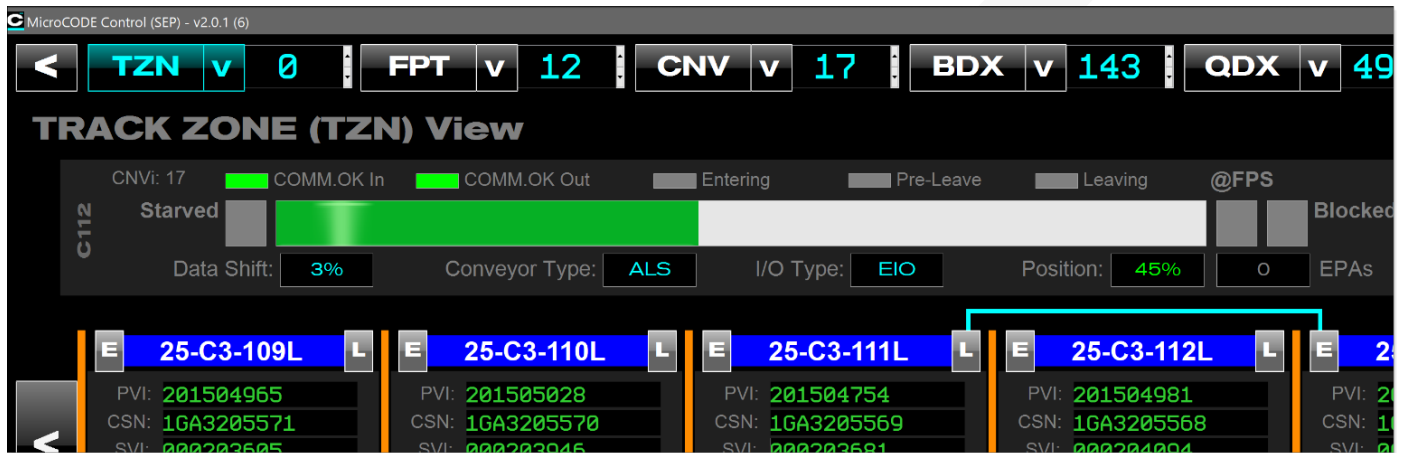


2) Clarify Conveyor Data

To make the Conveyor configuration data more recognizable it is now displayed in cyan like all other settings and configuration (even though is it read from the PLC).

Requirement: make it clear to the User what type of data is being displayed.

Implementation: Show all configuration data in cyan.

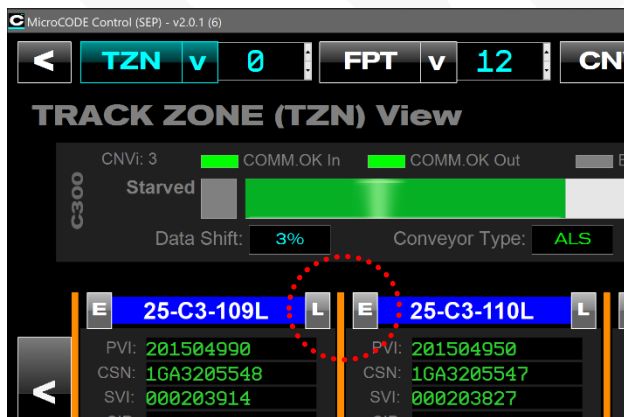


3) Clarify when Entering/Leaving Commands can be used

To make it clear that the simulation commands for Conveyor 'Entering/Leaving' on Stop Stations.

Requirement: Make sure Users know when they can use Conveyor Simulation commands.

Implementation: Disable the 'Entering/Leaving' commands when there is no API in the PLC and when the App is not in simulation mode.





Corrected in Version v2.0.1 b (5-7)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) Issues with simulated FPS Conveyors

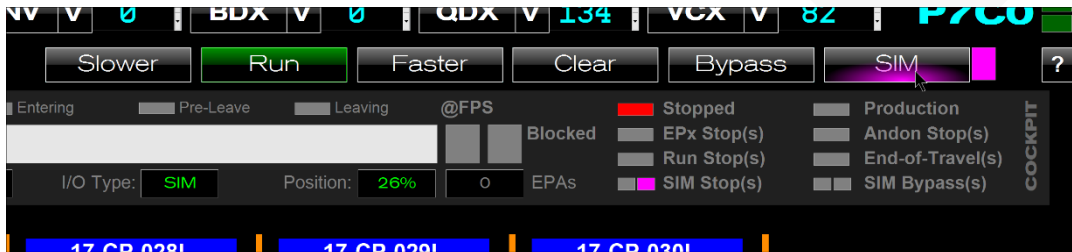
The FPS Conveyor controls were not enabled when in Simulation mode, even though they appeared normal.

Issue: Build v2.0.1b4 intended to disable the Conveyor Simulation commands when on an ALS/AGV Conveyor or when in REAL Interface mode.

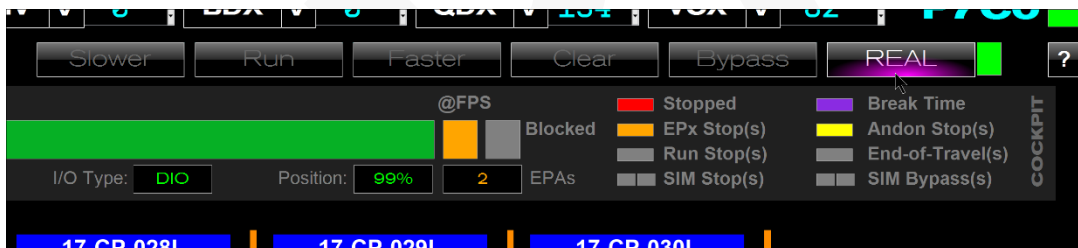
But they were disabled all the time. Whoops. Rushed that Build Release.

Solution: The intended behavior has been implemented properly.

On an FPS Conveyor with the Conveyor in SIM Mode:



On an FPS Conveyor with the Conveyor in REAL Mode – or any ALS/AGV Conveyors:



Major Test Cases:
EPP v2.0.1 b (1-4)
SEP v2.0.1 b (1-4)

Fixed Position Stop (FPS)

Task: FPS Conveyor Simulation (SEP)

- **SITE:** SPRING HILL
- **SSP:** LEGACY
- **CELL:** P7C0
- **TZN:** COCKPIT

TRACK ZONE (TZN) View

Slower Stop Faster Clear Bypass REAL

COMM.OK In COMM.OK Out @FPS

Starved Blocked EPx Stop(s) Break Time

Data Shift: 2% Conveyor Type: FPS I/O Type: DIO Position: 99% 0 EPAs Run Stop(s) Andon Stop(s)

Finalized Job Detect

Carrier No Read No Build

Set-In Set-Out

17-CP-001L 17-CP-002L 17-CP-003L 17-CP-004L 17-CP-005L 17-CP-006L

PVI: 210000015 CSN: 1GA0000015 SVI: 210000015 CID: 15 VIN: KPCRSX21M0015 Model: 6N425 996N426216VID3 001711GA090897

17-CP-001R 17-CP-002R

FPT: 0 LTAI: 1 FPT: 1 LTAI: 2

Job Restart Empty

F1 ABOUT... F2 PROFILES: SET-UP... F3 CONNECT: ONLINE F4 TZN SIM: RUNNING

Status: Ready 25-Mar-21 08:50:32 PM 77.526 MB

TRACK ZONE (TZN) View

Slower Stop Faster Clear Bypass SIM

COMM.OK In COMM.OK Out Entering Pre-Leave Leaving @FPS

Starved Blocked EPx Stop(s) Break Time

Data Shift: 2% Conveyor Type: SIM I/O Type: SIM Position: 90% 0 EPAs Run Stop(s) Andon Stop(s)

Finalized Job Detect

Carrier No Read No Build

Set-In Set-Out

17-CP-005L 17-CP-006L 17-CP-007L 17-CP-008L 17-CP-009L 17-CP-010L

PVI: 210000091 CSN: 1GA0000091 SVI: 210000091 CID: 91 VIN: 1GVKPCRSX21M008 Model: 6N425 996N426216VID3 001711GA090897

17-CP-005R 17-CP-006R 17-CP-007R 17-CP-008R 17-CP-009R 17-CP-010R

FPT: 4 LTAI: 4 FPT: 5 LTAI: 5 FPT: 6 LTAI: 6 FPT: 7 LTAI: 7 FPT: 8 LTAI: 8

Job Restart Empty Gap BDX...

Sync Options: 8

COCKPIT

Class: Application Object: App Name: MicroCODE Operator: The Simulated Tracking has CREATED a NEW JOB by generating it from your PROFILE into this Footprint. [27-Mar-21 12:22:24 PM] CLEAR EVENT

ABOUT... PROFILES: SET-UP... CONNECT: ONLINE TZN SIM: RUNNING TZN SIM: ABORT TZN CTL: CLEAR TZN CTL: SAVE TZN CTL: RESTORE

Status: Ready 27-Mar-21 12:22:26 PM 76.367 MB



Control.NET™ App for GM SEP

Release Notes Alpha Release: 3.0.0 a (1)



Task: FPS Conveyor Simulation (EPP)

- SITE: PLANT ZERO
- SSP: KUKA
- CELL: **GFF1TZ1**
- TZN: FINAL

The screenshot displays the MicroCODE Control (EPP) v2.0.1 (4) interface. At the top, a status bar shows various conveyor and zone indicators: TZN v 0, FPT v 0, CNV v 0, BDX v 0, QDX v 499, VCX v 0, and A4CO. Below this, the TRACK ZONE (TZN) View is active, showing a progress bar and status indicators like 'COMM OK In' and 'OK to RUN'. The main area displays six conveyor stations (27-F1-171L to 27-F1-176L) with their respective parameters: PVI, CSN, SVI, CID, VIN, and Model. A 'CONVEYOR Interface (CNV) View' window is overlaid, showing 'Position: 77%' and 'I/O Type: SIM'. At the bottom, there are function buttons for 'Job', 'Restart', 'Empty', and 'Gap', along with a status bar showing 'Status: Ready 26-Mar-21 03:45:46 PM' and '70.221 MB'.

This screenshot shows the MicroCODE Control (EPP) v2.0.1 (4) interface with a message overlay. The TRACK ZONE (TZN) View is visible, showing six conveyor stations (27-F1-171L to 27-F1-176R) with their parameters. A message box is displayed in the center: 'Operator: The Simulated Tracking has CREATED a NEW JOB by generating it from your PROFILE into this Footprint. [26-Mar-21 04:03:20 PM]'. Below the message, there are buttons for 'Job', 'Restart', 'Empty', 'Gap', and 'ODD...'. The status bar at the bottom shows 'Status: Ready 26-Mar-21 04:03:20 PM' and '79.495 MB'. The interface also includes function buttons for 'ABOUT...', 'PROFILES: SET-UP...', 'CONNECT: ONLINE', and 'TZN SIM: RUNNING'.



Accumulating Lane Stop (ALS)

Task: ALS Conveyor Simulation (SEP)

- **SITE:** SPRING HILL
- **SSP:** LEGACY
- **CELL:** PnCn
- **TZN:** VAC

[USE JANUARY ACD & SQL DB – CHASSIS 1/2]

DRAFT



Task: ALS Conveyor Simulation (EPP)

- **SITE:** PLANT ZERO
- **SSP:** KUKA
- **CELL:** GRC1TZ1
- **TZN:** CHASSIS MOD1

[GET CURRENT KUKA ACD & SQL DB]

DRAFT



Vertically Adjusted Carriers (VAC)

Task: VAC Conveyor Simulation (SEP)

- **SITE:** LANSING DELTA TWP
- **SSP:** LEGACY
- **CELL:** P1C2
- **TZN:** VAC

[USE FEBRUARY ACD & SQL DB – CHASSIS P1C2]



Task: VAC Conveyor Simulation (EPP)

- N/A – Not supported at this time

DRAFT



Automated Guided Vehicles (AGV)

Task: AGV Conveyor Simulation (SEP)

- **SITE:** SPRING HILL
- **SSP:** LEGACY
- **CELL:** PnCn
- **TZN:** AGV

[GET CURRENT ESYS ACD & SQL DB]

DRAFT



Task: AGV Conveyor Simulation (EPP)

- **SITE:** PLANT ZERO
- **SSP:** KUKA
- **CELL:** GRC1TZ1
- **TZN:** CHASSIS MOD1

[GET CURRENT KUKA ACD & SQL DB]

DRAFT



Sub-Assembly Stations (SAI)

Task: Sub-Assembly Stations (SEP)

- N/A – Handled by SEP Actions App

Task: SAI Conveyor Simulation (EPP)

- SITE: PLANT ZERO
- SSP: ESYS
- CELL: **GVR1TZ1**
- TZN: SUB-ASSEMBLIES
- (3) Footprint TZN
- (1) Footprint TZN
- (4) Footprint TZN

The screenshot shows the MicroCODE Control (EPP) v1.0.4 (1) interface. The top bar displays various conveyor and station identifiers: TZN 2, FPT 4, CNV 7, BDX 1, QDX 381, VCX 0, and A3C2. The main area is titled "TRACK ZONE (TZN) View" and includes control buttons like "Slower", "Stop", "Faster", "Clear", "Bypass", and "SIM". A progress bar shows "Data Shift: 3%" and "Conveyor Type: SAI". Below this, four job cards are visible, each with fields for PVI, CSN, SVI, CID, VIN, and Model. The bottom section shows a "Job" button and "Restart" and "Empty" options. A status bar at the bottom indicates "Status: Ready 25-Feb-21 08:45:40 PM 84.339 MB".



New in Version v2.0.1 b (8)

The following features were added to **MicroCODE Control (EPP)** in this Release:

NO NEW FEATURES in (8).

Corrected in Version v2.0.1 b (8)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) Unable to import a SEED JOB

After upgrading to v2.0.1b7—and reassociating GEPICS Formats—users are unable to import an existing SEED JOB.

Issue: Import a Seed Job does not immediately display that Job Data. But users rightly expect to see that data immediately after the import.

Solution: Immediately after import a new Seed Job it is display in the GEPICS Build Data Viewer.

New in Version v2.0.1 b (9)

The following features were added to **MicroCODE Control (EPP)** in this Release:

1) Queue can now be searched for DATA

The 'FILTER' command now includes the ability to search the data associated with a queue entry for a specific sub-string.

Requirement: Find bad trace data within a 2,000 entry Trace Data Queue.

Implementation: Added "Queued Data" filter and the code for the GEPICS Buffer and Trace Queue now support this during a search.

The screenshot displays the MicroCODE Control (SEP) v2.0.1 (9) interface. The top section shows the 'TRACE Queue (VCX) View' with a list of entries. An 'EPx Viewer Filter' dialog is open, showing the 'Queued Data' filter selected. The 'GEPICS Trace' window is also visible, showing the 'Trace Data Viewer' with a table of data.

TRACE Queue (VCX) View

VCX	PVI	CSN	MC	Parts	Trace
46	211152242	<SEPActions>	18098	1	VPPS=<V121
47	211152516	<SEPActions>	18308	1	VPPS=<V941
48	211152589	<SEPActions>	18212	1	VPPS=<V945
49	211152399	<SEPActions>	18099	2	S=<V111
50	211152449	<SEPActions>	18099	2	S=<V121
51	211152628	<SEPActions>	18099	2	S=<V945
52	211152316	<SEPActions>	18099	2	S=<V111
53	211152491	<SEPActions>	18099	2	S=<V121
54	211152453	<SEPActions>	18099	2	S=<V941
55	211152120	<SEPActions>	18212	1	VPPS=<V945
56	211151669	<SEPActions>	18099	2	S=<V111

EPx Viewer Filter

- PVI
- CSN ...without Prefix
- Machine Code
- Queued Data

GEPICS Trace

Station Name: 25EN222L System Code: GEPTTRACE
Machine Code: 18099 Record Key: Y11100000
PVI: 211153006 CSN: <SEPActions> Parts: 2
Timestamp: August 17, 2021 09:18:24.569 AM
CE P8C0:CE011 EPA: 11
VCVS ALL ENGINES SCAN 25-EN-222L

Trace Data Viewer

SEP	VPPS / Component	Part No.	DUNS / Vendor	Trace Data	Payload Data
0	Y111000000000X	P24000488	12V90456363Z	TF1211241MNMX1038	...
1	Y000000000000X				...



Corrected in Version v2.0.1 b (9)

The following defects were fixed in **MicroCODE Control (EPP)** in this Release:

1) API Command Warnings (SEP/EPP)

The internal App time allowed for a command to execute was too short for heavily loaded SEP PLCs.

Issue: The User would see erroneous warning about the API being busy.

Solution: Increase Command time allowance from 500ms to 750ms.

2) Incorrect new Job CSNs (SEP)

The App would generate odd CSNs depending on when the User last switches Track Zones in a multi-Track Zone Cell Controller.

Issue: CSNs being generated would jump from 0002, 0003 to 0369 for example.

Solution: The App was not always reading the new Track Zone parameters when switch from one to another.

3) Active JOB FOCUS issues (SEP/EPP)

In a Multi-Track Zone PLC, the JOB FOCUS cursor causes Job from the focused Track Zone to appear on the screen after changing to a different Track Zone.

Issue: With the JOB FOCUS a Job in Track Zone 0, switching to Track Zone 1 causes a refresh after the initial Tracking is displayed properly where the Jobs in Track Zone 0 re-appear. The JOB FOCUS PVI is Track Zone specific and was not being maintained during Track Zone switching.

The App was 'chasing' the focused Job from Track Zone 0 and finding it in the newly selected Track Zone.

Solution: The JOB FOCUS PVI is Track Zone specific and was not being maintained during Track Zone switching. The App was corrected to maintain the Track Zone specific Job Focus properly.

TZN 0 – Focused on PVI 00028:



TZN 1- Focused on PVI 00043:





4) BDX/QDX/VCX Displays (SEP/EPP)

The indices display in unoccupied rows in the Queue was incorrect. And—during paging—jumped the display twice before completing refresh properly.

Issue: Then the User was presented empty Queue space in the QDX Viewer the QDX indices were not always correct.

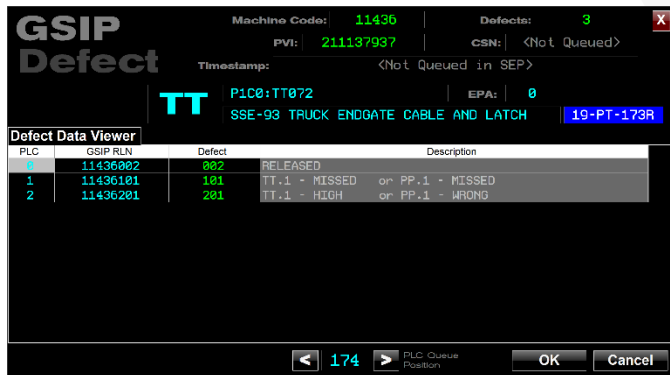
Solution: The Viewer display was corrected to refresh all rows including empty Queue positions. And, to initialize and fill the display the same way, with the user focused on the center of the Viewer.

5) QDX Timestamp Display (SEP)

The SEP PLC Queue for GSIP Defects does not contain a Timestamp.

Issue: The App was displaying the time it read the Defect from the PLC Queue, and this was misleading to Users.

Solution: The Viewer display was corrected to show “Not Queued”. Note that CSn is not queued either, both are now in gray.



6) AGV Handling (SEP)

The SEP SIM Mode from the Control App was now working for AGC Conveyors.

Issue: The App was not toggling AGV Conveyor interfaces into and out of SIM Mode properly.

Solution: The App now uses only “I/O Type” to control SIM Mode recognition because “Conveyor Type” is a hybrid of an integer value and control bits for AGVs.

Note: This required an update to the App's PLC API, and this must be updated along with the App.



New in Version v2.0.2 b (1-2)

The following features were added to **MicroCODE Control (SEP)** in this Release:

No new features

Corrected in Version v2.0.2 b (1-2)

The following defects were fixed in **MicroCODE Control (SEP)** in this Release:

1) ALS/AGV [E] / [L] Commands don't work first time (SEP/EPP)

The first click of the ENTERING [E] or LEAVING [L] commands in a Footprint do not work. The second time they do.



Issue: The Footprint requires 'focus' for the Commands to work properly. This changes with the addition of the new 'Job Focus' feature.

Solution: The ENTERING [E] and LEAVING [L] commands now move Job Focus first and then execute, all on the first click.

2) ALS/AGV [E] / [L] Commands don't work properly (SEP)

The ENTERING and LEAVING Commands were not generating interlocks properly to the SEP PLC.

Issues:

- 1) The conveyor type AGV was not being detected properly from the SEP SQL DB.
- 2) The interlocks in the PLC were overriding those from the Control App.

Solutions:

- 1) The PLC control bits now override the SQL DB type.
- 2) The ENTERING [E] and LEAVING [L] commands are not protected when the external Control App is in use (CNV[n].CTL.sp_80 / .ExternalApp).

NOTE: This requires two (2) updates to the SEP PLC:

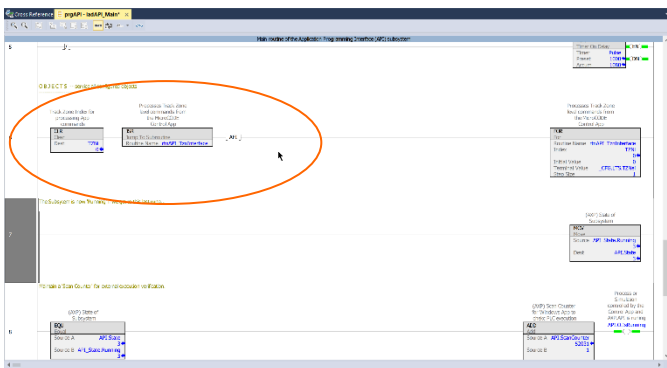
- 1) Update the **prgAPI** to v2.0.2b1
- 2) Update the SEP CNV routine: **ladCNV_ioXIO_Simulated** from the App install package.

3) Simulation Commands on anything but Track Zone 0 were not working

Users could start simulation in Track Zone 0 but switching to Track Zones 1-N and attempting a “Start Simulation” failed.

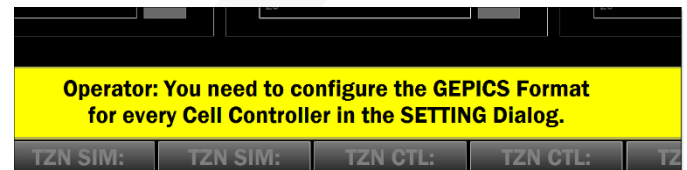
Issue: Debug code was left active in the **prgAPI: ladAPI_Main** that was shipped with the last release.

Solution: Remove the debug code shown below (the first three instructions) or re-install the **prgAPI** from this release.



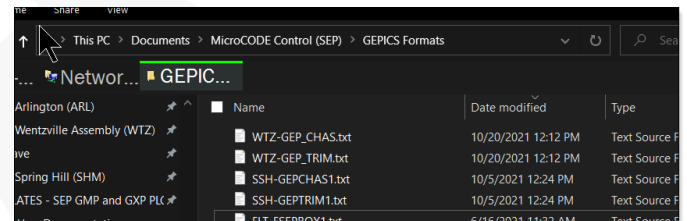
4) User receiving “Select GEPICS Format” after already having done so

After a user has selected the GEPICS Format for every Cell Controller he still sees the error...



Issue: Recent version of the App eliminated holding references to the original location of the GEPICS Format file and instead gather the selected files into the App private storage and hold only the file name. But the last release was not copying the files to that private storage.

Solution: The App now gathers copies of the GEPICS definitions files to its private storage properly.





New in Version v2.0.2 b (3)

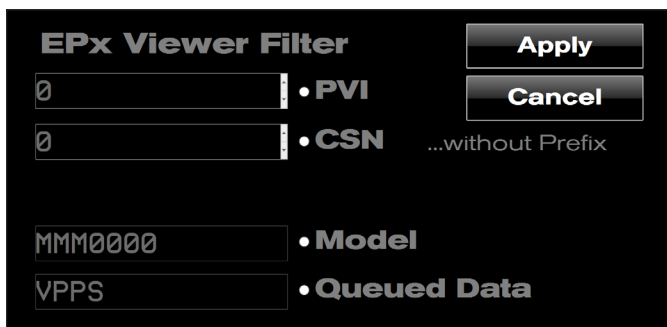
The following features were added in this Release:

1) GEPICS, GSIP, and TRACE data can be filtered by content

Requested came for a data filter on these in addition to the original PVI, CSN, etc.

Requirement: Allow filtering the queue and buffer searches by data like RPO codes or Trace Data.

Implementation: A new field was added to the FILTER dialog box: **“Queued Data”**. This can be set to any data that may appear in the objects in the queue or buffer you are looking at.



GEPICS: RPO Codes, PART NUMBERS, VPPS Codes, etc.

GSIP: None supported in this release.

TRACE: VPPS, Serial Numbers, Vendor Codes, etc.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Search GEPICS for RPO – **Passed**.
Test Case 2: Search GEPICS for PART – **Passed**.
Test Case 3: Search GEPICS for VPPS – **Passed**.

Test Case 4: Search TRACE for VPPS – **Passed**.
Test Case 5: Search TRACE for SN – **Passed**.
Test Case 6: Search TRACE for VENDOR – **Passed**.

Corrected in Version v2.0.2 b (3)

The following defects were fixed this Release:

1) The SEP PLC API needed documentation corrections for proper installation

It was very easy to miss a required addition to the SEP standard logic at the conveyor interface level.

Issue: Missing a single contact in the SEP code would cause the MicroCODE App to appear to fail.

Correction: Correct placement of the MicroCODE API being in simulation mode were required. The App's USER GUIDE, **Appendix A: The MicroCODE API Logix Program** was updated with clarifications.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: GAP command – **Passed**.

Test Case 2: JOB command – **Passed**.

Test Case 2: ENTERING command – **Passed**.

Test Case 3: LEAVING command – **Passed**.



New in Version v3.0.0 a (1)

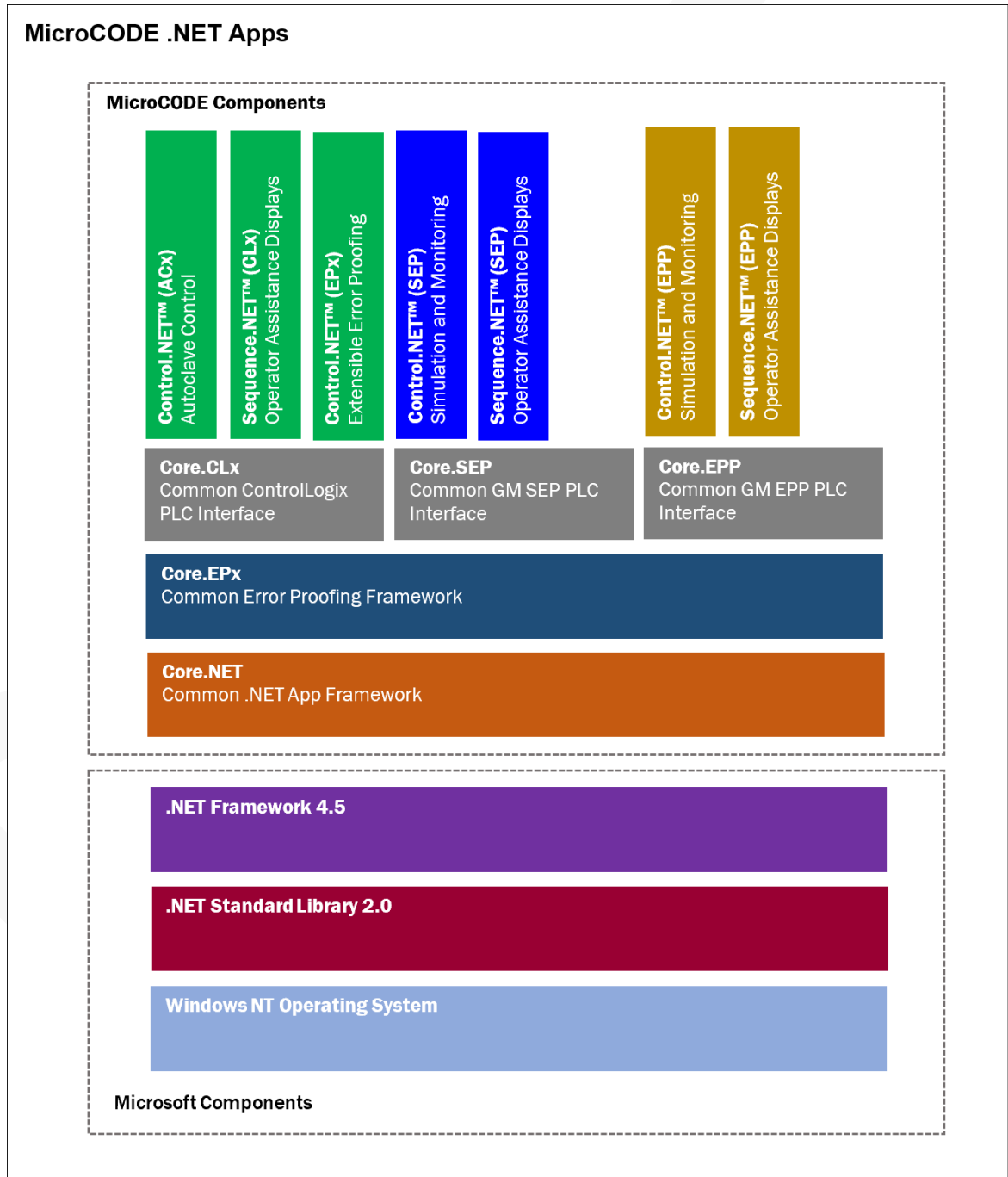
The following features were added in this Release:

1) Modern App Architecture

All the MicroCODE Control.NET™ and Sequence.NET™ Apps now share a commonized, modern, modular construction; sharing as many proven components as possible.

Requirement: Prepare to expand MicroCODE's App offerings while minimizing support requirements and leveraging all proven code from the past six (6) years.

Implementation: All of MicroCODE's manufacturing Apps now build on **.NET 4.5** (and they run on .NET 5 and .NET 6). .NET 4.5 is the lowest version currently being deployed by General Motors.





2) Support App Scaling and Resolution on any monitor of any DPI

The Control App needs to support scaling and resizing properly on all possible Windows display settings.

Requirement: There are many different monitor sizes and dots-per-inch (DPI) resolutions available.

Implementation: The App was originally designed to be 'Resolution' independent, but Windows also supports 'Scaling', and this forced a redesign of the MicroCODE services that provide display independence.

This resulted in MicroCODE's **Resizer**, an open-source component available on GitHub under an MIT License. This utility code is now used in all our Apps to provide true monitor independent.

GitHub Repository:
<https://github.com/MicroCODEIncorporated/mcode-resizer>

This is the MIT license that must accompany all Apps that use this component:

```
MIT License: MicroCODE.Resizer
Copyright (c) 2019-2023 Timothy J McGuire, MicroCODE, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Resolutions

Test Case 1: Run App on a **4K:** 3840x2160 pixels (also known as Ultra High Definition or UHD) Monitor and check all displays for proper layout and readability – **Passed.**

Test Case 2: Run App on an **FHD:** 1920x1080 pixels (also known as Full HD or FHD) Monitor and check all displays for proper layout and readability – **Passed.**

Test Case 3: Run App on a **HD:** 1280x720 pixels (also known as High Definition or HD) Monitor and check all displays for proper layout and readability – **Passed.**

Test Case 4: Run App on an **XGA:** 1024x768 pixels (also known as XGA) Monitor and check all displays for proper layout and readability – **Passed.**

Test Case 5: Run App on a **SVGA:** 800x600 pixels (also known as SVGA) Monitor and check all displays for proper layout and readability – **Passed.**

Resolution s changed in Windows 10 under Display Settings.

3840 × 2160 (Recommended)	
2560 × 2048	
2560 × 1920	1600 × 900
2560 × 1600	1440 × 900
2048 × 1536	1400 × 1050
2048 × 1152	1366 × 768
1920 × 1440	1360 × 768
1920 × 1200	1280 × 1024
1920 × 1080	1280 × 960
1856 × 1392	1280 × 800
1792 × 1344	1280 × 768
1680 × 1050	1280 × 720
1600 × 1200	1280 × 600
1600 × 900	1152 × 864
1440 × 900	1024 × 768
	800 × 600

Moving

Test Case 6: Drag the App from an 4K to an FHD Monitor and check all displays for proper layout and readability – **Passed.**

Test Case 7: Drag the App from an FHD to a 4K monitor and check all displays for proper layout and readability – **Passed.**

Resizing

Test Case 8: Resize/Minimize/Maximize the App on a 4K monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

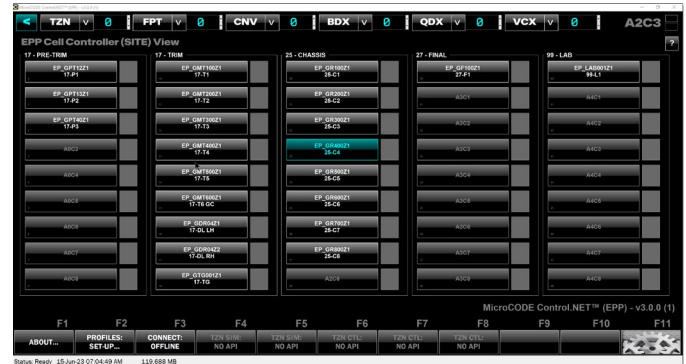
Test Case 9: Resize/Minimize/Maximize the App on a FHD monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

Test Case 10: Resize/Minimize/Maximize the App on a HD monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

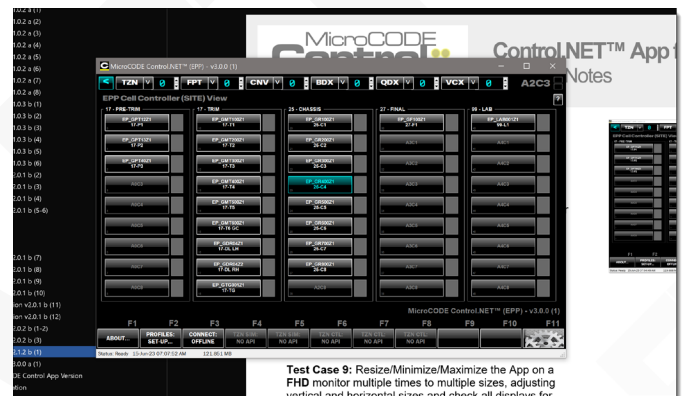
Test Case 11: Resize/Minimize/Maximize the App on a XGA monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

Test Case 12: Resize/Minimize/Maximize the App on a XGA monitor multiple times to multiple sizes, adjusting vertical and horizontal sizes and check all displays for proper layout and readability – **Passed.**

Test Case 13: With the App running on a monitor change the resolution from 4K to FHD to HD to XGA to SVGA and ensure all displays are shown properly – **Passed.**

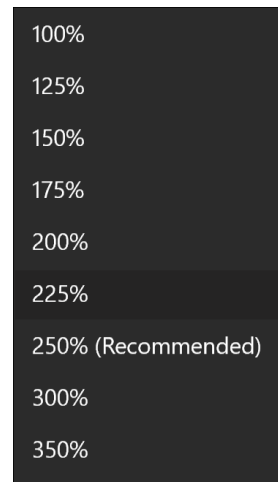


Any size, Any scale, Any monitor resolution....



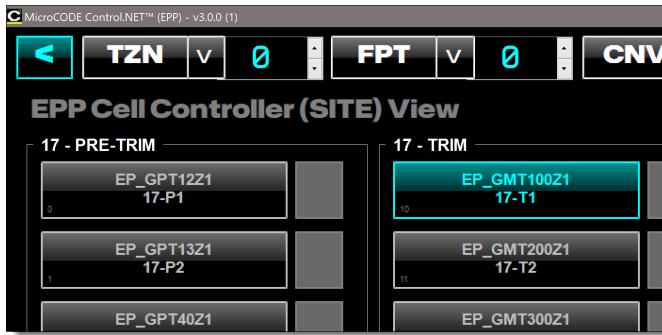
Scaling

Test Case 13: With the App running on a monitor change the **scaling** to each available setting and ensure all displays are shown properly – **Passed.**





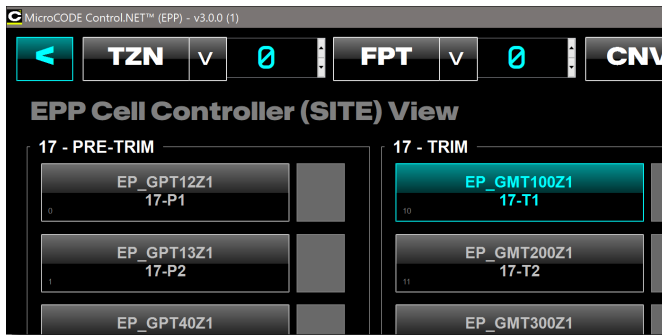
Without our 'Resizer':



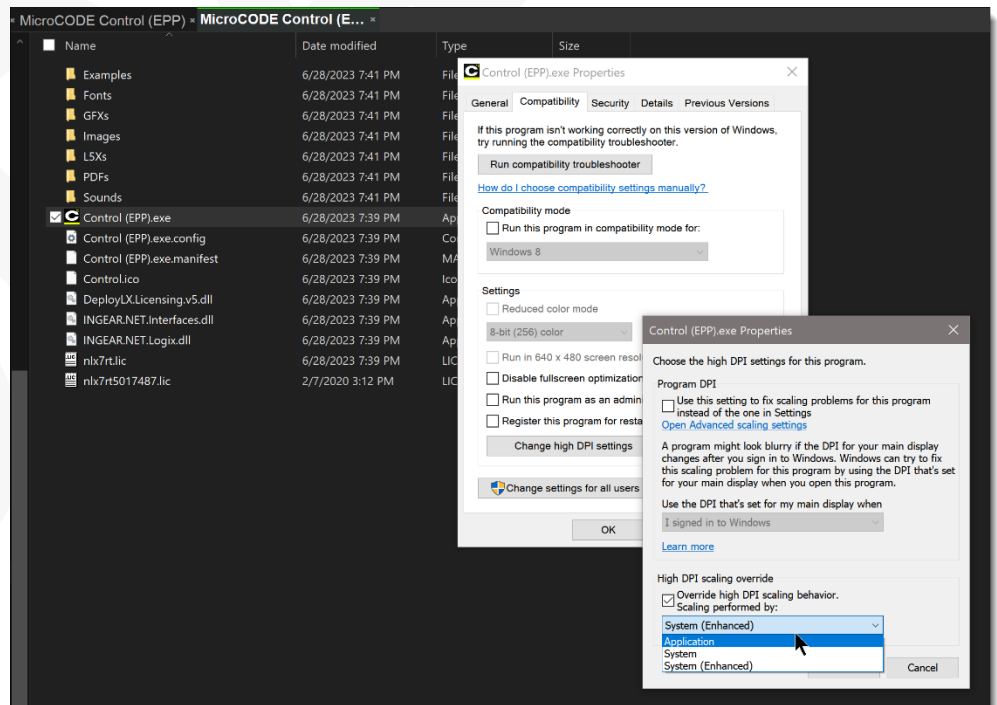
This update is critically dependent on a proper 'App Manifest' being delivered during installation. This is now included (along with an 'App Config' file). The 'Manifest' ensures that Windows 10 knows that the App handles all resizing, scaling, DPI changes, and support 'High DPI' (4K) monitors.

Without this Windows 10 tries to 'help' our Apps by ignoring our resizing code and scaling the app displays itself. This is very slow and looks 'blurry'. Once the proper 'App Manifest' is in place our Apps are 2x – 3x faster at doing the display work and all displays look 'razor sharp'.

With our 'Resizer':



You can see the effects of this if interested by right clicking any App's .EXE and manually changing the 'Compatibility' settings. The one that is important is 'Change high DPI settings > High DPI scaling override > Override high DPI scaling behavior'. Simple right? Ughh.





The critical piece, the **app.manifest**:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <assembly xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:asmv1="urn:schemas-microsoft-com:asmv1" xmlns:asmv2="urn:schemas-microsoft-com:asmv2" xmlns:asmv3="urn:schemas-microsoft-com:asmv3" name="Control (EPF).exe"/>
3 <description>Control.NET™ (EPF)</description>
4
5 <trustInfo xmlns="urn:schemas-microsoft-com:asmv2"
6 <security>
7 <requestedPrivileges xmlns="urn:schemas-microsoft-com:asmv3">
8
9
10 <!-- UAC Manifest Options
11
12 If you want to change the Windows User Account Control level replace the
13 requestedExecutionLevel node with one of the following.
14
15 <requestedExecutionLevel level="asInvoker" uiAccess="false" />
16 <requestedExecutionLevel level="requireAdministrator" uiAccess="false" />
17 <requestedExecutionLevel level="highestAvailable" uiAccess="false" />
18
19 Specifying requestedExecutionLevel element will disable file and registry virtualization.
20 Remove this element if your application requires this virtualization for backwards
21 compatibility. -->
22
23 <requestedExecutionLevel level="asInvoker" uiAccess="false" />
24
25 </requestedPrivileges>
26 </trustInfo>
27
28 <asmv3:application>
29 <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SM/2005/WindowsSettings">
30
31 <!-- DPI AWARE Options
32
33 Indicates that the application is DPI-aware and will not be automatically scaled by Windows at higher
34 DPIs. Windows Presentation Foundation (WPF) applications are automatically DPI-aware and do not need
35 to opt in. Windows Forms applications targeting .NET Framework 4.8 that opt into this setting, should
36 also set the 'EnableWindowsFormsHighDpiAutoResizing' setting to 'true' in their app.config. -->
37
38 <!-- Windows 10 v1507 and later
39
40 <dpiAware> Options:
41 "system" - The App does not handle DPI changes, scaling, and resizing.
42 "system" - DPI-aware for the system, doesn't respond to DPI changes after the application starts.
43 "PerMonitor" - Per-monitor DPI awareness, responds to DPI changes but might not be perfectly scaled.
44 "PerMonitorV2" - Per-monitor for DPI awareness as it improved scaling, but only available on Windows 10 Anniversary
45 Update.
46 <dpiAwareness xmlns="http://schemas.microsoft.com/SM/2016/WindowsSettings" PerMonitorV2 />
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
    
```

```

48 <!-- Legacy (pre-Windows 10 v1507)
49
50 <dpiAware> Options:
51 "true" - The App does not handle DPI changes, scaling, and resizing.
52 "true" - DPI-aware for the system, doesn't respond to DPI changes after the application starts.
53 "true/false" - Per-monitor DPI awareness, responds to DPI changes but might not be perfectly scaled. -->
54
55 <dpiAware xmlns="http://schemas.microsoft.com/SM/2016/WindowsSettings" true/false />
56
57 <!-- Makes the application long-path aware. See https://docs.microsoft.com/windows/win32/fileio/maximum-file
58 <longPathAware xmlns="http://schemas.microsoft.com/SM/2016/WindowsSettings" true />
59
60 </asmv3:windowsSettings>
61 </asmv3:application>
62
63 <compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1"
64 <application>
65
66 <!-- A list of the Windows versions that this application has been tested on
67 and is designed to work with. Uncomment the appropriate elements
68 and Windows will automatically select the most compatible environment. -->
69
70 <!-- Windows Vista -->
71 <!--supportedOS Id="{e281157f-1546-4355-af4e-808d6e3d3d9e}" />-->
72
73 <!-- Windows 7 - removed from supported list to stop 'Compatibility Mode' -->
74 <!--supportedOS Id="{9596b641-f966-429f-90bb-8392f86d0a78}" />-->
75
76 <!-- Windows 8 - not supported -->
77 <!--supportedOS Id="{4a2726a1-526f-4664-b09c-d6968438d683}" />-->
78
79 <!-- Windows 8.1 - not supported -->
80 <!--supportedOS Id="{1197b670-88e1-4239-90bb-8392f86d0a78}" />-->
81
82 <!-- Windows 10 - supported by this MicroCODE App -->
83 <supportedOS Id="{8e777a12-bf32-4f8b-b665-4878853a6274}" />
84
85 </application>
86 </compatibility>
87 </assembly>
    
```

3) Modern Open Source Fonts

The Control.NET™ Apps all needed to move away from the proprietary Fonts for ease of maintenance, future portability, human readability, and commonality.

Requirement: Remove the use of all licensed and proprietary Fonts from the MicroCODE Apps. Use Open Source Fonts where possible to facilitate moving the App to a Web-based deployment in the future.

Implementation: All the MicroCODE Apps now use the following Fonts under the industry standard SIL License.

For all PLC & Configuration Data, a monospaced coding font:
 SIL License: **PROFONT** Windows Font
 Copyright © 1997
 Carl R. Osterwald. All rights reserved.

For MicroCODE Business typography:
 SIL License: **MICROMA** Google Font
 Copyright © 2014
 Vernon Admas. All rights reserved.

For Dialog Box and Message Test:
 SIL License: **LIBRE-FRANKLIN** Google Font
 Copyright © 2015
 Impallari Type - Pablo Impallari. All rights reserved.

For all App UI Controls:
 SIL License: **MONA-SANS** GitHub Font
 Copyright © 2017
 Deni Anggara. All rights reserved.

The complete SIL License is display in all the App ABOUT... boxes.



4) JavaScript Object Notation (JSON)

The Control.NET™ Apps all needed to move away from the Microsoft Windows Registry and the 'flat' .CFG file formats for ease of maintenance, future portability, human readability, and commonality.

Requirement: Make all configuration and data storage files independent of the Windows Registry. Use a modern widely accepted file format.

Implementation: The App was originally designed to store configuration data in the Windows Registry and use a 'flat' text file (.CFG) as an import/export format and for backups.

- JavaScript Object Notation (JSON) was chosen as our data interchange format.
- All use of the Windows Registry has been removed.
- All configuration is not stored in files named:

<App Name>.CFG.json

JSON is an Internet standard established in July 2006 in RFC-4627. It was adopted by the European Computer Manufacturers Association (ECMA) as **ECMA-404** in October 2013.

For more information see:

Wikipedia: JSON ('Jay-sahn')

<https://en.wikipedia.org/wiki/JSON>

Network Working Group - RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON)

<https://datatracker.ietf.org/doc/html/rfc4627>

ECMA-404 - The JSON Data Interchange Syntax

A very concise and well-illustrated document.

https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf

JSON was chosen over XML for its compact notation, human readability and popular support among web developers, GitHub, etc.

OLD 'Flat' .CFG File Format:

```
C:\Users\TimMcGuire\Documents\MicroCODE Control (EPP)\Configuration>_obsolete> Control (EPP).CFG_BAK
1 \Control (EPP)\Site\Active Area Index\0\
2 \Control (EPP)\Site\Active Cell Index\0\
3 \Control (EPP)\Site\Active Track Zone Index\0\
4 \Control (EPP)\Site\A0 Name\CHASSIS 5\
5 \Control (EPP)\Site\A0 Description\<SQL-Configured>\
6 \Control (EPP)\Site\A0 Configured\True\
7 \Control (EPP)\Site\A0 Deployed\False\
8 \Control (EPP)\Site\A0 Cells\1\
9 \Control (EPP)\Site\A0C0 Name\25-C5\
10 \Control (EPP)\Site\A0C0 PLC Name\EPP_ORCSTZ1\
11 \Control (EPP)\Site\A0C0 Description\CHASSIS 5\
12 \Control (EPP)\Site\A0C0 Configured\True\
13 \Control (EPP)\Site\A0C0 Deployed\False\
14 \Control (EPP)\Site\A0C0 TCP/IP Octet.0\120\
15 \Control (EPP)\Site\A0C0 TCP/IP Octet.1\13\
16 \Control (EPP)\Site\A0C0 TCP/IP Octet.2\242\
17 \Control (EPP)\Site\A0C0 TCP/IP Octet.3\30\
18 \Control (EPP)\Site\A0C0 EMP Slot\0\
19 \Control (EPP)\Site\A0C0 I/O Interval\5000\
20 \Control (EPP)\Site\A0C0 TZNui\0\
21 \Control (EPP)\Site\A0C0 BDXui\0\
22 \Control (EPP)\Site\A0C0 QDXui\0\
23 \Control (EPP)\Site\A0C0 VCKui\0\
24 \Control (EPP)\Site\A0C0 BDXoi\0\
25 \Control (EPP)\Site\A0C0 QDXoi\0\
26 \Control (EPP)\Site\A0C0 VCKoi\0\
27 \Control (EPP)\Site\A0C0 Filter PVI\0\
28 \Control (EPP)\Site\A0C0 Filter CSN\0\
29 \Control (EPP)\Site\A0C0 Filter MCODE\0\
30 \Control (EPP)\Site\A0C0 Filter MODEL\MM0000\
```

NEW 'Hierarchical' CFG.json File Format:

```
C:\Users\TimMcGuire\Documents\MicroCODE Control (EPP)\Configuration> Control (EPP).CFG.json > {} Context
1 {
2   "App": {
3     "Name": "Control (EPP)",
4     "Title": "Control.NET™ (EPP)",
5     "Copyright": "© 2012-2023 MicroCODE, Inc.",
6     "Version": "2.1.2 Alpha (4)",
7     "Time": "2023-Jun-14 (14:49:39 PH)",
8     "Licensed Shop": "KUKA ROBOTICS 22500 Key Drive Clinton Township, MI 48036"
9   },
10  "Language": {
11    "Default Language": "English",
12    "LOG Language": "English",
13    "Enable English": true,
14    "Enable Korean": false,
15    "Enable Japanese": false,
16    "Enable Chinese": false,
17    "Enable Thai": false,
18    "Enable German": false,
19    "Enable Spanish": false,
20    "Enable Espanol": false,
21    "Enable Portuguese": false,
22    "Enable Russian": false,
23    "Enable Uzbek": false
24  },
25  "Units": {
26    "Default Units": "Imperial",
27    "LOG Units": "Imperial",
28    "CSV Units": "Imperial",
29    "Enable Imperial": true,
30    "Enable Metric": false
31  },
32  "Logs": {
33    "JSON Days": 60,
34    "LOG Days": 60,
35    "LOG Rollover": 8192,
36    "CSV Days": 60,
37    "CSV Rollover": 8192
38  },
39  "Control": {
40    "Site": {
41      "Active Area Index": 0,
42      "Active Cell Index": 0,
43      "Active Track Zone Index": 0,
44      "Name": "CHASSIS 5",
45      "Description": "<SQL-Configured>",
46      "Configured": true,
47      "Deployed": false,
48      "Cells": 1,
49      "A0C0": {
50        "Name": "25-C5",
51        "PLC Name": "EPP_ORCSTZ1",
52        "Description": "CHASSIS 5",
53        "Configured": true,
54        "Deployed": false,
55        "TCP/IP Octet.0": 120,
56        "TCP/IP Octet.1": 13,
57        "TCP/IP Octet.2": 242,
58        "TCP/IP Octet.3": 30,
59        "EMP Slot": 0,
60        "I/O Interval": 5000,
61        "TZNui": 0,
62        "BDXui": 0,
63        "QDXui": 0,
64        "VCKui": 0,
65        "BDXoi": 0,
66        "QDXoi": 0,
67        "VCKoi": 0,
68        "Filter PVI": 0,
69        "Filter CSN": 0,
70        "Filter MCODE": 0,
71        "Filter MODEL": "MM0000"
72      }
73    }
74  }
75 }
```

This file format is now used by all MicroCODE apps for data interchange.



5) Improved SOUND Control

All the MicroCODE Apps now give the User control of Sound effects, voice prompts, and sound effect testing.

Requirement: Many users work in a quiet office environment and need to suppress unwanted feedback.

Implementation: The Apps now allow you to turn On/Off sound in three (3) categories:

- 1) **Assistant Voice:** this is where the App reads dialog prompts and asks users for help when needed.
- 2) **Sound Effects:** this is where the App uses sound to tell the user than things have succeeded or failed, to alert them to major events, like simulated Job creation.
- 3) **Configuration Data:** this is where a user is actually testing how the App will sound to an Operator on the Plant Floor.

6) App Debug Console .LOG

The Control.NET™ Apps were originally designed to generate Event .LOG files for Production support. These are used by Controls Engineers to diagnose configuration, I/O, and Operator issues.

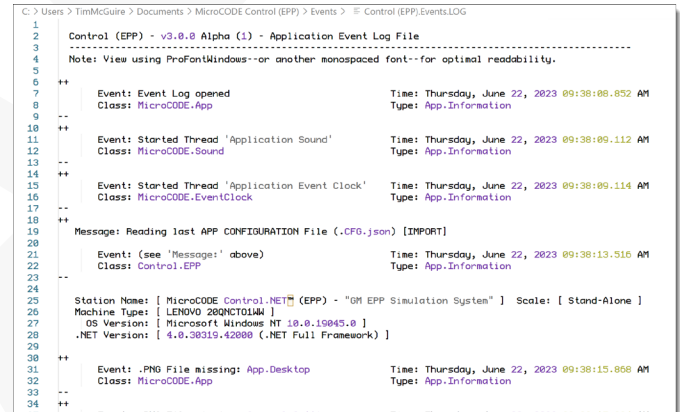
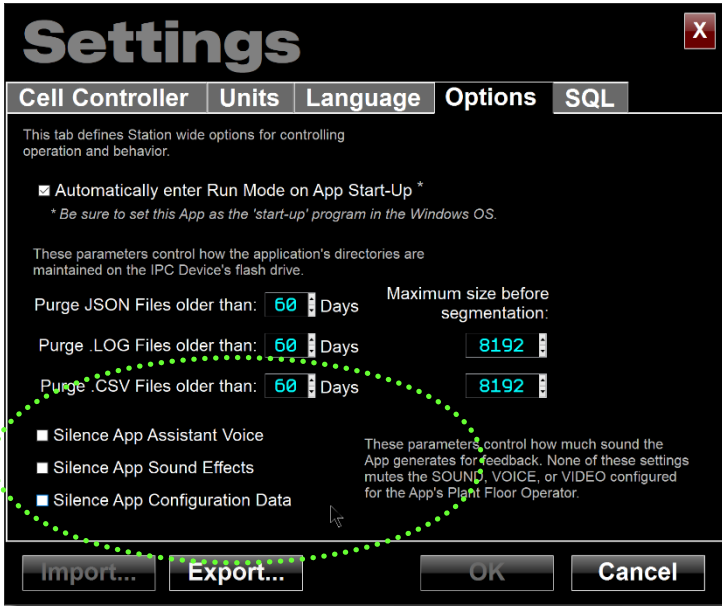
This .LOG file was not intended for Developer messaging, debug statements, data logging.

To clarify this—and help our developers—a new isolated .LOG file is being generated with developer related information only.

Requirement: Permanently remove 'debug' statements from the User's Events.LOG file.

Implementation: The Events .LOG file is kept in the "Events" folder and is named:

<App Name>.DD-MMM-YYY. [HH.MM.SSS] -X. **Events**.LOG



The Debug Console .LOG files is kept in the "Console" folder and is named:

<App Name>.DD-MMM-YYY. [HH.MM.SSS] .**Console**.LOG

The Console .LOG will be almost completely empty unless there is a problem in the App.





7) App I/O Memory

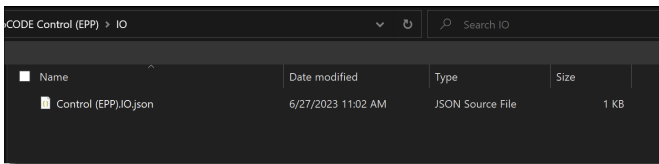
The Control.NET™ Apps now support persistent I/O memory via JSON file storage.

This .IO.json file will be used in future releases to retain the 'last state' of critical I/O devices and memory structures as the Control.NET Apps begin to control real-world I/O devices.

Requirement: Persist I/O 'latched' memory thru App power cycles.

Implementation: The Events .LOG file is kept in the "Events" folder and is named:

<App Name>.IO.json



8) Generate defaults GEPICS Formats and GEPICS Seed Jobs

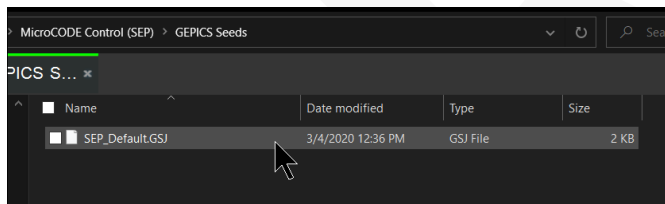
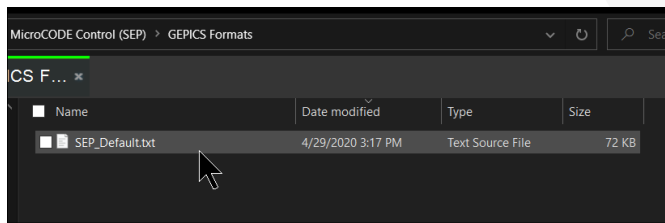
The Control App needs to see the GEPICS Format(s) (both AREA and TRACK ZONE for EPP) when interacting with the EPP or SEP PLCs.

It also needs a 'SEED JOB' to start new Jobs with, before applying User Profiles to modify Option content.

In previous versions the user could not use the App until they had these files. (Examples were supplied but the were delivered under the 'Program files' folder and were not obvious to access).

Requirement: Default the GEPICS Format(s) and Seed Job to something automatically to ensure a new user can get started with simulations immediately.

Implementation: Any use of the App without a selected GEPICS Format(s) or Seed Job will automatically copy the supplied default files into the User's documents folder and use them by default until the User selects something Plant specific.



9) Display Jobs Per Hour (JPH)

The App now shows Jobs Per Hour (JPH) in its Conveyor User Control to support proper simulation (or monitor) of the intended Production Environment.

Requirement: When testing SEP or EPP its helpful to run the Simulation at the target speed of the production Plant.

Implementation: the JPH that results from the existing Faster / 'Slower' commands is not displayed for the User to see...





Corrected in Version v3.0.0 a (1)

The following defects were fixed this Release:

1) Improved App Performance

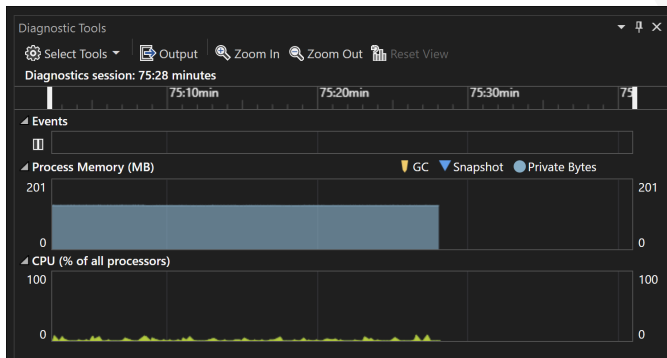
The App is now 10x faster than previous versions and uses less than 7% of available CPU time.

Issue: App was very slow and sometimes unresponsive when connected to a full-scale EPx deployment.

Correction: An initialization issue was creating a zero wait-state in the communication threads to all open PLCs. This caused non-stop I/O to the connected PLCs and was utilizing 100% of the user machine. This was corrected and zero-wait state detection was added to the **MicroCODE.Device** component to ensure this can never happen again.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Connected the Control App to a full-scale SEP deployment and monitor CPU utilization on the user machine – **Passed:** 1%-7% CPU usage with all Controllers active.



2) App resizing and scaling performance

All MicroCODE Apps utilize our own “MicroCODE.Resizer” C# library for handling monitors of different sizes, resolutions, and user scaling.

Issue: Under some conditions resizing was occurring multiple times based on redundant Windows events.

Correction: Additional context and event handling has been added to minimize all redrawing on resizing events. Additionally, the MicroCODE Resizer now uses control ‘invisibility’ during resizing for better performance.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Drag App from a 4K monitor to another, the App should only redraw a single time. – **Passed:** App does a single redraw at final size/scale.

Test Case 2: Minimize, Maximize, and free size, the App should only redraw a single time. – **Passed:** App does a single redraw at final size/scale.



3) Job Focus appears incorrect

The Job Focus shows the wrong PVI at times.

Issue: If the Job that had focus travels out of the Track Zone its focus was left on the last Footprint in CYAN, showing the original PVI.

Correction: The Job focus now turns gray showing the original PVI but denoting it has left.

TEST CASES: These software tests were performed prior to release to ensure App functionality.

Test Case 1: Allow a PVI focus to exit the Track Zone and check the display. – **Passed:** Focus turns gray and returns to cyan on a new job.



Control App Software Version Numbers

The SEP/GEP CE application software version numbers follow this convention...

vM.m.r c (B)

M = Major software version; represents application architecture, underlying technology, etc. Incrementing this number is associated with a **'Major Release'**.

m = Minor software version; represents new components or tech within the application. Incrementing this number is associated with a **'Component Support Release'**.

r = Incremental Release Number; represents collections of new features within the application. Incrementing this number is associated with a **'New Feature Release'**.

c = Development Cycle as in ALPHA/DEMO, BETA/PILOT, or PRODUCTION. In the case of PRODUCTION, the Cycle label is removed. Changing this label is associated with a **'Code Cycle Promotion'**, i.e.: Internal Build Promotion. This is a rebuild/relabeling only no code is changed. e.g.: v2.0.0 Beta (017), vs. v2.0.0 (001).

B = Build Number. This is the internal build number of the application from within the development group; any time code is changed and released into the Support Staff this number must be incremented, no matter how small the change. Incrementing this number is associated with a **'Defect Correction Release'**.

Current MicroCODE Control App Version

This is the highest currently released version of the MicroCODE Control application:

v3.0.0 a (a)
Alpha

For More Information

See the Control App System documentation on the software distribution **MicroCODE Site**:

Version Compatibility Matrix

This table explains which PLC Code releases are compatible with specific releases of the MicroCODE Control App CE Actions application.

Version	MicroCODE Control CE App – Build Matrix			
App	Win NT OS	.NET	Logix 5000	SEP + FW
v1.0.1	Windows 7 or 10	4.7.2	Up to L7x	v1.0.5 + v24.xx
v1.0.12-14	Windows 7 or 10	4.7.2	Up to L7x	Any + v28.xx
v2.0.0 -1	Windows 10	4.7.2	Up to L8x	Any + v32.xx
v3.m.R	Windows 10	4.5	Up to L8x	Any + v32.xx

NOTE: This release of the App now supports Logix 5000 PLCs up to the L8x Series. This required a new CPI Driver be used in the construction of the App. That Driver requires a 'per Site' License which is included in the cost of the App. This is related to architecture changes Rockwell made in the L8x and the need for an updated communication driver in this App.

Currently the SEP Block Point does not support L8x due to similar restrictions in the GEP Console, GEP Sequence, and SEP Actions Apps.

But this App is ready.

This app is part of our Extensible Error Proofing collection...



This application was designed, developed, and is owned by:

MICROCODE INC

Software Development / Controls Engineering – since 1987